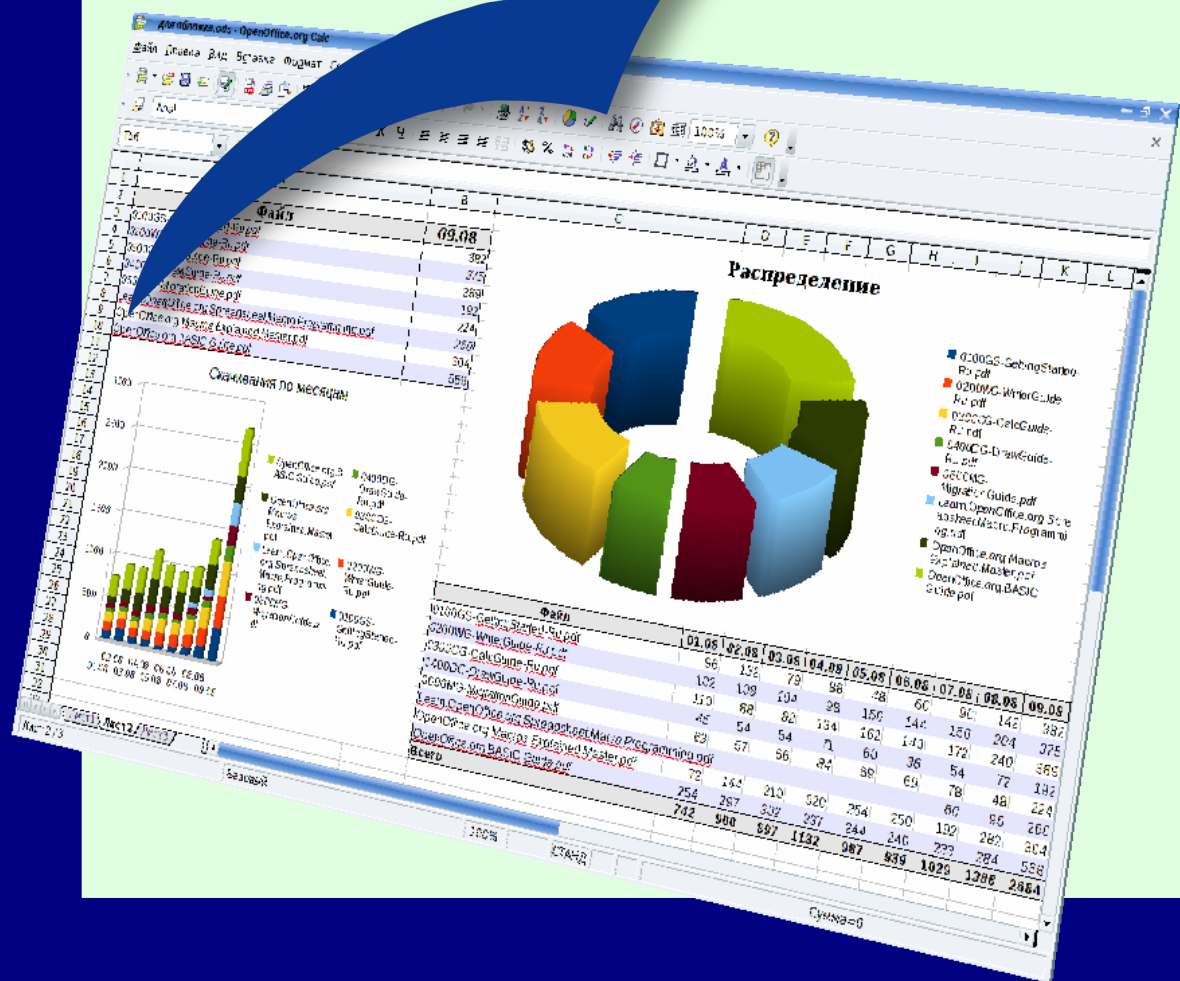


OpenOffice.org



Электронные таблицы:
справочник функций

Calc

OpenOffice.org

**Электронные таблицы:
Справочник функций**

OpenOffice.org. Электронные таблицы: Справочник функций. — СПб, 2008

Книга представляет собой справочное руководство по использованию функций в электронных таблицах офисного пакета OpenOffice.org. В книге приведены описания функций, рассмотрены возможные проблемы и пути их обхода, даны примеры использования функций. Материалы, приведенные в книге, собраны из нескольких источников.

Книга предназначена для пользователей, работающих с электронными таблицами.

© Чернов Дмитрий, 2008

Оглавление

Функции даты и времени.....	1
Введение.....	1
Григорианский и Юлианский календари.....	1
Переносимость.....	2
DATE.....	2
DATEVALUE.....	3
DAY.....	4
DAYS.....	4
DAYS360.....	5
DAYSINMONTH.....	5
DAYSINYEAR.....	6
EASTERSUNDAY.....	6
EDATE.....	6
EOMONTH.....	7
HOUR.....	7
ISLEAPYEAR.....	8
MINUTE.....	8
MONTH.....	9
MONTHS.....	9
NETWORKDAYS.....	10
NOW.....	11
SECOND.....	11
TIME.....	11
TIMEVALUE.....	12
TODAY.....	13
WEEKDAY.....	13
WEEKNUM.....	14
WEEKNUM_ADD.....	15
WEEKS.....	15
WEEKSINYEAR.....	16
WORKDAY.....	16
YEAR.....	17
YEARFRAC.....	17
YEARS.....	18
Функции базы данных.....	20
Краткий обзор.....	20
Критерии.....	21
Параметры настройки для текстовых критериев.....	22
Регулярные выражения в функциях Calc.....	23
DAVERAGE.....	24
DCOUNT.....	25
DCOUNTA.....	26
DGET.....	27
DMAX.....	28
DMIN.....	29
DPRODUCT.....	30
DSTDEV.....	31
DSTDEVP.....	32
DSUM.....	33
DVAR.....	34
DVARP.....	35

Информационные функции.....	37
CELL.....	37
CURRENT.....	39
FORMULA.....	40
ISBLANK.....	40
ISERR.....	40
ISERROR.....	41
ISEVEN.....	42
ISEVEN_ADD.....	42
ISFORMULA.....	43
ISLOGICAL.....	43
ISNA.....	44
ISNONTEXT.....	44
ISNUMBER.....	45
ISODD.....	45
ISODD_ADD.....	46
ISREF.....	47
ISTEXT.....	47
N.....	48
NA.....	49
TYPE.....	49
Логические функции.....	50
Краткий обзор логических значений.....	50
AND.....	50
FALSE.....	51
IF.....	51
NOT.....	52
OR.....	52
TRUE.....	53
Математические функции.....	54
ABS.....	54
ACOS.....	54
ACOSH.....	55
ACOT.....	55
ACOTH.....	55
ASIN.....	56
ASINH.....	56
ATAN.....	57
ATAN2.....	57
ATANH.....	58
BESSELI.....	58
BESSELJ.....	59
BESSELK.....	59
BESSELY.....	59
CEILING.....	60
COMBIN.....	61
COMBINA.....	61
CONVERT.....	62
CONVERT_ADD.....	62
COS.....	66
COSH.....	66
COT.....	67
COTH.....	67
COUNTBLANK.....	68

COUNTIF.....	68
DEGREES.....	70
DELTA.....	70
ERF.....	71
ERFC.....	71
EVEN.....	72
EXP.....	72
FACT.....	73
FACTDOUBLE.....	73
FLOOR.....	74
GCD.....	74
GCD_ADD.....	75
GESTEP.....	76
INT.....	76
LCM.....	77
LCM_ADD.....	77
LN.....	78
LOG.....	78
LOG10.....	79
MOD.....	79
MROUND.....	80
MULTINOMIAL.....	81
ODD.....	81
PI.....	82
POWER.....	82
PRODUCT.....	83
QUOTIENT.....	83
RADIANS.....	83
RAND.....	84
RANDBETWEEN.....	84
ROUND.....	85
ROUNDDOWN.....	86
ROUNDUP.....	86
SERIESSUM.....	87
SIGN.....	88
SIN.....	88
SINH.....	89
SQRT.....	89
SQRTPI.....	89
SUBTOTAL.....	90
SUM.....	91
SUMIF.....	92
SUMSQ.....	94
TAN.....	94
TANH.....	95
TRUNC.....	95
Условный подсчет и суммирование.....	96
Условия в диапазонах ячеек.....	96
Функция SUMPRODUCT.....	97
Функция SUM с формулами массива.....	97
Суммирование элементов с определенным форматированием.....	97
Суммирование элементов, соответствующих элементам в отдельном списке.....	98
Функции комплексных чисел.....	99
История.....	99

COMPLEX.....	100
IMABS.....	100
IMAGINARY.....	101
IMARGUMENT.....	101
IMCONJUGATE.....	102
IMCOS.....	102
IMDIV.....	103
IMEXP.....	103
IMLN.....	104
IMLOG10.....	104
IMLOG2.....	104
IMPOWER.....	105
IMPRODUCT.....	105
IMREAL.....	106
IMSIN.....	106
IMSUB.....	106
IMSUM.....	107
IMSQRT.....	107
Статистические функции.....	109
AVEDEV.....	109
AVERAGE.....	109
AVERAGEA.....	110
B.....	110
BETADIST.....	111
BETAINV.....	111
BINOMDIST.....	112
CHIDIST.....	113
CHIINV.....	113
CHISQDIST.....	113
CHITEST.....	114
CONFIDENCE.....	115
CORREL.....	115
COUNT.....	116
COUNTA.....	116
COVAR.....	117
DEVSQ.....	117
EXPONDIST.....	118
FDIST.....	118
FINV.....	119
FISHER.....	119
FISHERINV.....	119
FORECAST.....	120
FTEST.....	120
GAMMADIST.....	121
GAMMAINV.....	122
GAMMALN.....	122
GAUSS.....	122
GEOMEAN.....	123
HARMEAN.....	123
HYPGEOMDIST.....	124
INTERCEPT.....	124
KURT.....	125
LARGE.....	126
LOGINV.....	126

LOGNORMDIST.....	127
MAX.....	128
MAXA.....	128
MEDIAN.....	128
MIN.....	129
MINA.....	129
MODE.....	130
NEGBINOMDIST.....	130
NORMDIST.....	131
NORMINV.....	132
NORMSDIST.....	132
NORMSINV.....	133
PEARSON.....	133
PERCENTILE.....	134
PERCENTRANK.....	134
PERMUT.....	135
PERMUTATIONA.....	136
PHI.....	136
POISSON.....	137
PROB.....	137
QUARTILE.....	138
RANK.....	139
RSQ.....	140
SKEW.....	140
SLOPE.....	141
SMALL.....	142
STANDARDIZE.....	142
STDEV.....	143
STDEVA.....	143
STDEVP.....	144
STDEVPA.....	145
STEYX.....	145
TDIST.....	146
TINV.....	147
TRIMMEAN.....	147
TTEST.....	147
VAR.....	148
VARA.....	148
VARP.....	149
VARPA.....	149
WEIBULL.....	150
ZTEST.....	151
Финансовые функции.....	153
Финансовые системы дат.....	153
ACCRINT.....	154
ACCRINTM.....	155
AMORDEGRC.....	155
AMORLINC.....	156
COUPDAYBS.....	158
COUPDAYSNC.....	158
COUPNCD.....	159
COUPNUM.....	160
COUPPCD.....	161
CUMIPMT.....	161

CUMIPMT_ADD.....	163
CUMPRINC.....	164
CUMPRINC_ADD.....	165
DB.....	166
DDB.....	167
DISC.....	168
DOLLARDE.....	169
DOLLARFR.....	169
DURATION.....	170
DURATION_ADD.....	170
EFFECT_ADD.....	171
EFFECTIVE.....	172
FV.....	173
FVSCCHEDULE.....	174
INTRATE.....	175
IPMT.....	176
IRR.....	177
ISPMT.....	177
MDURATION.....	178
MIRR.....	179
NOMINAL.....	180
NOMINAL_ADD.....	181
NPER.....	182
NPV.....	182
ODDFPRICE.....	183
ODDFYIELD.....	184
ODDLPRICE.....	185
ODDLYIELD.....	185
PMT.....	186
PPMT.....	188
PRICE.....	189
PRICEDISC.....	190
PRICEMAT.....	191
PV.....	192
RATE.....	194
RECEIVED.....	194
RRI.....	195
SLN.....	196
SYD.....	197
TBILLEQ.....	198
TBILLPRICE.....	198
TBILLYIELD.....	199
VDB.....	200
XIRR.....	202
XNPV.....	203
YIELD.....	204
YIELDDISC.....	205
YIELDMAT.....	206
Примеры использования финансовых функций.....	207
Расчет по кредиту.....	207
Расчет суммы погашения и процентов по кредиту.....	208
Функции CUMIPMT и CUMPRINC.....	209
Создание таблицы выплаты.....	210
Накопление.....	212

Сложное накопление.....	213
Пример Шарового кредита.....	213
Функции электронных таблиц.....	215
ADDRESS.....	215
AREAS.....	216
CHOOSE.....	216
COLUMN.....	217
COLUMNS.....	218
DDE.....	218
ERRORTYPE.....	219
HLOOKUP.....	220
HYPERLINK.....	222
INDEX.....	223
INDIRECT.....	224
INFO.....	225
LOOKUP.....	226
MATCH.....	227
OFFSET.....	228
ROW.....	229
ROWS.....	230
SHEET.....	231
SHEETS.....	231
STYLE.....	232
VLOOKUP.....	233
Функции преобразования чисел.....	237
ARABIC.....	237
BASE.....	237
BIN2DEC.....	238
BIN2HEX.....	239
BIN2OCT.....	240
DEC2BIN.....	241
DEC2HEX.....	242
DEC2OCT.....	242
DECIMAL.....	243
HEX2BIN.....	244
HEX2DEC.....	245
HEX2OCT.....	245
OCT2BIN.....	246
OCT2DEC.....	247
OCT2HEX.....	248
ROMAN.....	249
Текстовые функции.....	251
BAHTTEXT.....	251
CHAR.....	251
CLEAN.....	252
CODE.....	252
CONCATENATE.....	253
DOLLAR.....	253
EXACT.....	253
FIND.....	254
FIXED.....	254
LEFT.....	255
LEN.....	255
LOWER.....	256

MID.....	256
PROPER.....	256
REPLACE.....	257
REPT.....	257
RIGHT.....	258
SEARCH.....	259
SUBSTITUTE.....	259
T.....	260
TEXT.....	261
TRIM.....	261
UPPER.....	262
VALUE.....	262
Функции массивов.....	264
Использование массивов.....	264
Функции, которые понимают массивы в качестве параметров.....	265
Функции, не ожидающие в качестве параметров массивы.....	265
Формулы массивов.....	266
Вычисления формулы массива.....	266
Функции массива.....	267
Проблемы.....	268
FREQUENCY.....	268
GROWTH.....	269
LINEST.....	270
LOGEST.....	272
MDETERM.....	273
MINVERSE.....	273
MMULT.....	274
MUNIT.....	275
SUMPRODUCT.....	275
SUMX2MY2.....	277
SUMX2PY2.....	277
SUMXMY2.....	278
TRANSPOSE.....	278
TREND.....	279
Список литературы.....	281

Функции даты и времени

К функциям даты и времени относятся следующие:

- Функции, берущие параметр типа *Date* или принимают строку даты, которая внутренне может быть преобразована в значение даты с применением функции **DATEVALUE**;
- Функции, берущие дробное значение даты типа *Time* или принимают строку времени, которая внутренне может быть преобразована во дробное значение даты с применением функции **TIMEVALUE**.

Введение

В Calc, даты и время представляются числами. Например число 39441 может представлять дату 25 Декабря 07. Мы можем ввести 39441 в ячейку, а затем (выбрав **Формат - Ячейки ...**) выбрать формат даты, чтобы отобразить число в виде даты.

Хотя число, представляющее дату/время — то же самое, что и любое другое число (за исключением того, что мы хотим отобразить его в виде даты или времени), может быть полезным использовать термин “дата/время в числовом формате”. Дата в числовом формате — просто число дней, которые прошли начиная с выбранной начальной даты. По умолчанию (обычно) начальная дата — 30 декабря 1899; выберите **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, чтобы изменить его на 1 января 1904 для программ Apple, или 1 января 1900 для старого программного обеспечения StarCalc 1.0 в случае необходимости.

Время представляется как часть дня — например 0,5 — половина дня, или 12 часов, или полдень; 0,25 - четверть дня, или 6:00. Таким образом 39441,25 представляет 25 Декабря 2007 6:00.

Было бы чрезвычайно неудобно, если для ввода даты в ячейку, мы должны были вводить дату в числовом формате в нее, а затем изменять формат отображения. Поэтому Calc пробует сделать это за нас: если мы вводим 25 Декабря 07, Calc распознает, что это — дата, преобразует ее в числовой формат и задает формат отображения в виде даты. Calc распознает широкий диапазон возможных записей дат — например 25-декабря-07, 25 декабря 2007, 25 декабря 07, а также 25.12.07 или 12.25.07 (последние два зависят от системы дат, которую Вы используете — дни или месяцы записываются сначала).

Если мы вводим только 2 цифры года (например 07 вместо 2007), Calc должен знать, подразумеваем ли мы 2007 или 1907 год. В **Сервис — Параметры — OpenOffice.org - Общие** Вы можете установить диапазон лет, в котором Calc признает годы, заданные двумя цифрами.

Часть функций даты и времени Calc возвращает “дату” или “время” (например **TODAY()**). Это просто дата в числовом формате, но если ячейка будет не отформатирована, то Calc покажет дату или время, а не числовой формат.

Примеры в описаниях функций Calc используют международный стандартизированный ISO формат даты для ясности, потому что он не зависит от региональных настроек; например, 23 мая 2009 представляется как 2009-05-23.

Григорианский и Юлианский календари

Календарная система, которую мы используем в настоящее время, называется Григорианским

календарем. В каждом високосном году, месяц февраль имеет 29 дней вместо 28. Мы часто думаем, что високосные годы происходят каждые четыре года (это хорошо работает для сегодняшних дат), но в действительности в нашем Григорианском календаре годы 1800, 1900, 2100, 2200 ... не високосные годы; год является високосным, если он делится на 4, а также не делится 100, или если он делится на 400. Это просто соглашение, которое старается учитывать, что время на земле зависит от вращения вокруг солнца.

Григорианский календарь был принят в различных странах в разное время. В Италии и Испании днем после 4 октября 1582 было 15 октября 1582. В Великобритании изменение было сделано в 1752 году. В США изменения происходили между 1582 и 1867 годами (различные области в разное время). В России григорианский календарь введён в 1918 году декретом Совнаркома, согласно которому в 1918 году после 31 января следовало 14 февраля. Последние страны (например Греция) осуществили переход уже в 1920-ых годах.

Calc вычисляет все даты после 15 октября 1582 года с использованием григорианского календаря. Если день раньше 4 октября 1582, Calc использует Юлианский календарь для этих дат.

В Юлианском календаре, используемом до 4 октября 1582 года, годы являются високосным, если он делится на 4 — то есть каждые 4 года; например 1500 — високосный год.

Однако функции Calc **не должны использоваться с датами до 15 октября 1582** — например, **DAY("1582-10-04")** возвращает 14, а не 4.

Очевидно, некоторая осторожность необходима при интерпретации ранних дат, даже для недавних дат, таких как 1924 год, с тем чтобы гарантировать что даты имеют значение для страны или ситуации.

Переносимость

Excel заявляет, что первая дата в его системе дат — 1 января 1900 или 2 января 1904 (в зависимости от параметров настройки); более ранние даты не действительны.

Есть известная проблема в Excel — он вычисляет год 1900 как високосный год по историческим причинам (очевидно для совместимости с более ранними версиями электронных таблиц), тогда как в Григорианском календаре 1900 не високосный год. Это затрагивает только месяцы январь и февраль в 1900, поскольку система дат Excel не действительна до этого. Более поздние годы, такие как 2100, 2200 вычисляются правильно.

Calc правильно определяет 1900 год как не високосный, и система дат Calc действительна до этой даты, как описано выше.

DATE

Функция **DATE** возвращает дату, заданную как год, месяц и день месяца.

Синтаксис функции:

=DATE(year; month; day)

где:

- **year** — целое число между 1583 и 9956 или между 0 и 99;
- **month** — целое число, представляющее месяц года. Если значение аргумента выходит за пределы диапазона для действительной даты, введенное число месяцев отсчитывается от первого месяца указанного года;
- **day** — целое число, представляющее день месяца. Если значение аргумента выходит за пределы диапазона для действительной даты, введенное число дней отсчитывается от первого дня месяца.

Функция **DATE** возвращает дату, выраженную в виде даты в числовом формате. Значение даты в числовом формате зависит от текущей эпохи. Заметим, что некоторые приложения могут некорректно обращаться с датами до 1904; в частности многие электронные таблицы неправильно утверждают, что 1900 — високосный год (он им не был; в нем нет 1900-02-29).

В файле OpenDocument, параметр вычислений **table:null-date** затрагивает эту функцию.

	A	B	C	D	E
1					
2			09.11.07	=DATE(2007; 11; 9)	
3					
4			01.01.08	=DATE(2007; 12; 32)	
5					
6			29.02.04	=DATE(2004; 3; 0)	
7					
8			08.03.07	=DATE(2006; 15; 8)	
9					

Значения даты позволяют осуществлять “вычисления с переносом”, потому что это обеспечивает удобный способ вычислить “следующее” значение. Это особенно удобно для месяцев, и так как это реализовано для месяцев, то по логике лучше делать то же самое для дней. Excel, Calc и Gnumeric все выполняют подобное вычисление с переносом. Kspread 1.4.2 и Lotus1-2-3 v9 ограничивают параметры “разумными” значениями, и не принимают и не осуществляют вычисление с переносом при выходе значения за диапазон; эти программы потребуют изменений для удовлетворения своих требований, но, поскольку действительные таблицы данных для этих систем, как правило, не имеют их вообще, они должны быть совместимы.

Calc выдает ошибку для дат меньше чем 15 октября 1582, первый день Григорианского календаря. Многие реализации не принимают (или обрабатывают не правильно) даты до 1904-01-01 или 1900-03-01.

Некоторые реализации не позволяют отрицательные числа года для функция **DATE()**, хотя это правдоподобное расширение и не запрещается спецификацией.

DATEVALUE

Функция **DATEVALUE** возвращает дату в числовом формате, соответствующую дате, заданной в виде текста.

Синтаксис функции:

=DATEVALUE(datetext)

где:

- **datetext** - дата, представленная в виде текста.

Функция **DATEVALUE** возвращает дату в числовом формате, которая может быть отформатирована для чтения как дата. Она вычисляет дату в числовом формате для текстовой строки **datetext**, используя региональные настройки. Эта функция **должна** принимать ISO формат даты (YYYY-MM-DD), который является независимым от региональных настроек. Это семантический эквивалент функции **VALUE(Date)**, если **Date** имеет формат даты, так как текст, соответствующий формату даты автоматически преобразуется к числовому формату даты когда используется как число. Если текст **datetext** имеет объединенный формат даты и времени, например YYYY-MM-DD HH:MM:SS, возвращена целая часть числового формата даты. Если текст Даты не соответствует формату даты или времени, *может* возвратиться ошибка. См. **VALUE** для получения дополнительной информации о форматах даты.

В файле OpenDocument, параметры вычисления **table:null-year** и **table:null-date** затрагивают

эту функцию.

	A	B	C	D	E
1					
2		39409	=DATEVALUE("2007-11-23")		
3					

Возвращается **39409**, дата в числовом формате для 23 ноября 2007 (при условии, что начальная дата задана по умолчанию).

Excel 2000 не позволяет входному значению быть ссылкой, но Excel 2002 позволяет (он позволяет нормальные операции). **DATEVALUE** широко распространен, но не доступен в SheetToGo. Кроме теста на “соответствие формату даты”, это — в основном то же самое, что и **VALUE()**.

DAY

Функция **DAY** возвращает день заданной даты.

Синтаксис функции:

=DAY(date)

где:

- **date** — дата в текстовом виде или в числовом формате.

Функция **DAY** возвращает день в виде целого числа (от 1 до 31).

	A	B	C	D	E
1					
2		4	=DAY("2008-06-04")		
3					
4		23 ноя 83			
5		23	=DAY(B4)		
6					

DAYS

Функция **DAYS** возвращает число дней между двумя датами.

Синтаксис функции:

=DAYS(enddate; startdate)

где:

- **enddate** — заданная конечная дата;
- **startdate** — заданная начальная дата.

startdate и **enddate** могут быть датами в числовом формате или в виде текста (который преобразуется к числовому формату). **DAYS** возвращает **enddate – startdate**. Результат может быть отрицательным.

	A	B	C	D	E
1					
2		2	=DAYS("2008-03-03"; "2008-03-01")		
3					
4		09.06.2008			
5		02.06.2008			
6		7	=DAYS(B4;B5)		
7					

В ячейке **B2** возвращается 2, число дней между 1 марта 2008 и 3 марта 2008. В ячейке **B6** возвращается 7, число дней между датами, заданными в ячейках **B4** и **B5**.

Заметьте, что **DAYS** не учитывает первый день периода, в то время как **NETWORKDAYS** делает это.

DAYS не совместима с Excel.

DAYS действительно возвращает **enddate** – **startdate**; таким образом, если любая из дат имеет компонент времени (например "2008-01-03 15:15"), возвращенное число дней, возможно, не целое число. Чтобы гарантировать в результате целое число, используйте **INT(DAYS())**.

Lotus 1-2-3 v9.8 имеет функцию по имени **DAYS**, но с отличающейся семантикой. Она поддерживает необязательный параметр **basis** с многочисленными вариантами. Без этого параметра, по умолчанию **basis** соответствует системе дат 30/360, таким образом в Lotus 1-2-3 v9.8 **DAYS(DATE(1993;4;16); DATE(1993;9;25))** возвращает — 159, а не — 162. Calc возвращает — 162 для **DAYS**, как показано выше.

DAYS360

Функция **DAYS360** возвращает число дней между двумя датами на основе 360-дневного года (двенадцать 30-дневных месяцев).

Синтаксис функции:

=DAYS360(enddate; startdate; method)

где:

- **enddate** — заданная конечная дата;
- **startdate** — заданная начальная дата;
- **method** — необязательный параметр; если 0 или опущен, используется метод вычисления Американской Национальной Ассоциации Дилеров Ценных бумаг (NASD); если 1 (или > 0) используется европейский метод вычисления.

Вычисление предполагает, что все месяцы имеют по 30 дней, таким образом год (12 месяцев) имеет 360 дней.

	A	B	C	D	E	F
1						
2			180	=DAYS360("2008-02-29"; "2008-08-31")		
3						

Возвращается 180, то есть 6 месяцев по 30 дней.

DAYSINMONTH

Функция **DAYSINMONTH** возвращает число дней в месяце.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=DAYSINMONTH(date)

где:

- **date** — дата, заданная в виде текста или в числовом формате даты.

Функция **DAYSINMONTH** возвращает число дней в месяце, в котором находится указанная дата.

	A	B	C	D	E
1					
2		29	=DAYSINMONTH("2012-02-14")		
3					

В ячейке **B2** возвращается 29, число дней в феврале 2012 года (2012 — високосный год).

Функция **DAYSINMONTH** не является частью готовящегося международного стандарта ODFE, и не должна расцениваться как переносимая. Альтернативой может быть **DAY(DATE(YEAR(date);MONTH(date)+1;0))**.

DAYSINYEAR

Функция **DAYSINYEAR** возвращает число дней в году.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=DAYSINYEAR(date)

где:

- **date** — дата, заданная в виде текста или в числовом формате даты.

Функция **DAYSINYEAR** возвращает число дней в году, в котором находится указанная дата.

	A	B	C	D	E
1					
2		366	=DAYSINYEAR("2012-02-14")		
3					

В ячейке **B2** возвращается 366, число дней в 2012 году (2012 - високосный год).

Функция **DAYSINYEAR** не является частью готовящегося международного стандарта ODFE, и не должна расцениваться как переносимая. Попробуйте **365+(MONTH(DATE(YEAR(date);2;29))=2)**, если Вы хотите иметь переносимый результат.

EASTERSUNDAY

Функция **EASTERSUNDAY** возвращает дату пасхального воскресенья в заданном году.

Синтаксис функции:

=EASTERSUNDAY(year)

где:

- **year** — целое число между 1583 и 9956 или между 0 и 99, определяющее год.

	A	B	C	D	E
1					
2		23.03.08	=EASTERSUNDAY(2008)		
3					

Возвращает дату 23 марта 2008 г., которая соответствует дате пасхального воскресенья в 2008 году.

Эта функция не доступна в Excel.

EDATE

Функция **EDATE** возвращает дату в числовом формате, отстоящую на заданное количество месяцев вперед или назад от заданной даты.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=EDATE(startdate; months)

где:

- **startdate** — дата, заданная в виде текста или в числовом формате даты;
- **months** — число месяцев, которые добавляются к **startdate**. День месяца остается неизменным, если он не больше чем число дней в новом месяце (тогда он устанавливается в последний день этого месяца). **months** может быть отрицательным.

	A	B	C	D	E
1					
2		39797	=EDATE("2008-10-15"; 2)		
3		15.12.2008	=B2		
4					
5		39568	=EDATE("2008-05-31"; -1)		
6		30.04.2008	=B5		
7					

В ячейке **B2** возвращается дата **15.12.2008** в числовом формате. В ячейке **B5** возвращается **30.04.2008**, так как в апреле только 30 дней.

EOMONTH

Функция **EOMONTH** возвращает дату последнего дня месяца.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=EOMONTH(startdate; addmonths)

где:

- **startdate** — дата, заданная в виде текста или в числовом формате даты;
- **addmonths** - количество месяцев, которые будут добавлены к **startdate** для получения новой даты. Для этой новой даты, **EOMONTH** возвращает дату последнего дня месяца, в виде даты в числовом формате. **addmonths** может быть положительным (даты в будущем), нулевым или отрицательным (даты в прошлом).

	A	B	C	D	E
1					
2		39507	=EOMONTH("2008-02-14"; 0)		
3		29.02.2008	=B2		
4					
5		16.12.07			
6		39507	=EOMONTH(B5;2)		
7		29.02.2008	=B6		
8					

В ячейке **B2** возвращается число **39507**, которое может быть отформатировано как **29 февраля 2008**. 2008 год является високосным. В ячейке **B6** также возвращается число **39507**, которое может быть отформатировано как **29 февраля 2008**.

HOURL

Функция **HOURL** возвращает час заданного времени.

Синтаксис функции:

=HOUR(time)

где:

- **time** — время в текстовом виде или в числовом формате.

Функция **HOUR** возвращает час для **time** в виде целого числа от 0 до 23. Формулу, по которой осуществляется расчет, можно представить в следующем виде:

$$\text{DayFraction} = (\text{time} - \text{INT}(\text{time}))$$

$$\text{HOUR} = \text{INT}(\text{DayFraction} * 24)$$

	A	B	C	D	E
1					
2		21	=HOUR("2008-01-06 21:30:15")		
3					
4		09:25:10			
5		9	=HOUR(B4)		
6					

ISLEAPYEAR

Функция **ISLEAPYEAR** определяет, находится ли указанная дата в високосном году или нет.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=ISLEAPYEAR(date)

где:

- **date** — дата, заданная в виде текста или в числовом формате даты.

Функция **ISLEAPYEAR** возвращает 1, если дата находится в високосном году, и 0 в противном случае.

	A	B	C	D	E
1					
2		1	=ISLEAPYEAR("2012-05-24")		
3					

Возвращается 1; 2012 — високосный год.

Эта функция не доступна в Excel. Попробуйте **=MONTH(DATE(YEAR(date);2;29))=2**, если Вам необходима совместимость с Excel.

MINUTE

Функция **MINUTE** возвращает минуты для заданного времени.

Синтаксис функции:

=MINUTE(time)

где:

- **time** — время в текстовом виде или в числовом формате.

Функция **MINUTE** возвращает минуты для заданного времени в виде целого числа от 0 до 59. Формулу, по которой осуществляется расчет, можно представить в следующем виде:

$$\text{DayFraction} = (\text{time} - \text{INT}(\text{time}))$$

$$\text{HourFraction} = (\text{DayFraction} * 24 - \text{INT}(\text{DayFraction} * 24))$$

$$\text{MINUTE} = \text{INT}(\text{HourFraction} * 60)$$

	A	B	C	D	E
1					
2					
3					
4					
5					
6					

2					
3					
4					
5					
6					

MONTH

Функция **MONTH** возвращает месяц заданной даты.

Синтаксис функции:

=MONTH(date)

где:

- **date** — дата в текстовом виде или в числовом формате.

Функция **MONTH** возвращает месяц даты в виде целого числа, где январь — 1, а декабрь — 12.

	A	B	C	D	E
1					
2					
3					
4					
5					
6					

2					
3					
4					
5					
6					

MONTHS

Функция **MONTHS** возвращает количество месяцев между двумя датами.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=MONTHS(startdate; enddate; mode)

где:

- **startdate** — заданная начальная дата;
- **enddate** — заданная конечная дата;
- **mode** — режим. Если **mode** — 0, **MONTHS** возвращает число целых месяцев между **startdate** и **enddate**, от дня месяца до дня месяца (см. примеры). Если **mode** — 1, **MONTHS** идентифицирует месяц, как содержащие **startdate** и **enddate**, и возвращает разницу между этими месяцами. Другими словами она возвращает:

$$\text{MONTH}(\text{enddate}) - \text{MONTH}(\text{startdate}) + 12 * (\text{YEAR}(\text{enddate}) - \text{YEAR}(\text{startdate}))$$

Если **startdate** будет после **enddate** результат будет отрицательным.

	A	B	C	D	E	F
1						
2		14	=MONTHS("2010-04-03"; "2011-06-17"; 0)			
3						
4		0	=MONTHS("2010-03-31"; "2010-04-30"; 0)			
5						
6		2	=MONTHS("2010-03-31"; "2010-06-30"; 0)			
7						
8		14	=MONTHS("2010-04-03"; "2011-06-17"; 1)			
9						
10		1	=MONTHS("2010-03-31"; "2010-04-01"; 1)			
11						

В ячейке **B2** возвращается 14, число месяцев между 3 апреля 2010 г. и 17 июня 2011 г.

В ячейке **B4** возвращается 0. С 31 марта до 30 апреля не считается целым месяцем, даже при том, что оба они - последний день своих месяцев.

В ячейке **B6** возвращается 2. Даже при том, что обе даты — последний день своих месяцев, последний месяц не учитывается, потому что в нем день меньше чем 31.

В ячейке **B8** возвращается 14, число месяцев между апрелем 2010 и июнем 2011.

В ячейке **B10** возвращается 1. Даже при том, что даты располагаются на расстоянии в один день, они лежат в разных месяцах.

Эта функция не доступна в Excel.

NETWORKDAYS

Функция **NETWORKDAYS** возвращает число рабочих дней между двумя датами. Праздники и выходные в это число не включаются.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=NETWORKDAYS(startdate; enddate; hols_range)

где:

- **startdate** — заданная начальная дата;
- **enddate** — заданная конечная дата;
- **hols_range** — (необязательный) диапазон, содержащий даты праздничных дней.

Суббота и воскресенье, как предполагают, являются нерабочими днями.

	A	B	C	D	E	F
1						
2		2	=NETWORKDAYS("2008-09-05"; "2008-09-08")			
3						
4		1	=NETWORKDAYS("2008-09-05"; "2008-09-08"; B6:B7)			
5						
6		08.09.08				
7		25.12.08				
8						

В ячейке **B2** возвращается 2. 5 сентября 2008 г. — пятница, а 8 сентября 2008 г. — понедельник. Пятница и понедельник учитываются как рабочие дни; а суббота и воскресенье игнорируются.

Ячейки **B6:B7** содержат 8 сентября 2008 г. и 25 декабря 2008 г., поэтому в ячейке **B4**

возвращается 1. 5 сентября 2008 г. — пятница, которая является рабочим днем; суббота и воскресенье игнорируются; 8 сентября 2008 г. — понедельник, но она определена как праздник (в ячейке **B6**).

NOW

Функция **NOW** возвращает текущую дату и время.

Синтаксис функции:

=NOW()

Возвращает текущую дату и время (в виде даты и времени в числовом формате) используя текущие региональные настройки. **NOW** обновляется при каждом перерасчете, например если изменяется ячейка. Если Вам необходим только текущая дата, используйте функцию **TODAY**.

	A	B	C	D
1				
2		01.10.08 11:29	=NOW()	
3				

SECOND

Функция **SECOND** возвращает секунды для заданного времени.

Синтаксис функции:

=SECOND(time)

где:

- **time** — время в текстовом виде или в числовом формате.

Функция **SECOND** возвращает секунды для заданного времени в виде целого числа от 0 до 59. Формулу, по которой осуществляется расчет, можно представить в следующем виде:

$$\begin{aligned} \text{DayFraction} &= (\text{time} - \text{INT}(\text{time})) \\ \text{HourFraction} &= (\text{DayFraction} * 24 - \text{INT}(\text{DayFraction} * 24)) \\ \text{MinuteFraction} &= (\text{HourFraction} * 60 - \text{INT}(\text{HourFraction} * 60)) \\ \text{SECOND} &= \text{INT}(\text{MinuteFraction} * 60) \end{aligned}$$

	A	B	C	D	E
1					
2			15	=SECOND("2008-01-06 21:30:15")	
3					
4			09:25:10		
5			10	=SECOND(B4)	
6					

TIME

Функция **TIME** возвращает время, заданное посредством часов, минут и секунд.

Синтаксис функции:

=TIME(hours; minutes; seconds)

где:

- **hours** — целое число, задающее часы;
- **minutes** — целое число, задающее минуты;

- **seconds** — целое число, задающее секунды.

Функция **TIME** возвращает время, выраженное как время в числовом формате. Поскольку время представляет собой дробную часть числового формата даты времени, функция **TIME** возвращает часть дня занимаемую заданным временем, то есть:

$$\text{TIME} = \frac{(\text{hours} * 60 * 60) + (\text{minutes} * 60) + \text{seconds}}{24 * 60 * 60}$$

Если **hours**, **minutes** или **seconds** выходят за пределами диапазона действительного времени, то реализуется “вычисление с переносом”, как показано в примере ниже.

	A	B	C	D	E
1					
2		09:31:20	=TIME(9; 31; 20)		
3		09:32:15	=TIME(9; 31; 75)		
4					

В ячейке B2 возвращается время 9:31:20 (как время в числовом формате). В ячейке B3 возвращается 9:32:15 — осуществляется “вычисление с переносом”, поскольку в минуте только 60 секунд.

Время — подтип числа, где значение времени 1 = 1 день = 24 часа. Отметьте, что время в течение одного дня — дробное число между 0 и 1, таким образом возможно вычислить только приближенное значение правильного времени. Значения параметров **hours**, **minutes** и **seconds** могут быть произвольными числами (они не должны быть ограничены диапазонами 0.. 24, 0..59, или 0..59 соответственно).

Lotus1-2-3 v9 ограничивает параметры **TIME** “разумными” значениями, и не реализует “вычисление с переносом”. Такое поведение не было принято, так как способность реализовывать “вычисление с переносом” полезно. Не думаю, что многие существующие документы зависят от **TIME**, параметры которой ограничиваются разумными значениями.

TIMEVALUE

Функция **TIMEVALUE** возвращает время в числовом формате, соответствующее времени, заданному в виде текста. Время в числовом формате — это десятичная дробь в интервале от 0 до 0,99999999, представляющая время от 0:00:00 до 23:59:59.

Синтаксис функции:

=**TIMEVALUE**(timetext)

где:

- **timetext** — время, заданное в виде текста.

TIMEVALUE возвращает время в числовом формате, которое может быть отформатировано, чтобы читаться в виде времени. Эта функция должна принимать формат времени ISO (HH:MM:SS), который не зависит от региональных настроек. Если текстовый параметр **timetext** имеет объединенный формат даты и времени, например YYYY-MM-DD HH:MM:SS, должна возвращаться дробная часть даты/времени в числовом формате. Однако Calc возвращает полное число (целую и дробную части) если передается объединенная строка даты и времени.

	A	B	C	D	E
1					
2		0,7	=TIMEVALUE("16:48:00")		
3					

Возвращается 0.7, время в числовом формате для 16:48:00. Это может быть отформатировано, для отображения, например, в виде **4:48 PM**.

Gnumeric 1.4 не допускает объединенный формат даты и времени для входного параметра.

TODAY

Функция **TODAY** возвращает текущую дату.

Синтаксис функции:

=TODAY()

Возвращает текущую дату (в виде даты в числовом формате) используя текущие региональные настройки. **TODAY** обновляется при каждом перерасчете, например если изменится ячейка. Она возвращает только дату, а не значение даты и времени; если Вам также необходимо точное время дня, используйте функцию **NOW**.

	A	B	C	D
1				
2		01.10.08	=TODAY()	
3				

Функция **TODAY()** возвращает целое число. Выражение **=INT(TODAY())=TODAY()** возвращает **True**. Это тестовое выражение может вернуть **False** в полночь, так как к **TODAY()** обращаются дважды. Однако это вряд ли может произойти в действительности.

WEEKDAY

Функция **WEEKDAY** возвращает день недели для данной даты.

Синтаксис функции:

=WEEKDAY(date; type)

где:

- **date** — дата в текстовом виде или в числовом формате;
- **type** — определяет тип вычисления. Для **type=1**, будние дни считаются, начиная с воскресенья (это — значение по умолчанию, когда параметр **type** отсутствует). Для **type=2**, будние дни считаются, начиная с понедельника (соответствует 1). Для **type=3**, будние дни считаются, начиная с понедельника (соответствует 0).

День недели	type=1	type=2	type=3
Воскресенье	1	7	6
Понедельник	2	1	0
Вторник	3	2	1
Среда	4	3	2
Четверг	5	4	3
Пятница	6	5	4
Суббота	7	6	5

Функция **WEEKDAY** возвращает целое число, которое представляет день недели. Эта функция полезна для определения выходных и будних дней набора дат.

	A	B	C	D	E
1					
2		7	=WEEKDAY("2008-06-14"; 1)		
3					

Полезность данной функции состоит в возможности определения выходных и будних дней в последовательности дат. Она очень проста в использовании. Заполним столбец последовательностью дат, используя функциональность OpenOffice Calc. Мы можем использовать функцию **WEEKDAY** для включения только требуемых нам дней недели — например, исключая выходные дни.

В примере ниже, мы используем функцию **WEEKDAY** для идентификации выходных дней.

	A	B	C	D	E	F	G
1							
2		01.09.08	Будний	=IF(WEEKDAY(B2;2)>5;"Выходной";"Будний")			
3		02.09.08	Будний	=IF(WEEKDAY(B3;2)>5;"Выходной";"Будний")			
4		03.09.08	Будний	=IF(WEEKDAY(B4;2)>5;"Выходной";"Будний")			
5		04.09.08	Будний	=IF(WEEKDAY(B5;2)>5;"Выходной";"Будний")			
6		05.09.08	Будний	=IF(WEEKDAY(B6;2)>5;"Выходной";"Будний")			
7		06.09.08	Выходной	=IF(WEEKDAY(B7;2)>5;"Выходной";"Будний")			
8		07.09.08	Выходной	=IF(WEEKDAY(B8;2)>5;"Выходной";"Будний")			
9							

Далее Мы используем функцию **WEEKDAY**, чтобы создать столбец дат, который не содержит выходных дней.

	A	B	C	D	E	F
1						
2						
3		01.09.08				
4		02.09.08	=IF(WEEKDAY(B3+1;2)=6;B3+3;B3+1)			
5		03.09.08	=IF(WEEKDAY(B4+1;2)=6;B4+3;B4+1)			
6		04.09.08	=IF(WEEKDAY(B5+1;2)=6;B5+3;B5+1)			
7		05.09.08	=IF(WEEKDAY(B6+1;2)=6;B6+3;B6+1)			
8		08.09.08	=IF(WEEKDAY(B7+1;2)=6;B7+3;B7+1)			
9		09.09.08	=IF(WEEKDAY(B8+1;2)=6;B8+3;B8+1)			
10						

WEEKNUM

Функция **WEEKNUM** возвращает ISO номер недели заданной даты.

Синтаксис функции:

=WEEKNUM(date; mode)

где:

- **date** — дата в текстом виде или в числовом формате;
- **mode** — определяет, какой день считается первым днем недели: 1 для воскресенья, 2 для понедельника.

Функция **WEEKNUM** возвращает номер недели согласно международному стандарту ISO8601, где неделя 1 — неделя, которая содержит 4 января. Чтобы подражать не-ISO поведению функции Excel **WEEKNUM**, используйте **WEEKNUM_ADD**.

	A	B	C	D	E
1					
2			1	=WEEKNUM("2010-01-07"; 2)	
3					

Возвращается 1. 7 января 2010 г. — четверг, а 4 января 2010 г. понедельник этой же недели.

WEEKNUM называют **ISOWEEKNUM** в готовящемся стандарте ODFP.

WEEKNUM_ADD

Функция **WEEKNUM_ADD** возвращает не-ISO номер недели заданной даты.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=WEEKNUM_ADD(date; mode)

где:

- **date** — дата в текстовом виде или в числовом формате;
- **mode** — определяет, какой день считается первым днем недели: 1 для воскресенья, 2 для понедельника.

Функция **WEEKNUM_ADD** возвращает “обычный” не-ISO номер недели, где неделя 1 — неделя, которая содержит 1 января. **WEEKNUM_ADD** подражает поведению функции **WEEKNUM** в Excel. Функция **WEEKNUM** Calc возвращает номер недели согласно международному стандарту ISO8601.

	A	B	C	D	E
1					
2					
3					

2=WEEKNUM_ADD("2010-01-07"; 2)

Возвращается 2. 7 января 2010 г. — четверг, а 1 января 2010 г. — пятница предыдущей недели.

WEEKNUM_ADD называют **WEEKNUM** в готовящемся стандарте ODFD.

WEEKS

Функция **WEEKS** возвращает число недель между двумя датами.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=WEEKS(startdate; enddate; mode)

где:

- **startdate** — заданная начальная дата;
- **enddate** — заданная конечная дата;
- **mode** — режим. Если **mode** = 0, **WEEKS** возвращает число целых недель между **startdate** и **enddate** — что соответствует **INT(number_of_days_difference / 7)**. Если **mode** = 1, **WEEKS** идентифицирует недели с *Понедельника по Воскресенье*, которые содержат **startdate** и **enddate**, и возвращает разность между этими неделями.

Если **startdate** будет после **enddate** результат будет отрицательным.

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						
8						
9						

2=WEEKS("2010-05-03"; "2010-05-17"; 0)

3=WEEKS("2010-05-04"; "2010-05-27"; 0)

2=WEEKS("2010-05-06"; "2010-05-17"; 1)

1=WEEKS("2010-05-09"; "2010-05-10"; 1)

В ячейке **B2** возвращается 2. Обе даты — понедельники, и между ними — 14 дней.

В ячейке **B4** возвращается 3; между этими двумя датами — 23 дня.

В ячейке **B6** возвращается 2; 6 мая 2010 содержится в неделе 3-9 мая, а 17 мая 2010 находится в неделе 17-23 мая, две недели спустя.

В ячейке **B8** возвращается 1. 9 мая 2010 — воскресенье, а 10 мая 2010 — понедельник следующей недели.

Эта функция не доступна в Excel.

WEEKSINYEAR

Функция **WEEKSINYEAR** возвращает число недель в году.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=WEEKSINYEAR(date)

где:

- **date** — дата в текстовом виде или в числовом формате.

Функция **WEEKSINYEAR** возвращает число недель в году, в котором находится **date**. Неделя, как предполагается, начинается в понедельник, и находится в году, если большинство ее дней находится в этом году.

	A	B	C	D	E
1					
2		52	=WEEKSINYEAR("2010-02-14")		
3					
4		53	=WEEKSINYEAR("2009-02-14")		
5					

В ячейке **B2** возвращается 52, число недель в 2010. В ячейке **B4** возвращается 53, число недель в 2009. В 2009, и 1 января и 31 декабря выпадают на четверг; недели, содержащие 1 января и 31 декабря обе имеют большинство своих дней в 2009, и поэтому обе учитываются.

WEEKSINYEAR не входит в разрабатываемый международный стандарт ODF, и не должна рассматриваться как переносимая. Она не доступна в Excel.

WORKDAY

Функция **WORKDAY** возвращает число, представляющее дату, отстоящую на заданное количество рабочих дней вперед или назад от заданной даты. Рабочими днями не считаются выходные дни и дни, заданные как праздничные.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=WORKDAY(startdate; days; hols_range)

где:

- **startdate** — заданная начальная дата;
- **days** — число (которое может быть отрицательным, для поиска даты перед **startdate**), задающее количество не выходных и не праздничных дней до или после **startdate**.
- **hols_range** - (необязательный) диапазон, содержащий даты праздничных дней.

Суббота и воскресенье, как предполагают, являются нерабочими днями.

	A	B	C	D	E	F	G
1							
2		39699	=WORKDAY("2008-09-05"; 1)				
3		08.09.2008	=B2				
4							
5		39700	=WORKDAY("2008-09-05"; 1; F5:F6)			08.09.2008	
6		09.09.2008	=B5			25.12.2008	
7							

В ячейке **B2** возвращается дата в числовом формате для понедельника 8 сентября 2008 г. 5 сентября 08 — пятница. В ячейке **B5** возвращается дата в числовом формате для 9 сентября 2008 г., так как ячейки F5 и F6 содержат даты праздничных дней и 8 сентября 2008 г. определяется как праздник.

YEAR

Функция **YEAR** возвращает год заданной даты.

Синтаксис функции:

=YEAR(date)

где:

- **date** — дата в текстовом виде или в числовом формате.

Если год задается как число с двумя цифрами, т.е. в виде “21.05.15”, то возвращаемый год может быть — 1915 или 2015, в зависимости от точки разрыва в контексте вычисления. В документе OpenDocument, эта точка разрыва определяется посредством **table:null-year**.

Приложения **должны** поддерживать извлечение года из даты, начиная с 1900. Числа года с тремя цифрами предшествуют принятию Григорианского календаря, и могут вернуть или ошибку или число года. Числа года с четырьмя цифрами, предшествующие 1582 (начало Григорианского календаря) могут вернуть или ошибку или число года. Числа года с четырьмя цифрами после 1582 **должны** вернуть число года.

	A	B	C	D	E
1					
2		2008	=YEAR("2008-06-04")		
3					
4		23 ноя 83			
5		1983	=YEAR(B4)		
6					

YEARFRAC

Функция **YEARFRAC** возвращает число лет, включая дробную часть между двумя датами.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=YEARFRAC(startdate; enddate; basis)

где:

- **startdate** — заданная начальная дата;
- **enddate** — заданная конечная дата;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:

- 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
- 1 - точное число дней в месяцах, точное число дней в году;
- 2 - точное число дней в месяце, год имеет 360 дней;
- 3 - точное число дней в месяце, год имеет 365 дней;
- 4 - европейский метод, 12 месяцев по 30 дней каждый.

Функция **YEARFRAC** возвращает число лет между **startdate** и **enddate**; **basis** — целое число, определяющее используемую финансовую систему дат.

	A	B	C	D	E	F
1						
2			2,5	=YEARFRAC("2007-01-01";"2009-07-01";0)		
3						
4			0,498630136986	=YEARFRAC("2008-01-01";"2008-07-01";3)		
5						

В ячейке **B2** возвращается 2,5.

В ячейке **B4** возвращается 0,498630136986.

И Excel, и Calc оба возвращают результаты с незначительной ошибкой в некоторых ситуациях.

Отметьте, что если используется параметр **basis** по умолчанию (т.е. он опущен) используется *не* фактическое число дней в месяце или году.

Имеются другие фактические системы, для которых в настоящее время нет стандартного значения параметра **basis**. В будущих версиях они могут быть добавлены.

YEARS

Функция **YEARS** возвращает число лет между двумя датами.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=YEARS(startdate; enddate; mode)

где:

- **startdate** — заданная начальная дата;
- **enddate** — заданная конечная дата;
- **mode** — режим. Если **mode** — 0, **YEARS** возвращает число целых лет между **startdate** и **enddate**, от дня/месяца до дня/месяца. Если **mode** — 1, **YEARS** идентифицирует год, которому соответствуют **startdate** и **enddate**, и возвращает разницу между этими годами. Другими словами она возвращает $YEAR(enddate) - YEAR(startdate)$.

Если **startdate** будет после **enddate**, результат будет отрицательным.

	A	B	C	D	E	F
1						
2		2	=YEARS("2009-04-03"; "2011-11-17"; 0)			
3						
4		1	=YEARS("2011-02-28"; "2012-02-28"; 0)			
5						
6		0	=YEARS("2012-02-29"; "2013-02-28"; 0)			
7						
8		2	=YEARS("2009-04-03"; "2011-11-17"; 1)			
9						
10		1	=YEARS("2009-12-31"; "2010-01-01"; 1)			
11						

В ячейке **B2** возвращается 2.

В ячейке **B4** возвращается 1. 2012 — високосный год, но период с 28 февраля 2011 г. до 28 февраля 2012 г. классифицируется как целый год.

В ячейке **B6** возвращается 0. 2012 — високосный год, но период с 29 февраля 2012 г. до 28 февраля 2013 г. не классифицируется как целый год, даже при том, что обе даты — последний день февраля.

В ячейке **B8** возвращается 2.

В ячейке **B10** возвращается 1. Даже при том, что только один день между датами, они располагаются в различных годах.

Эта функция не доступна в Excel.

Функции базы данных

Краткий обзор

В электронных таблицах OpenOffice.org Calc, “база данных” — просто таблица значений, представляющая собой список связанных данных, в котором строки данных являются записями, а столбцы — полями, и не имеет никакого отношения к более сложной базе данных OpenOffice.org Base. Функции базы данных в Calc весьма просты в использовании, и позволяют Вам выполнять анализ исходных данных — но в них также встроена развитая фильтрация данных.

Таблицы “базы данных” Calc могут быть похожи на эту:

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Денизл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						

Первая строка таблицы “базы данных” содержит заголовки (Имя, Класс, Возраст ...), а каждая последующая строка содержит значения данных.

“Критерии”, которые используются для отбора строк из “базы данных”, вводятся в другую таблицу:

	A	B	C	D	E	F
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14				>600		
15						

Первая строка этой таблицы содержит заголовки. Последующие строки определяют критерии: например, используемая выше таблица критериев будет соответствовать тем детям, расстояние до школы у которых — больше чем 600.

Все функции “базы данных” Calc имеют подобную форму. Как простой первый пример, используя таблицы приведенные выше:

=DCOUNT(A1:E10; 0; A13:E14)

Возвращается 5, количество детей, у которых “Расстояние до школы” больше чем 600.

A1:E10 — таблица базы данных, а **A13:E14** - таблица критериев.

Имеются другие функции “базы данных”, которые возвращают сумму, среднее число, стандартное отклонение и так далее.

Критерии

Таблица критериев может размещаться в любом месте электронной таблице, но зачастую разумно, чтобы она находилась рядом с таблицей базы данных.

Заголовки таблицы критериев должны точно соответствовать заголовкам в таблице базы данных, но они могут появиться в любом порядке и не один раз:

	G	H	I	J	K
3					
4		Расстояние до школы	Возраст	Возраст	
5		>600	>8	<=10	
6					

Все критерии в строке должны удовлетворяться для строки, который будет им удовлетворять, поэтому в данном примере выше мы находим строки, которые имеют:

- “Расстояние до школы” больше чем 600 И
- “Возраст”, больше чем 8 И
- “Возраст”, меньше или равен 10.

другими словами школьники 9 и 10 лет, которым приходится проходить до школы больше 600.

Если таблица критериев имеет более чем одну строку критериев, любая удовлетворяющая строка означает, что удовлетворяется вся таблица критериев:

	G	H	I	J	K
3					
4		Расстояние до школы	Возраст	Возраст	
5		>600	>8	<=10	
6			<=8		
7					

В этом примере проверяется:

- (“Расстояние до школы” больше чем 600 И
- “Возраст”, больше чем 8 И
- “Возраст”, меньше или равен 10)

ИЛИ

- (“Возраст”, меньше или равен 8).

Другими словами мы ищем строки, в которых дети соответствуют или “в возрасте 9 и 10 лет, которым приходится проходить до школы больше 600” или “в возрасте 8 лет или менее”.

Условие, которое вводится в ячейку таблицы критериев (т.е. >4) - просто текст и имеет следующую форму:

comparator value

где:

- **comparator** — одно из >, <, >=, <=, =, <> (если **comparator** опущен, принимается =);
- **value** — значение (число или текст) для сравнения.

Например:

- условие “> 4” проверяет, что содержимое ячеек больше чем 4;
- условие “<=8” проверяет, что содержимое ячеек находится в алфавитном порядке

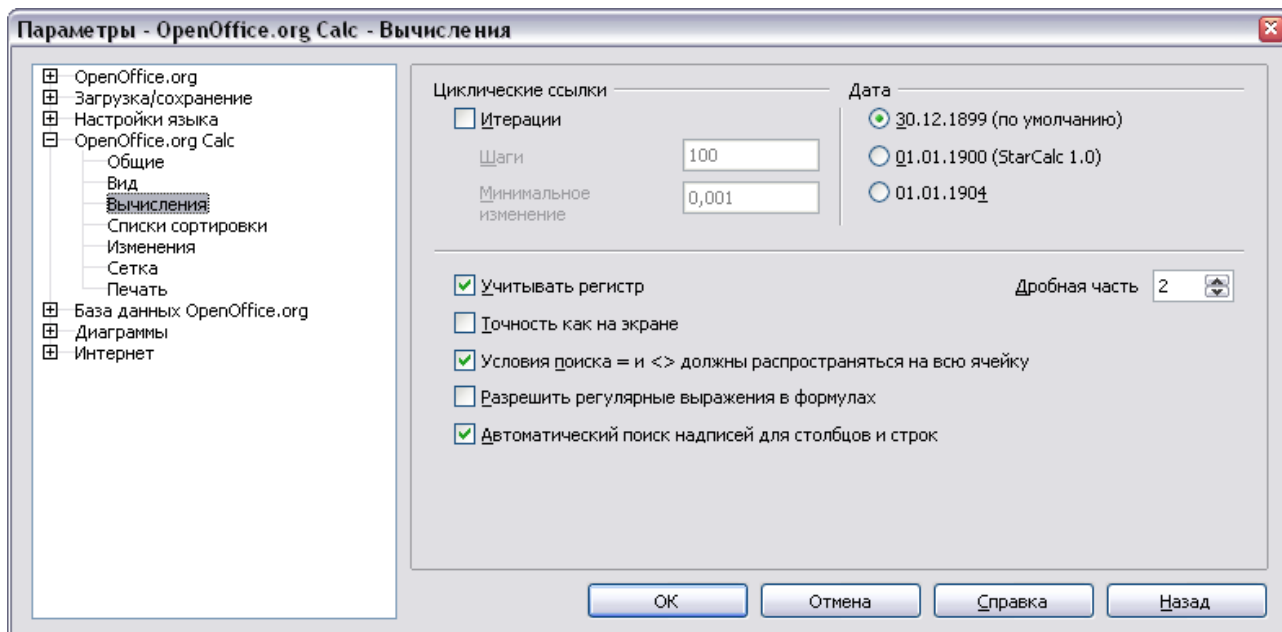
прежде, чем **lamp** (таким образом **lady** и **ant** соответствуют условию, но **late** и **zebra**, не соответствуют);

- условие “**lamp**” проверяет, что содержимое ячеек — **lamp**.

Однако Calc имеет несколько параметров настройки, которые определяют точное поведение — пожалуйста прочитайте следующий раздел.

Параметры настройки для текстовых критериев

Когда проверяется соответствие простому текстовому условию (такому как “lamp”) может быть важно проверить параметры настройки в диалоге **Сервис — Параметры — OpenOffice.org Calc — Вычисления**:



Учитывать регистр

этот флажок не имеет никакого влияния на функции баз данных Calc. Регистр игнорируется — таким образом, “**lamp**” всегда соответствует **lamp**, **Lamp** и **LAMP** и т.д.

Условия поиска = и <> должны распространяться на всю ячейку

если этот флажок установлен, то “**lamp**” будет соответствовать ячейке, содержащей только **lamp**. Если этот флажок не установлен, то “**lamp**” будет соответствовать **lamp**, **clamp**, **lamproon** и т.д. — другими словами ячейка будет удовлетворять условию, если **lamp** будет найдена где-нибудь в тексте ячейки.

Разрешить регулярные выражения в формулах

если этот флажок установлен, то условие рассматривается как регулярное выражение. Например, условие “**!amp**” соответствовала бы **lamp** и **amp** (в регулярных выражениях “**!**” означает, что “**I**” является необязательным). Регулярные выражения подобны групповым символам, но более мощные; они описаны в разделе *Регулярные выражения в функциях Calc*. (Заметьте, что “**Условия поиска = и <> должны распространяться на всю ячейку**” *устанавливающиеся выше также работают, когда выбраны регулярные выражения.*)

Результаты функций могут зависеть от этих параметров настройки — но, к сожалению, они являются скорее скрытыми так, чтобы пользователь не мог знать, если параметры заданы неправильно.

Есть два способа бороться с этим:

Во-первых Вы можете разрабатывать электронную таблицу, чтобы работать независимо от параметров настройки. Наиболее легко, никогда не делать сравнение текста — или если Вы

его делаете, всегда смотрите что соответствует вся ячейка, и удостоверьтесь, что ни одна из ячеек, которые Вы проверили, не соответствует частично любой другой ячейке (например, если Вы ищете “apple” как единственное содержимое ячейки, удостоверьтесь, что никакая другая ячейка не может содержать “crabapple”), И

не используйте регулярные выражения И

не используйте специальные символы регулярных выражений, таких как ., *, +, [, { в критериях .

Во-вторых Вы можете включить в вашей электронной таблице предупреждение, если параметры настройки неправильны. В ячейке **A3** введите текст: **Check:**.

и создайте соответствующую формулу в ячейке **A4** - например:

	A	B	C	D
1				
2				
3	Check:			
4	Error: Turn on regular expressions.			
5				

Теперь заметьте, что:

COUNTIF(A3;".*") — возвращает 1, если регулярные выражения включены.

COUNTIF(A3;"<e") — возвращает 1, если распространение на всю ячейку включено.

Для проверки и регулярных выражений и соответствия всей ячейки:

=IF(AND(COUNTIF(A3; ".*"); COUNTIF(A3; "<e")); "OK"; "Error: " & IF(COUNTIF(A3;".*")=0; "Включите регулярные выражения. "; "")) & IF(COUNTIF(A3;"<e")=0; "Включите соответствие всей ячейке."; ""))

Для проверки регулярных выражений, но без соответствия всей ячейки:

=IF(AND(COUNTIF(A3;".*"); COUNTIF(A3;"<e")=0); "OK"; "Error: " & IF(COUNTIF(A3;".*")=0; " Включите регулярные выражения. "; "")) & IF(COUNTIF(A3;"<e"); "Выключите соответствие всей ячейке."; ""))

Для проверки соответствия всей ячейки, но без регулярных выражений:

=IF(AND(COUNTIF(A3;".*")=0; COUNTIF(A3;"<e")); "OK"; "Error: " & IF(COUNTIF(A3;".*"); "Выключите регулярные выражения."; "")) & IF(COUNTIF(A3;"<e")=0; "Включите соответствие всей ячейке ."; ""))

Для проверки что отключены регулярные выражения и соответствие всей ячейки:

=IF(AND(COUNTIF(A3;".*")=0; COUNTIF(A3;"<e")=0); "OK"; "Error: " & IF(COUNTIF(A3;".*"); "Выключите регулярные выражения. "; "")) & IF(COUNTIF(A3;"<e"); "Выключите соответствие всей ячейке."; ""))

Регулярные выражения в функциях Calc

Есть множество функций в Calc, которые позволяют использовать регулярные выражения:

SUMIF, COUNTIF, MATCH, SEARCH, LOOKUP, HLOOKUP, VLOOKUP, DCOUNT, DCOUNTA, DSUM, DPRODUCT, DMAX, DMIN, DAVERAGE, DSTDEV, DSTDEVP, DVAR, DVARP, DGET.

Действительно ли используются регулярные выражения, выбирается в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления.**

Например **=COUNTIF (A1:A6; "r.d")**, с установленным флажком “Разрешить регулярные выражения в формулах”, будет подсчитывать ячейки в диапазоне A1:A6, которые содержат “red” и “ROD”.

Кроме того, если флажок “Условия поиска = и <> должны распространяться на всю ячейку” не установлен, то “Fred”, “bride” и “Ridge” также будут подсчитаны.

	A	B	C	D
1	Fred			
2	red			
3	ROD			
4	bride			
5	blue			
6	Ridge			
7				5

Поиск с использованием регулярных выражений в пределах функций — всегда не зависит от регистра, независимо от установки флажка “Учитывать регистр” в вышеуказанном диалоговом окне — поэтому “red” и “ROD” всегда будут учитываться в вышеупомянутом примере. Эта независимость от регистра также относится к структурам регулярных выражений (**[[:lower:]]**) и (**[[:upper:]]**), которые соответствуют символам независимо от регистра.

Регулярные выражения не будут работать для простых сравнений. Например: **A1 = "r.d"** будет всегда возвращаться **FALSE**, если A1 будет содержать “red”, даже если регулярные выражения будут разрешены - оно возвратит **TRUE**, только если A1 будет содержать “r.d” (“r”, потом точка, потом “d”). Если Вы желаете проверить использование регулярных выражений, пробуйте функцию **COUNTIF** — **COUNTIF(A1; "r.d")**, возвратится '1' или '0', интерпретируемые как **TRUE** или **FALSE** в формулах подобных **=IF (COUNTIF (A1; "r.d"); "ура"; "фу")**.

Установка параметра “Разрешить регулярные выражения в формулах”, означает, что все вышеупомянутые функции будут интерпретировать любые специальные символы регулярных выражений (такие как круглые скобки) используемые в строках в пределах формул, чтобы “избежать” этого, необходимо использовать предшествующую обратную косую черту, чтобы символ воспринимался не как часть регулярного выражения. Эту обратную косую черту будут необходимо удалить, если параметр будет позднее снят.

Отметьте, что в OOo2.4 изменение параметра “Разрешить регулярные выражения в формулах”, не обязательно отражается на результатах, даже если выполнен перерасчет. Эта ошибка 89047 не относится к OOo2.3, и устранена в OOo3.0.

DAVERAGE

Функция **DAVERAGE** возвращает среднее значений в столбце таблицы “базы данных” Calc, строки которых удовлетворяют заданному условию.

Синтаксис функции:

=DAVERAGE(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заклученный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк исследуемой **database_table**.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Дениэл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16						
17		650				
18		650				
19		650				
20						

650=DAVERAGE(A1:E10; "Расстояние до школы"; A13:E14)
650=DAVERAGE(A1:E10; 4; A13:E14)
650=DAVERAGE(A1:E10; D1; A13:E14)

В ячейках **B17**, **B18** и **B19** возвращается среднее расстояние до школы, которое должны преодолевать дети, учащиеся во втором классе (650).

DCOUNT

Функция **DCOUNT** подсчитывает ячейки, содержащие числа в столбце таблицы “базы данных” Calc, строки которых соответствуют заданным критериям.

Синтаксис функции:

=DCOUNT(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца, или 0 для включения всех столбцов.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк исследуемой **database_table**.

	A	B	C	D	E	F	G
1							
2		Имя	Класс	Возраст	Расстояние до школы	Вес	
3		Энди	3	9	150	40	
4		Бетти	4	10	1000		
5		Чарли	3	10	300		
6		Дениэл	5	11	1200	48	
7		Ева	2	8	650	33	
8		Франк	2	7	300		
9		Гретта	1	7	200	36	
10		Гарри	3	9	1200	44	
11		Ирена	2	8	1000	42	
12							
13							
14		Имя	Класс	Возраст	Расстояние до школы	Вес	
15			2				
16							
17							
18			3	=DCOUNT(B2:F11;0;B14:F15)			
19			2	=DCOUNT(B2:F11;5;B14:F15)			
20			2	=DCOUNT(B2:F11;F2;B14:F15)			
21							

В ячейке **C18** возвращается количество детей, которые учатся во втором классе (3).

В ячейке **C19** возвращается количество детей во втором классе, которые были взвешены (2). Франк не включается, потому что его вес не указан (не число).

В ячейке **C20** также возвращается количество детей во втором классе, которые были взвешены (2).

Точное использование 0 в качестве параметра **field** непонятно, как в эксплуатации, так и для совместимости.

DCOUNTA

Функция **DCOUNTA** подсчитывает непустые ячейки в столбце таблицы “базы данных” Calc, строки которых соответствуют заданным критериям.

Синтаксис функции:

=DCOUNTA(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца, или 0 для включения всех столбцов.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк исследуемой **database_table**.

Функция **DCOUNTA** подсчитывает непустые ячейки, то есть те ячейки, которые содержат числа, текст, **TRUE/FALSE** или ошибки.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000		
4	Чарли	3	10	300	нет	
5	Дэниел	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300		
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирен	2	8	1000	нет	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16			3=DCOUNTA(A1:E10;0;A13:E14)			
17			2=DCOUNTA(A1:E10;5;A13:E14)			
18			2=DCOUNTA(A1:E10;E1;A13:E14)			
19						

В ячейке **C16** возвращается количество детей, которые учатся во втором классе (3).

В ячейке **C17** возвращается количество детей, которые учатся во втором классе, вес которых был или измерен или помечен как “Нет” (2). Франк не включен, потому что его вес не указан. Ирен включена, потому что ее вес отмечен как “Нет”, что является текстом.

В ячейке **C18** также возвращает количество детей во втором классе, вес которых был или измерен или отмечен как “Нет” (2).

DGET

Функция **DGET** возвращает содержимое ячейки столбца таблицы базы данных, в уникальной строке удовлетворяющей критерию.

Синтаксис функции:

=DGET(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк исследуемой **database_table**.

Функция **DGET** извлекает значение поля **field** одной записи из диапазона данных в базе данных **database_table**, которая соответствует критериям **criteria_table**. Возвращает **#VALUE!** если строка не найдена, и **Err502**, если найдены более чем одна строка.

	А	В	С	Д	Е	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Дениэл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14	Франк					
15						
16						
17		7				
18		7				
19						

В ячейках В17 и В18 возвращается возраст Франка (7).

DMAX

Возвращает наибольшее значение в столбце таблицы “базы данных” Calc, в строках, которые удовлетворяют заданным критериям.

Синтаксис функции:

=DMAX(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк исследуемой **database_table**.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Дениэл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16						
17				1000		=DMAX(A1:E10; "Расстояние до школы"; A13:E14)
18				1000		=DMAX(A1:E10; 4; A13:E14)
19				1000		=DMAX(A1:E10; D1; A13:E14)
20						

В ячейках **B17**, **B18** и **B19** возвращается максимальное расстояние до школы, которое должны преодолевать дети (1000, для Ирен).

DMIN

Возвращает минимальное значение в столбце таблицы “базы данных” Calc, в строках, которые удовлетворяют заданным критериям.

Синтаксис функции:

=DMIN(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк исследуемой **database_table**.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Дениэл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16						
17				300		=DMIN(A1:E10; "Расстояние до школы"; A13:E14)
18				300		=DMIN(A1:E10; 4; A13:E14)
19				300		=DMIN(A1:E10; D1; A13:E14)
20						

В ячейках **B17**, **B18** и **B19** возвращается минимальное расстояние до школы, которое должны преодолевать дети (300, для Франка).

DPRODUCT

Функция **DPRODUCT** возвращает произведение ячеек в столбце таблицы “базы данных” Calc, для строк, которые удовлетворяют заданным критериям.

Синтаксис функции:

=DPRODUCT(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк **database_table**.

Функция **DPRODUCT** перемножает содержимое всех ячеек, найденных в столбце **field**.

	A	B	C	D	E	F
1	Размер ящика	Измерение	см		Размер ящика	
2	A	ширина	23		C	
3	A	высота	19			
4	A	глубина	35			
5	B	ширина	35			
6	B	высота	35			
7	B	глубина	19			
8	C	ширина	23			
9	C	высота	35			
10	C	глубина	21			
11	D	ширина	19			
12	D	высота	17			
13	D	глубина	12			
14						
15						
16						
17						

16905 =DPRODUCT(A1:C13; "см"; E1:E2)

Возвращается объем ящика размера C - то есть, произведение всех ячеек в колонке "см", для которой размер ящика — C (ширина * высота * глубина = 23 * 35 * 21 = 16905).

DSTDEV

Функция **DSTDEV** возвращает выборочное среднее квадратичное отклонение значений в столбце таблицы “базы данных” Calc, для строк, которые удовлетворяют заданным критериям.

Синтаксис функции:

=DSTDEV(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк **database_table**.

Функция **DSTDEV** возвращает **выборочное среднее квадратичное отклонение**, которое предполагает, что выбранные строки **database_table** содержат случайную выборку с нормальным законом распределения из генеральной совокупности. Если выбранные данные — все генеральная совокупность, используйте функцию **DSTDEVP**. Для вычисления **DSTDEV** используется формула:

$$\sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

где N — количество включенных значений, а x_i — эти значения.

Функция **DSTDEV** игнорирует любую ячейку, содержащую текст в столбце **field**.

Проще говоря, среднее квадратичное отклонение — мера того, как широко разбросаны значения данных. Это — квадратный корень *дисперсии* (см. **DVAR**, **DVARP**). Среднее квадратичное отклонение — надежная мера, только в том случае, когда есть достаточно много данных для исследования.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Денизл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16						
17						
18						

5,2 =DSTDEVP(A1:E10; "Вес"; A13:E14)

Возвращается выборочное среднеквадратичное отклонение веса детей во втором классе. Это не очень полезная мера, поскольку имеется небольшое количество детей.

Логические значения **TRUE** и **FALSE** интерпретируются как 1 и 0 при вычислении. Это не совместимо с Excel, который игнорирует логические значения. Подобное поведение будет в редких случаях вызывать затруднения.

DSTDEVP

Функция **DSTDEVP** возвращает среднеквадратичное отклонение значений в столбце таблицы “базы данных” Calc, для строк, которые удовлетворяют заданным критериям.

Синтаксис функции:

=DSTDEVP(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк **database_table**.

Функция **DSTDEVP** возвращает **среднеквадратичное отклонение генеральной совокупности**, которое предполагает, что выбранные строки **database_table** — вся генеральная совокупность с нормальным законом распределения. Если выбранные данные — выборка из генеральной совокупности, используйте функцию **DSTDEV**. Для вычисления **DSTDEVP** используется формула:

$$\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

где N — количество включенных значений, а x_i — эти значения.

Функция **DSTDEV** игнорирует любую ячейку, содержащую текст в столбце **field**.

Проще говоря, среднее квадратичное отклонение — мера того, как широко разбросаны значения данных. Это — квадратный корень *дисперсии* (см. **DVAR**, **DVARP**). Среднее квадратичное отклонение — надежная мера, только в том случае, когда есть достаточно много данных для исследования.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Денизл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16						
17						
18						

4,24 = DSTDEVP(A1:E10; "Вес"; A13:E14)

Возвращается выборочное среднее квадратичное отклонение веса детей во втором классе, если предположить, что это единственные дети во втором классе. Это не очень полезная мера, поскольку имеется небольшое количество детей.

Логические значения **TRUE** и **FALSE** интерпретируются как 1 и 0 при вычислении. Это не совместимо с Excel, который игнорирует логические значения. Подобное поведение будет в редких случаях вызывать затруднения.

DSUM

Функция **DSUM** суммирует ячейки в столбце таблицы “базы данных” Calc, для строк, которые удовлетворяют заданным критериям.

Синтаксис функции:

=DSUM(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк **database_table**.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Дениэл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16						
17				1950		=DSUM(A1:E10; "Расстояние до школы"; A13:E14)
18				1950		=DSUM(A1:E10; 4; A13:E14)
19				1950		=DSUM(A1:E10; D1; A13:E14)
20						

В ячейках **B17**, **B18** и **B19** возвращается суммарное расстояние до школы всех детей, которые учатся во втором классе (1950).

DVAR

Функция **DVAR** возвращает исправленную выборочную дисперсию значений в столбце таблицы “базы данных” Calc, для строк, которые удовлетворяют заданным критериям.

Синтаксис функции:

=DVAR(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк **database_table**.

Функция **DVAR** возвращает **исправленную выборочную дисперсию**, которая предполагает, что выбранные строки **database_table** содержат случайную выборку нормально распределенной генеральной совокупности. Если выбранные данные — вся генеральная совокупность, используйте функцию **DVARP**. Для вычисления функция **DVAR** использует формулу:

$$\frac{1}{(N-1)} \sum_{i=1}^N (x_i - \bar{x})^2$$

где N — количество включенных значений, а x_i — эти значения.

Функция **DVAR** игнорирует любую ячейку, содержащую текст в столбце **field**.

Проще говоря, дисперсия — мера того, как широко разбросаны значения данных. Это квадрат *среднеквадратичного отклонения* (см. **DSTDEV**, **DSTDEVP**). Дисперсия —

надежная мера, только если есть достаточно многие данных для исследования.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Денизл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16						
17						
18						

27 =DVAR(A1:E10; "Вес"; A13:E14)

Возвращается выборочная дисперсия веса детей во втором классе. Это не полезная мера, поскольку имеется немного детей.

Логические значения **TRUE** и **FALSE** интерпретируются как 1 и 0 при вычислении. Это не совместимо с Excel, который игнорирует логические значения. Подобное поведение будет в редких случаях вызывать затруднения.

DVARP

Функция **DVARP** возвращает дисперсию генеральной совокупности значений в столбце таблицы “базы данных” Calc, для строк, которые удовлетворяют заданным критериям.

Синтаксис функции:

=DVARP(database_table; field; criteria_table)

где:

- **database_table** — диапазон, определяющий данные, которые будут обработаны.
- **field** — столбец для подсчета. Это может быть номер столбца (1 — первый столбец таблицы базы данных, 2 — второй ...), или заголовок столбца (заключенный в кавычки “”), или ячейка, ссылающаяся на заголовок столбца.
- **criteria_table** — диапазон, содержащий критерии, используемые для выбора строк **database_table**.

Функция **DVARP** возвращает **дисперсию генеральной совокупности**, которая предполагает, что выбранные строки **database_table** — вся нормально распределенная генеральная совокупность. Если выбранные данные — выборка из генеральной совокупности, используйте функцию **DVAR**. Для вычисления функция **DVARP** использует формулу:

$$\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$

где N — количество включенных значений, а x_i — эти значения.

Функция **DVARP** игнорирует любую ячейку, содержащую текст в столбце **field**.

Проще говоря, дисперсия — мера того, как широко разбросаны значения данных. Это квадрат *среднеквадратичного отклонения* (см. **DSTDEV**, **DSTDEVP**). Дисперсия — надежная мера, только если есть достаточно многие данных для исследования.

	A	B	C	D	E	F
1	Имя	Класс	Возраст	Расстояние до школы	Вес	
2	Энди	3	9	150	40	
3	Бетти	4	10	1000	42	
4	Чарли	3	10	300	51	
5	Дениэл	5	11	1200	48	
6	Ева	2	8	650	33	
7	Франк	2	7	300	42	
8	Гретта	1	7	200	36	
9	Гарри	3	9	1200	44	
10	Ирена	2	8	1000	42	
11						
12						
13	Имя	Класс	Возраст	Расстояние до школы	Вес	
14		2				
15						
16						
17						
18						

18 =DVARP(A1:E10; "Вес"; A13:E14)

Возвращает дисперсию генеральной совокупности веса детей во втором классе. Это не полезная мера, поскольку имеется небольшое количество детей.

Информационные функции

Информационные функции возвращают информацию о содержимом ячейки, электронной таблице или лежащем в основе окружении, включая специальные функции для преобразования между типами данных.

CELL

Функция **CELL** возвращает информацию об адресе, форматировании или содержимом ячейки.

Синтаксис функции:

=CELL(info_type; cell_ref)

где:

- **info_type** — символьная строка, которая определяет тип информации. Символьная строка вводится всегда на английском языке. Прописные или строчные буквы — неважно.
- **cell_ref** — положение ячейки, которая будет исследоваться. Если **cell_ref** - диапазон, берется верхняя левая ячейка диапазона. Если **cell_ref** отсутствует, OpenOffice.org Calc использует положение ячейки, в которой расположена формула. Microsoft Excel использует ссылку на ячейку, в которую помещен курсор.

Функция **ISEVEN** возвращает информацию о ячейке **cell_ref**. Возвращаемая информация зависит от текстовой строки **info_type** (независимой от регистра), которая может быть установлена следующим образом:

- **COL** — возвращается номер столбца, на который ссылаются;
- **ROW** — возвращается номер строки, на который ссылаются;
- **SHEET** — возвращается номер листа, на который ссылаются;
- **ADDRESS** — возвращается абсолютный адрес ячейки, на которую ссылаются, в виде текста;
- **FILENAME** — возвращается имя файла и номер листа ячейки, на которую ссылаются, в виде текста;
- **COORD** — возвращается полный адрес ячейки в нотации Lotus™, в виде текста.
- **CONTENTS** — возвращается содержимое ячейки, на которую ссылаются, без форматирования.
- **TYPE** - возвращается текст, который обозначает тип содержимого ячейки:
 - **b** (blank) обозначает пустую ячейку;
 - **l** (label) обозначает текст, или результат формулы в виде текста;
 - **v** (value) обозначает число, или результат формулы в виде числа;
- **WIDTH** — возвращается ширина столбца, на который ссылаются. Единица измерения — количество нулей (0), которые вписываются в столбец шрифтом и размером шрифта по умолчанию;
- **PREFIX** — возвращается текст, который обозначает выравнивание ячейки, на которую ссылаются, в виде:

' = выравнивание по левому краю или по ширине;

" = выравнивание по правому краю;

^ = выравнивание по центру;

\ = заполнение (в настоящее время не активно).

- **PROTECT** — возвращается статус защиты ячейки:

1 = ячейка защищена;

0 = ячейка не защищена.

- **FORMAT** — возвращается текст, который обозначает формат числа:

, = число с разделителем тысяч;

F = число без разделителя тысяч;

C = денежный формат;

S = экспоненциальное представление, например 1,234+E56;

P = процентный формат;

В вышеупомянутых форматах, число десятичных разрядов после десятичного разделителя дается в виде числа. Например: формат числа #,##0.0 возвращает ,1, а формат числа 00.000% возвращает **P3**;

D1 = MMM-D-YY, MM-D-YY и подобные форматы;

D2 = DD-MM;

D3 = MM-YY;

D4 = DD-MM-YYYY HH:MM:SS;

D5 = MM-DD;

D6 = HH:MM:SS AM/PM;

D7 = HH:MM AM/PM;

D8 = HH:MM:SS;

D9 = HH:MM;

G = все другие форматы;

- (минус) в конце = отрицательные числа выделяются цветом;

() (скобки) в конце = в коде формата есть открывающая скобка.

- **COLOR** — возвращается 1, если отрицательные значения выделяются цветом, в противном случае 0.

- **PARENTHESSES** — возвращается 1, если код формата содержит открывающую скобку (, в противном случае 0.

	A	B	C	D	E
1		Информация о ячейке			
2					
3			1400		
4					
5		3	=CELL("COL";C3)		
6		3	=CELL("ROW";C3)		
7		1	=CELL("SHEET";C3)		
8		\$C\$3	=CELL("ADDRESS";C3)		
9		file:///E:/cellinfo.ods#Лист1	=CELL("FILENAME";C3)		
10		\$A:\$C\$3	=CELL("COORD";C3)		
11		1400	=CELL("CONTENTS";C3)		
12		v	=CELL("TYPE";C3)		
13		17	=CELL("WIDTH";C3)		
14		1	=CELL("Protect";C3)		
15		0	=CELL("COLOR";C3)		
16		0	=CELL("PARENTHESES";C3)		
17					

Функция **CELL** возвращает захваченную информацию, когда целевая ячейка была последний раз обновлена. Например **CELL("WIDTH"; D2)**, возвращает ширину столбца D — но если столбец D сделать шире, возвращаемое значение не обновится, пока содержимое **D2** не будет изменено. Чтобы гарантировать обновление, используйте (например) **CELL("WIDTH"; D2) + 0*RAND()**; это работает, потому что **RAND()** обновляется, когда изменяется ширина столбца.

CURRENT

Функция **CURRENT** возвращает текущий результат (до текущего момента) вычисления формулы.

Синтаксис функции:

=CURRENT()

Эта функция возвращает результат вычисления формулы, частью которой она является (другими словами результат, который имеется при вычислении на момент вызова функции). Ее главное использование — совместно с функцией **STYLE** для применения выбранных стилей к ячейке в зависимости от содержимого ячейки.

	A	B	C	D	E	F	G
1							
2	6	6	=1 + 2 + CURRENT()				
3							
4		12	=A2 + B2 + STYLE(IF(CURRENT()>10; "Красный"; "Базовый"))				
5							
6		choochoo	= "choo" & CURRENT()				
7							

В ячейке **B2** возвращается 6. Формула вычисляется слева направо как: 1 + 2 равняется 3, давая результат до момента столкновения с функцией **CURRENT()**; **CURRENT()** поэтому возвращает 3, которые добавляются к исходным 3, давая 6.

В ячейке **B4** возвращается **A2 + B2** (**STYLE** возвращает 0 в этом месте). Если эта сумма больше чем 10, к ячейке применяется стиль **Красный**. Смотри функцию **STYLE** для получения дополнительной информации.

В ячейке **B6** возвращается **choochoo**.

Эта функция лучше всего используется только в простых ситуациях. Более сложный пример:

$1+2*\text{CURRENT}()$ возвращает 5, потому что, хотя вычисляемая формула при столкновении имеет значение 1, оно игнорируется, в то время как выполняется более важное умножение. Таким образом, **CURRENT()** в этом случае дает 2, следовательно результат: $1 + 2 * 2$ равняется 5. Однако на точный порядок вычислений в более сложных случаях нельзя положиться.

FORMULA

Функция **FORMULA** возвращает формулу, используемую в ячейке в виде текстовой строки.

Синтаксис функции:

=FORMULA(cell)

где:

- **cell** — ссылка на ячейку.

Функция **FORMULA** возвращает формулу, введенную в ячейку **cell** в виде текстовой строки. Эта функция предназначена для помощи при отладке, упрощая отображение формул в других ячейках. Ошибочные результаты в формулах переданных ячеек не передаются.

ISBLANK

Функция **ISBLANK** проверяет, пустая ли ячейка.

Синтаксис функции:

=ISBLANK(value)

где:

- **value** — проверяемое значение.

Если **value** — число, текст, или логическое значение, возвращается **FALSE**. Если **value** — ссылка на ячейку, проверяется ячейка; если она пустая (не содержит значения), возвращается **TRUE**, но если она содержит значение, возвращается **FALSE**. Ячейка с пустой строкой не считается пустой.

	А	В	С	Д
1				
2		ИСТИНА	=ISBLANK(A2)	
3		ЛОЖЬ	=ISBLANK(A3)	
4				

ISERR

Функция **ISERR** выполняет проверку на значение ошибки кроме **#N/A**.

Синтаксис функции:

=ISERR(value)

где:

- **value** — проверяемое значение.

Функция **ISERR** возвращает **TRUE**, если **value** - ссылается на или оценивается как значение ошибки, кроме ошибки **#N/A** (не доступно) и **FALSE** в противном случае.

Используйте функцию **ISERROR**, чтобы выполнить проверку на любые ошибки, включая ошибку **#N/A**.

	A	B	C	D
1				
2		ИСТИНА	=ISERR(SQRT(-1))	
3				
4	123	ЛОЖЬ	=ISERR(A4)	
5				
6		ЛОЖЬ	=ISERR(NA())	
7				

В ячейке **B2** возвращается **TRUE**, потому что извлечение квадратного корня из -1 — возвращает ошибку.

В ячейке **B4** возвращается **FALSE**, потому что в ячейке **A4** содержится 123, что 123 не является ошибкой.

В ячейке **B6** возвращается **FALSE**, потому что значение ошибки **#N/A** игнорируется этой функцией.

В приведенной ниже таблице мы видим два пути использования функции **SEARCH** в электронной таблице. Если **SEARCH** не выдает результат, а получение в ячейке **#VALUE!** — возможно не приемлемо с эстетической точки зрения. Однако, используя функцию **ISERR**, мы можем отфильтровать записи **#VALUE!** - как проиллюстрировано в примере.

	A	B	C	D	E	F	G	H
1			Пример ISERR					
2								
3								
4	Фрэд		3	=SEARCH("Эд";B4)				
5	Поль		#ЗНАЧЕН!	=SEARCH("Эд";B5)				
6	Джозеф		#ЗНАЧЕН!	=SEARCH("Эд";B6)				
7	Эдуард		1	=SEARCH("Эд";B7)				
8	Джимми		#ЗНАЧЕН!	=SEARCH("Эд";B8)				
9	Келли		#ЗНАЧЕН!	=SEARCH("Эд";B9)				
10								
11								
12	Фрэд		3	=IF(ISERR(SEARCH("Эд";B12));"";SEARCH("Эд";B12))				
13	Поль			=IF(ISERR(SEARCH("Эд";B13));"";SEARCH("Эд";B13))				
14	Джозеф			=IF(ISERR(SEARCH("Эд";B14));"";SEARCH("Эд";B14))				
15	Эдуард		1	=IF(ISERR(SEARCH("Эд";B15));"";SEARCH("Эд";B15))				
16	Джимми			=IF(ISERR(SEARCH("Эд";B16));"";SEARCH("Эд";B16))				
17	Келли			=IF(ISERR(SEARCH("Эд";B17));"";SEARCH("Эд";B17))				
18								

ISERROR

Функция **ISERROR** выполняет проверку на любое значение ошибки.

Синтаксис функции:

=ISERROR(value)

где:

- **value** — проверяемое значение.

Функция **ISERROR** возвращает **TRUE**, если **value** ссылается на или оценивается как значение ошибки, включая **#N/A**, и **FALSE** в противном случае.

Используйте функцию **ISERR**, чтобы выполнить проверку на любые ошибки кроме **#N/A**.

	A	B	C	D
1				
2		ИСТИНА	=ISERROR(SQRT(-1))	
3				
4	123	ЛОЖЬ	=ISERROR(A4)	
5				
6		ИСТИНА	=ISERROR(NA())	
7				

В ячейке **B2** возвращается **TRUE**, потому что извлечение квадратного корня из -1 — возвращает ошибку.

В ячейке **B4** возвращается **FALSE**, потому что в ячейке **A4** содержится 123, что 123 не является ошибкой.

В ячейке **B6** возвращается **TRUE**, потому что **NA()** возвращает ошибку **#N/A**.

CHOOSE(0;"A";"B";"C";"D") возвращает **Err:502** (это правильно, так как индекс выходит за диапазон), но **ISERROR(CHOOSE(0;"A";"B";"C";"D"))** также возвращает **Err:502**, а не **TRUE**. Ошибка передается, а не оценивается.

ISEVEN

Функция **ISEVEN** возвращает **TRUE**, если значение — четное число, или **FALSE**, если значение — нечетное.

Синтаксис функции:

=ISEVEN(value)

где:

- **value** — проверяемое значение.

Если **value** не целое число, любые цифры после десятичного разделителя игнорируются. Знак **value** также игнорируется.

	A	B	C	D
1				
2		ИСТИНА	=ISEVEN(48)	
3				
4		ЛОЖЬ	=ISEVEN(33)	
5				
6		ИСТИНА	=ISEVEN(0)	
7				
8		ЛОЖЬ	=ISEVEN(3,999)	
9				
10		ИСТИНА	=ISEVEN(-2,1)	
11				

В ячейке **C8** возвращается **ЛОЖЬ**. Игнорирование цифр после десятичного разделителя дает 3, которое не четно.

В ячейке **C10** возвращается **ИСТИНА**. Игнорирование знака и цифр после десятичного разделителя дает 2, которое является четным.

ISEVEN_ADD

Функция **ISEVEN_ADD** возвращает 1, если значение — четное число, или 0, если значение — нечетное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=ISEVEN_ADD(value)

где:

- **value** — проверяемое значение.

Если **value** не целое число, любые цифры после десятичного разделителя игнорируются. Знак **value** также игнорируется.

	A	B	C	D	E
1					
2			1	=ISEVEN_ADD(48)	
3					
4			0	=ISEVEN_ADD(33)	
5					
6			1	=ISEVEN_ADD(0)	
7					
8			0	=ISEVEN_ADD(3,999)	
9					
10			1	=ISEVEN_ADD(-2,1)	
11					

В ячейке **C8** возвращается 0. Игнорирование цифр после десятичного разделителя дает 3, которое нечетно.

В ячейке **C10** возвращается 1. Игнорирование знака и цифр после десятичного разделителя дает 2, которое является четным.

Функции **ISEVEN_ADD** и **ISODD_ADD** предназначены для того, чтобы подражать определенному нестандартному поведению в Excel.

ISFORMULA

Функция **ISFORMULA** проверяет содержит ли ячейка формулу.

Синтаксис функции:

=ISFORMULA(cell)

где:

- **cell** — ссылка на ячейку.

Функция **ISFORMULA** возвращает **TRUE**, если **cell** ссылается на ячейку, значение которой вычисляется формулой и **FALSE** в противном случае. Заметьте, что сама формула может просто вычислять константу; в этом случае функция будет возвращать **TRUE**, так как это по-прежнему формула.

	A	B	C	D	E
1					
2		2 =SQRT(4)	ИСТИНА	=ISFORMULA(A2)	
3					
4		2	ЛОЖЬ	=ISFORMULA(A4)	
5					

В ячейке **C2** возвращается **TRUE**, так как ячейка **A2** содержит **=SQRT(4)**.

В ячейке **C4** возвращается **FALSE**, так как ячейка **A4** содержит 2.

ISLOGICAL

Функция **ISLOGICAL** проверяет, содержит ли ячейка логическое значение, **TRUE** или

FALSE.

Синтаксис функции:

=ISLOGICAL(value)

где:

- **value** — проверяемое значение.

Функция **ISLOGICAL** возвращает **TRUE**, если **value** одно из двух — **TRUE** или **FALSE**, и **FALSE** в противном случае.

	A	B	C	D
1				
2	ЛОЖЬ	ИСТИНА	=ISLOGICAL(A2)	
3				
4	123	ЛОЖЬ	=ISLOGICAL(A4)	
5				

В ячейке **C2** возвращается **TRUE**, так как ячейка **A2** содержит **FALSE**.

В ячейке **C4** возвращается **FALSE**, так как ячейка **A4** содержит 123.

Логическое значение в Calc — просто число с логическим форматом, а не отдельный тип как в Excel.

ISNA

Функция **ISNA** выполняет проверку на значение ошибки **#N/A** (не доступно).

Синтаксис функции:

=ISNA(value)

где:

- **value** — проверяемое значение.

Функция **ISNA** возвращает **TRUE**, если **value** является или ссылается на ошибку **#N/A** (не доступно) и **FALSE** в противном случае. Заметьте, что, если **value** — ссылка, рассматривается значение на которое указывает эта ссылка. Эта функция *не вызывает* размножение значения ошибки.

	A	B	C	D
1				
2		ИСТИНА	=ISNA(NA())	
3				
4		ЛОЖЬ	=ISNA(1/0)	
5				
6	Собака	ЛОЖЬ	=ISNA(A6)	
7				

В ячейке **C2** возвращается **TRUE**, потому что **NA()** возвращает ошибку **#N/A**.

В ячейке **C4** возвращается **FALSE**, так как **1/0** вызывает ошибку **#DIV/0!**, а не ошибку **#N/A**.

В ячейке **C6** возвращается **FALSE**, так как ячейка **A6** содержит текст “Собака”, а не ошибку.

ISNONTTEXT

Функция **ISNONTTEXT** выполняет проверку что ячейка не содержит текст.

Синтаксис функции:

=ISNONTTEXT(value)

где:

- **value** — проверяемое значение.

Функция **ISNONTEXT** возвращает **TRUE**, если **value** не является или не ссылается на текст, а в этом случае она возвращает **FALSE**. Ссылки на пустые ячейки не считаются текстом, таким образом ссылка на пустую ячейку будет возвращать **TRUE**.

ISNONTEXT(value) эквивалентно **NOT(ISTEXT(value))**.

	A	B	C	D
1				
2		ИСТИНА	=ISNONTEXT(123)	
3				
4	Кошка	ЛОЖЬ	=ISNONTEXT(A4)	
5				
6	#DIV/0!	ИСТИНА	=ISNONTEXT(A6)	
7				
8		ИСТИНА	=ISNONTEXT(A8)	
9				

В ячейке **C2** возвращается **TRUE**, потому что 123 это число, а не текст

В ячейке **C4** возвращается **FALSE**, так как ячейка **A4** содержит текст “Кошка”.

В ячейке **C6** возвращается **TRUE**, так как ячейка **A6** содержит =1/0, вызывающее ошибку, а не текст.

В ячейке **C8** возвращается **TRUE**, так как ячейка **A8** пустая (пустая ячейка не содержит текст).

ISNUMBER

Функция **ISNUMBER** выполняет проверку содержит ли ячейка число.

Синтаксис функции:

=ISNUMBER(value)

где:

- **value** — проверяемое значение.

Функция **ISNUMBER** возвращает **TRUE**, если **value** — число или логическое значение и **FALSE** в противном случае.

Функция **ISNUMBER** в настоящее время (OOo2.3) считает логические значения **TRUE** и **FALSE** числами.

	A	B	C	D
1				
2		ИСТИНА	=ISNUMBER(123)	
3				
4	Кошка	ЛОЖЬ	=ISNUMBER(A4)	
5				
6	99	ИСТИНА	=ISNUMBER(A6)	
7				

ISNUMBER(A1), когда ячейка **A1** содержит логическое значение, возвращает **TRUE**. Это поведение не совместимо с другими конкурирующими электронными таблицами.

ISODD

Функция **ISODD** возвращает **TRUE**, если значение — нечетное число, или **FALSE**, если

значение — четное.

Синтаксис функции:

=ISODD(value)

где:

- **value** — проверяемое значение.

Если **value** не целое число, любые цифры после десятичного разделителя игнорируются. Знак **value** также игнорируется.

	A	B	C	D
1				
2		ИСТИНА	=ISODD(33)	
3				
4		ЛОЖЬ	=ISODD(48)	
5				
6		ИСТИНА	=ISODD(3,999)	
7				
8		ИСТИНА	=ISODD(-3,1)	
9				

В ячейке **C6** возвращается **TRUE** (ИСТИНА). Игнорирование цифр после десятичного разделителя дает 3, которое нечетно.

В ячейке **C8** возвращается **TRUE** (ИСТИНА). Игнорирование знака и цифр после десятичного разделителя дает 3, которое нечетно.

ISODD_ADD

Функция **ISODD_ADD** возвращает 1, если значение — нечетное число, или 0, если значение — четное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=ISODD_ADD(value)

где:

- **value** — проверяемое значение.

Если **value** не целое число, любые цифры после десятичного разделителя игнорируются. Знак **value** также игнорируется.

	A	B	C	D	E
1					
2		1	=ISODD_ADD(33)		
3					
4		0	=ISODD_ADD(48)		
5					
6		1	=ISODD_ADD(3,999)		
7					
8		1	=ISODD_ADD(-3,1)		
9					

В ячейке **C6** возвращается 1. Игнорирование цифр после десятичного разделителя дает 3, которое нечетно.

В ячейке **C8** возвращается 1. Игнорирование знака и цифр после десятичного разделителя дает 3, которое нечетно.

Функции **ISEVEN_ADD** и **ISODD_ADD** предназначены для того, чтобы подражать определенному нестандартному поведению в Excel.

ISREF

Функция **ISREF** выполняет проверку, что аргумент — ссылка на ячейку или диапазон ячеек.

Синтаксис функции:

=ISREF(value)

где:

- **value** — проверяемое значение.

Функция **ISREF** возвращает **TRUE**, если **value** — ссылка на ячейку или диапазон ячеек и **FALSE** в противном случае. Заметьте, что в отличие от почти всех других функций, когда задана ссылка эта функция не исследует ссылаемое значение. Некоторые функции и операторы возвращают ссылки, и таким образом **ISREF** возвратит **TRUE** когда получит их результат. **value** может быть таблица ссылок, в этом случае **ISREF** возвратит **TRUE**. Ошибки размножаются этой функцией, поэтому если **value** — ошибка, **ISREF** возвратит ошибку.

	A	B	C	D	E
1					
2		ИСТИНА	=ISREF(C5)		
3					
4		ИСТИНА	=ISREF(A1:B3)		
5					
6		ЛОЖЬ	=ISREF("A5")		
7					
8		ЛОЖЬ	=ISREF(NA())		
9					
10		ИСТИНА	=ISREF(range1)		
11					
12	D1	ИСТИНА	=ISREF(INDIRECT(A12))		
13					

В ячейке **B2** возвращается **TRUE**, потому что **C5** это ссылка на ячейку.

В ячейке **B4** возвращается **TRUE**, так как ячейка **A1:B3** ссылка на диапазон ячеек.

В ячейке **B6** возвращается **FALSE**, так как “**A5**” текст, а не ссылка на ячейку.

В ячейке **B8** возвращается **FALSE**, так как ячейка **NA()** возвращает ошибку, а не ссылку на ячейку.

В ячейке **B10** возвращается **TRUE**, так как **range1** — именованный диапазон **A1:B3**.

В ячейке **B12** возвращается **TRUE**, так как **A12** содержит текст “**D1**”, а функция **INDIRECT** возвращает ссылку.

Стандарт ODFD не ясно определяет, что должна вернуть функция **ISREF** когда представлен неопределенный диапазон.

ISTEXT

Функция **ISTEXT** выполняет проверку, что ячейка содержит текст.

Синтаксис функции:

=ISTEXT(value)

где:

- **value** — проверяемое значение.

Функция **ISTEXT** возвращает **TRUE**, если **value** является или ссылается на текст и **FALSE** в противном случае.

	А	В	С	Д
1				
2		ИСТИНА	=ISTEXT("собака")	
3				
4	Кошка	ИСТИНА	=ISTEXT(A4)	
5				
6	#Н/Д	ЛОЖЬ	=ISTEXT(A6)	
7				
8		ЛОЖЬ	=ISTEXT(A8)	
9				

В ячейке **B2** возвращается **TRUE**, потому что аргумент это текст.

В ячейке **B4** возвращается **TRUE**, так как ячейка **A4** содержит текстовое значение “Кошка”.

В ячейке **B6** возвращается **FALSE**, так как ячейка **A6** содержит =NA(), возвращающее ошибку.

В ячейке **B8** возвращается **FALSE**, так как ячейка **A8** пустая (пустая ячейка не содержит текст).

N

Функция **N** возвращает числовое значение.

Синтаксис функции:

=N(value)

где:

- **value** — проверяемое значение.

Функция **N** возвращает числовое значение **value** если возможно. Если **value** ссылка, сначала выполняется разыменовывание ссылки. Потом проверяется тип значения. Если оно имеет числовой тип, возвращается число. Если оно имеет логический тип, возвращается 1 для **TRUE** и 0 для **FALSE**.

Функция **N()** не предназначена для преобразования текста в число, используйте для этого функцию **VALUE()**.

	А	В	С	Д
1				
2		123	=N(123)	
3				
4	ИСТИНА	1	=N(A4)	
5				
6	ЛОЖЬ	0	=N(A6)	
7				

В ячейке **B2** возвращается 123, потому что 123 — (уже) число.

В ячейке **B4** возвращается 1, так как ячейка **A4** содержит логическое значение **TRUE**.

В ячейке **B6** возвращается 0, так как ячейка **A6** содержит логическое значение **FALSE**.

N() в Calc преобразует текст в число, но в Excel это не будет. Стандарт ODFP преднамеренно не определяет ее поведение. Используйте **VALUE()** для преобразования текстового значения в числовое.

NA

Функция **NA** возвращает значение ошибки #N/A (не доступно).

Синтаксис функции:

=NA()

Эта функция не имеет аргументов, и возвращает значение ошибки #N/A.

	A	B	C	D
1				
2		#N/A		
3				

TYPE

Функция **TYPE** возвращает тип значения (число, текст, и т.д.).

Синтаксис функции:

=TYPE(value)

где:

- **value** — проверяемое значение.

Функция **TYPE** возвращает тип значения в виде числа: **1** = число, **2** = текст, **4** = логическое значение, **8** = формула, **16** = значение ошибки.

Если ячейка содержит ошибку и формулу, возвращается **16** (значение ошибки).

Пустая ячейка классифицируется как содержащая числовое значение, и возвращается **1**.

	A	B	C	D
1				
2	Собака	2		
3				
4	#N/A	16		
5				
6		1		
7				

В ячейке **B2** возвращается 2, потому что ячейка **A2** содержит текстовое значение “Собака”.

В ячейке **B4** возвращается 16, так как ячейка **A4** содержит =NA(), возвращающее значение ошибки.

В ячейке **B6** возвращается 1, так как ячейка **A6** пустая.

Логические функции

Логические функции оперируют логическими (“булевыми”) значениями, то есть **TRUE** или **FALSE**. Спецификация OpenDocument упоминает “логические операторы”; это просто другое название для логических функций.

Это не поразрядные операции, например, **AND(12;10)**, возвращает **TRUE**, а не 8. См. функции поразрядных операций для ознакомления с поразрядными операциями.

Краткий обзор логических значений

В Calc, логические значения представлены числами: 0 — **FALSE**, и 1 — **TRUE**. Можно ввести в ячейку 1, и затем (выполнив **Формат — Ячейки...**) выбрать формат “Логический”, чтобы отобразить число в виде **ИСТИНА (TRUE)**.

Любая функция Calc, которая возвращает логический результат, фактически возвращает число 0 или 1. Любая формула ячейки Calc, которая возвращает логический результат, говорит ячейке показывать **ЛОЖЬ (FALSE)** или **ИСТИНА (TRUE)** — если ячейка имеет формат по умолчанию (общий). Например **=TRUE()** возвращает значение 1, которое отображается как **ИСТИНА**; если Вы измените формат на числовой, она отобразится как 1.

Любая функция Calc, которая выполняет проверку на логический результат, фактически проверяет, что значение оценивается как число 0. 0 воспринимается как **FALSE**, а что-либо еще воспринимается как **TRUE**. Например, когда ячейка **A1** содержит “яблоко”, **NOT(A1)** возвращается **TRUE**, потому что “яблоко” оценивается как 0 = **FALSE**. Другой пример: **NOT(57)** возвращает **FALSE**, потому что 57 оценивается как **TRUE**.

Excel в противоположность имеет отдельный тип для логических значений — они не числа, но иногда преобразуются к числам. Поэтому, если Вы нуждаетесь в совместимости:

- В Excel функции подобные **SUM** и **AVERAGE** игнорируют логические значения; в Calc они — числа и поэтому учитываются. Это вряд ли вызовет трудности, потому что Вы обычно не предполагаете складывать логические значения.
- В Calc **SUMPRODUCT(A1:A6=“красный”; B1:B6=“большой”)**, подсчитывает число больших красных элементов. В Excel логические результаты, возвращаемые **A1:A6=“красный”** и **B1:B6=“большой”** игнорируются функцией **SUMPRODUCT**; логические значения преобразуются, если ожидается число, таким образом

SUMPRODUCT(1*(A1:A6=“красный”); 1*(B1:B6=“большой”)),

SUMPRODUCT(0+(A1:A6=“красный”); 0+(B1:B6=“большой”)),

SUMPRODUCT(-(A1:A6=“красный”); -(B1:B6=“большой”)) или

SUMPRODUCT((A1:A6=“красный”)*(B1:B6=“большой”)) все работают в обеих электронных таблицах.

AND

Функция **AND** возвращает **TRUE**, если все аргументы определяются как **TRUE**, и **FALSE** в противном случае.

Синтаксис функции:

=AND(argument1; argument2 ...argument30)

где:

- **argument1; argument2 ...argument30** — до 30 аргументов, каждый из которых может быть логическим результатом или значением, ссылкой на ячейку или диапазон.

Функция **AND** проверяет каждое значение (как аргумент, или в каждой ячейке, на которую ссылаются), и возвращает **TRUE**, если они все **TRUE**. Любое значение, которое является **числом отличным от нуля** или **текстом**, как полагают, является **TRUE**.

	A	B	C	D	E	F
1						
2		ЛОЖЬ	ИСТИНА	=TRUE()		
3			Собака			
4						
5	ИСТИНА	ИСТИНА		ИСТИНА	=AND(2<4;A5:B8;C2)	
6	ИСТИНА	ИСТИНА		ЛОЖЬ	=AND(2<4;B2)	
7	ИСТИНА	ИСТИНА		ИСТИНА	=AND(C2:C3)	
8	ИСТИНА	ИСТИНА				
9						

FALSE

Функция **FALSE** возвращает логическое значение **FALSE**.

Синтаксис функции:

=FALSE()

Функция **FALSE** не имеет аргументов, и всегда возвращает логическое значение **FALSE**.

Синтаксически это вызов функции, но семантически — константа.

	A	B	C	D
1				
2		ЛОЖЬ	=FALSE()	
3				
4		ИСТИНА	=NOT(FALSE())	
5				

Логическое значение в Calc — просто число с логическим форматом, а не отдельный тип как в Excel.

IF

Функция **IF** возвращает одно из двух значений, в зависимости от результатов проверки условия.

Синтаксис функции:

=IF(test; value1; value2)

где:

- **test** — логическое значение или выражение (или ссылка на логическое значение или выражение), которое возвращает логическое значение (**TRUE** или **FALSE**).
- **value1** — значение, которое возвращается функцией если результат проверки **TRUE**.
- **value2** — значение, которое возвращается функцией если результат проверки **FALSE**.

Эта функция вычисляет только **value1** или **value2**, и никогда оба выражения. Если **value2** опущено, оно, как предполагается, равняется **FALSE**; если **value1** также опущено, оно, как предполагается, является **TRUE**.

	A	B	C	D	E
1					
2		too small	=IF(A1>5; 100; "too small")		
3					
4		ЛОЖЬ	=IF(1>2; "nonsense")		
5					
6		ИСТИНА	=IF(2>1)		
7					
8		2	=IF(1=2; 1/0; SQRT(4))		
9					

В ячейке **B2** возвращается число 100, если **A1** больше чем 5, и текст “too small” в противном случае.

В ячейке **B4** возвращается **FALSE**, потому что **value2** было опущено, а 1 не больше 2.

В ячейке **B6** возвращается **TRUE**, потому что и **value1** и **value2** были опущены, а 2 больше чем 1.

В ячейке **B8** возвращается 2, квадратный корень 4. **IF()** вычисляет только выбранное значение — в этом случае выражение 1/0 дало бы ошибку **#DIV/0!**, но оно не вычисляется.

NOT

Меняет на противоположное логическое значение своего аргумента. Возвращается **TRUE**, если аргумент **FALSE**, и **FALSE**, если аргумент **TRUE**.

Синтаксис функции:

=NOT(logical_value)

где:

- logical_value** — логическое значение, которое будет изменено на противоположное.

	A	B	C	D
1				
2		ЛОЖЬ	=NOT(TRUE())	
3				

Введенная в ячейке формула **=NOT(ВЕРНЫЙ)** правильно возвращает **FALSE**, но отображается в строке ввода формулы как **=NOT(1)**.

OR

Функция **OR** возвращает **TRUE**, если любой из аргументов определяется как **TRUE**, и **FALSE** в противном случае.

Синтаксис функции:

=OR(argument1; argument2 ...argument30)

где:

- argument1; argument2 ...argument30** — до 30 аргументов, каждый из которых может быть логическим результатом или значением, ссылкой на ячейку или диапазон.

Функция **OR** проверяет каждое значение (как аргумент, или в каждой ячейке, на которую ссылаются), и возвращает **TRUE**, если любой из них **TRUE**. Любое число отличное от нуля, как полагают, соответствует **TRUE**. Любые текстовые ячейки в диапазонах игнорируются.

	A	B	C	D	E	F	G	H
1								
2		ИСТИНА	=OR(TRUE(); FALSE())			ИСТИНА	=TRUE()	
3								
4		ИСТИНА	=OR(0; 5)					
5						ЛОЖЬ	ЛОЖЬ	
6		ИСТИНА	=OR(1>2; F5:G8; F2)			ЛОЖЬ	ЛОЖЬ	
7						ЛОЖЬ	ЛОЖЬ	
8						ЛОЖЬ	ЛОЖЬ	
9								

В ячейке **B2** возвращается **TRUE**.

В ячейке **B4** возвращается **TRUE**, потому что 5 воспринимается как **TRUE**.

В ячейке **B6** возвращается **TRUE**, потому что ячейка **F2** содержит **TRUE**.

TRUE

Функция **TRUE** возвращает логическое значение **TRUE**.

Синтаксис функции:

=TRUE()

Функция **TRUE()** не имеет аргументов, и всегда возвращает логическое значение **TRUE**.

	A	B	C	D
1				
2		ИСТИНА	=TRUE()	
3				
4		ЛОЖЬ	=NOT(TRUE())	
5				

Логическое значение в Calc — просто число с логическим форматом, а не отдельный тип как в Excel.

Математические функции

Математические функции включают арифметические, тригонометрические, гиперболические и логарифмические функции.

ABS

Функция **ABS** возвращает абсолютное значение числа.

Синтаксис функции:

=ABS(number)

где:

- **number** — число, абсолютное значение которого должно быть вычислено.

Абсолютное значение числа — его значение без знака. Если **number** < 0, возвращается **-number**, в противном случае возвращается **number**.

	A	B	C	D
1				
2		56	=ABS(-56)	
3				
4		12,3	=ABS(12,3)	
5				
6		0	=ABS(0)	
7				

ACOS

Функция **ACOS** возвращает обратный косинус (арккосинус) числа.

Синтаксис функции:

=ACOS(number)

где:

- **number** — число, косинус искомого угла, значение должно находиться в диапазоне от -1 до 1.

Функция **ACOS** возвращает обратный тригонометрический косинус **number**, другими словами угол (в радианах), чей косинус — **number**. Возвращаемый угол — от 0 до π .

Чтобы возвращать угол в градусах, умножьте результат на **180/PI()** или используйте функцию **DEGREES**.

	A	B	C	D
1				
2		3,14159265358979	=ACOS(-1)	
3				
4		60	=DEGREES(ACOS(0,5))	
5				

В ячейке **B2** возвращается 3,14159265358979 (π радиан). В ячейке **B4** возвращается 60, косинус 60 градусов — 0,5.

ACOSH

Функция **ACOSH** возвращает обратный гиперболический косинус (гиперболический арккосинус) числа.

Синтаксис функции:

=ACOSH(number)

где:

- **number** — действительное число, для которого требуется найти гиперболический арккосинус, число должно быть больше или равно 1.

Функция **ACOSH** вычисляет гиперболический арккосинус. Гиперболический арккосинус — аналог обычного (тригонометрического) арккосинуса. Формула для вычисления гиперболического арккосинуса имеет следующий вид:

$$\operatorname{acosh}(N) = \ln(N \pm \sqrt{N^2 - 1})$$

	A	B	C	D
1				
2		0	=ACOSH(1)	
3				
4		4	=ACOSH(COSH(4))	
5				

В ячейке **B2** возвращается 0. В ячейке **B4** возвращается 4.

ACOT

Функция **ACOT** возвращает обратный котангенс (арккотангенс) заданного числа. Арккотангенс числа — это угол, котангенс которого равен заданному *числу*.

Синтаксис функции:

=ACOT(number)

где:

- **number** — число, котангенс искомого угла.

Функция **ACOT** возвращает обратный тригонометрический котангенс **number**, другими словами угол (в радианах), чей котангенс — **number**. Возвращаемый угол — от 0 до π .

Чтобы возвращать угол в градусах, умножьте результат на **180/PI()** или используйте функцию **DEGREES**.

	A	B	C	D
1				
2		0,785398163397448	=ACOT(1)	
3				
4		45	=DEGREES(ACOT(1))	
5				

В ячейке **B2** возвращается 0,785398163397448 ($\pi/4$ радиан). В ячейке **B4** возвращается 45, котангенс 45 градусов — 1.

ACOTH

Функция **ACOTH** возвращает обратный гиперболический котангенс (гиперболический арккотангенс) заданного числа.

Синтаксис функции:

=ACOTH(number)

где:

- **number** — действительное число, для которого требуется найти гиперболический аркосинус, число должно быть больше 1. Возвращается ошибка, если значение **number** — между -1 и 1 включительно.

Вычисляет гиперболический аркотангенс. Гиперболический аркотангенс — аналог обычного (тригонометрического) аркотангенса. Формула для вычисления гиперболического аркотангенса имеет следующий вид:

$$\operatorname{acoth}(N) = \frac{1}{2} \ln\left(\frac{x+1}{x-1}\right)$$

	A	B	C	D
1				
2		1,52226	=ACOTH(1,1)	
3				

В ячейке **B2** возвращается гиперболический аркотангенс 1,1, приблизительно 1,52226.

ASIN

Функция **ASIN** возвращает обратный синус (арксинус) числа.

Синтаксис функции:

=ASIN(number)

где:

- **number** — число, синус искомого угла, значение должно находиться в диапазоне от -1 до 1.

Функция **ASIN** возвращает обратный тригонометрический синус **number**, другими словами угол (в радианах), чей синус — **number**. Возвращаемый угол — от $-\pi/2$ до $+\pi/2$.

Чтобы возвращать угол в градусах, умножьте результат на **180/PI()** или используйте функцию **DEGREES**.

	A	B	C	D
1				
2		0	=ASIN(0)	
3				
4		1,5707963267949	=ASIN(1)	
5				
6		30	=DEGREES(ASIN(0,5))	
7				

В ячейке **B2** возвращается 0. В ячейке **B4** возвращается 1,5707963267949 ($\pi/2$ радиан). В ячейке **B6** возвращается 30, синус 30 градусов — 0,5.

ASINH

Функция **ASINH** возвращает обратный гиперболический синус (гиперболический арксинус) заданного числа.

Синтаксис функции:

=ASINH(number)

где:

- **number** — любое действительное число, для которого требуется найти

гиперболический арксинус.

Функция **ASINH** вычисляет гиперболический арксинус. Гиперболический арксинус — аналог обычного (тригонометрического) арксинуса. Формула для вычисления гиперболического аркотангенса имеет следующий вид:

$$\operatorname{asinh}(N) = \ln(N + \sqrt{N^2 + 1})$$

	A	B	C	D
1				
2		-5,1929877	=ASINH(90)	
3				
4		4	=ASINH(SINH(4))	
5				

В ячейке **B2** возвращается приблизительно -5.1929877 . В ячейке **B4** возвращается 4.

ATAN

Функция **ATAN** возвращает обратный тангенс (арктангенс) числа.

Синтаксис функции:

=ATAN(number)

где:

- **number** — число, тангенс искомого угла.

Функция **ATAN** возвращает обратный тригонометрический тангенс **number**, другими словами угол (в радианах), чей тангенс — **number**. Возвращаемый угол — от $-\pi/2$ до $+\pi/2$.

Чтобы возвращать угол в градусах, умножьте результат на **180/PI()** или используйте функцию **DEGREES**.

	A	B	C	D
1				
2		0,785398163397448	=ATAN(1)	
3				
4		45	=DEGREES(ATAN(1))	
5				

В ячейке **B2** возвращается 0,785398163397448 ($\pi/4$ радиан). В ячейке **B4** возвращается 45, тангенс 45 градусов — 1.

ATAN2

Функция **ATAN2** возвращает обратный тангенс (арктангенс) для указанных координат x и y.

Синтаксис функции:

=ATAN2(x_coord; y_coord)

где:

- **x_coord** — значение координаты x точки;
- **y_coord** — значение координаты y точки.

Функция **ATAN2** возвращает обратный тригонометрический тангенс, то есть, угол (в радианах) между осью X и линией от точки **x_coord**, **y_coord** к началу координат. Возвращаемый угол — от $-\pi$ до $+\pi$.

Чтобы возвращать угол в градусах, умножьте результат на **180/PI()** или используйте функцию **DEGREES**.

	A	B	C	D	E
1					
2		0,785398163397448	=ATAN2(20; 20)		
3					
4		45	=DEGREES(ATAN2(12,3; 12,3))		
5					

В ячейке **B2** возвращается 0,785398163397448 ($\pi/4$ радиан). В ячейке **B4** возвращается 45, тангенс 45 градусов — 1.

ATANH

Функция **ATANH** возвращает обратный гиперболический тангенс (гиперболический арктангенс) числа.

Синтаксис функции:

=**ATANH**(number)

где:

- **number** — действительное число, для которого требуется найти гиперболический арктангенс, число должно удовлетворять условию $-1 < \text{number} < 1$.

Вычисляет гиперболический арктангенс. Гиперболический арктангенс — аналог обычного (тригонометрического) арктангенса. Формула для вычисления гиперболического арктангенса имеет следующий вид:

$$\operatorname{atanh}(N) = \frac{1}{2} \ln \left(\frac{1+N}{1-N} \right)$$

	A	B	C	D
1				
2		0,0000000	=ATANH(0)	
3				

В ячейке **B2** возвращается 0.

BESSELI

Функция **BESSELI** вычисляет модифицированную функцию Бесселя первого рода, что эквивалентно вычислению функции Бесселя для чисто мнимого аргумента.

Синтаксис функции:

=**BESSELI**(x; n)

где:

- **x** — значение, для которого вычисляется функция;
- **n** — порядок функции Бесселя.

Функция **BESSELI** возвращает модифицированную функцию Бесселя первого рода, **n**-ого порядка от переменной **x**.

Модифицированная функция Бесселя первого рода это:

$$I_n(x) = (i)^{-n} J_n(ix)$$

где: J_n — функция Бесселя первого рода.

	A	B	C	D
1				
2		0,56515910399234	=BESSELI(1; 1)	
3				

BESSELJ

Функция **BESSELJ** вычисляет функцию Бесселя первого рода.

Синтаксис функции:

=BESSELJ(x; n)

где:

- **x** — значение, для которого вычисляется функция;
- **n** — порядок функции Бесселя.

Функция **BESSELJ** возвращает функцию Бесселя первого рода, **n**-ого порядка от переменной **x**. Функция Бесселя первого рода $J_n(x)$ — решения дифференциального уравнения Бесселя.

	A	B	C	D
1				
2		0,57672480775763	=BESSELJ(2; 1)	
3				

BESSELK

Функция **BESSELK** вычисляет модифицированную функцию Бесселя второго рода, что эквивалентно вычислению функции Бесселя для чисто мнимого аргумента.

Синтаксис функции:

=BESSELK(x; n)

где:

- **x** — значение, для которого вычисляется функция;
- **n** — порядок функции Бесселя.

Функция **BESSELK** возвращает модифицированную функцию Бесселя второго рода, **n**-ого порядка от переменной **x**.

Модифицированные функции Бесселя второго рода (также известные как функции Бассета) часто обозначаются $K_n(x)$.

$$K_n(x) = \frac{\pi}{2} i^{n+1} [J_n(ix) + i Y_n(ix)]$$

	A	B	C	D
1				
2		0,13986588	=BESSELK(2; 1)	
3				

BESSELY

Функция **BESSELY** вычисляет функцию Бесселя второго рода (функцию Неймана или Вебера).

Синтаксис функции:

=BESSELY(x; n)

где:

- **x** — значение, для которого вычисляется функция;
- **n** — порядок функции Бесселя.

Функция **BESSELY** возвращает функцию Бесселя второго рода, **n**-ого порядка от переменной **x**. Функции Бесселя второго рода $Y_n(x)$ (также известные как функции Неймана $N_n(x)$ или Вебера) — решения дифференциального уравнения Бесселя, бесконечные в точке $x = 0$.

$$Y_n(x) = \lim_{\nu \rightarrow n} \frac{J_\nu(x) \cos(\nu \pi) - J_{-\nu}(x)}{\sin(\nu \pi)}$$

	A	B	C	D
1				
2		0,3979257090507	=BESSELY(4; 1)	
3				

CEILING

Функция **CEILING** возвращает число округленное до ближайшего числа, кратного другому числу.

Синтаксис функции:

=CEILING(number; mult; mode)

где:

- **number** — округляемое число;
- **mult** — кратное, до которого требуется округлить.
- **mode** — режим округления. Если **mode** — ноль или опущен, функция **CEILING** выполняет округление к кратному большему (большему или равному) числу. Если **mode** — отличен от нуля, функция **CEILING** выполняет округление в сторону от нуля. Это различие существенно только для отрицательных чисел. Используйте **mode=1** для совместимости, если Вы имеете отрицательные числа и хотите экспортировать в MS Excel. В MS Excel эта функция принимает только два аргумента.

Многие пользовательские приложения имеют функцию **CEILING** только с двумя параметрами, и несколько отличной семантикой чем приведенная здесь (например, они работают, как будто была значение **mode** отлично от нуля). Эти функции **CEILING** несовместимы со стандартным математическим определением **CEILING**. Подобные приложения должны преобразовать такие формулы в формат с 3 параметрами при сохранении в формате OpenFormula, так, чтобы сохраненные формулы работали правильно.

	A	B	C	D	E
1					
2		6	=CEILING(4; 3)		
3					
4		6	=CEILING(6; 3)		
5					
6		-10	=CEILING(-11; -2)		
7					
8		-12	=CEILING(-11; -2; 1)		
9					

В ячейке **B2** возвращается 6, потому что $2 \cdot 3 = 6$ — следующее кратное 3 число большее 4.

В ячейке **B6** возвращается -10, округляется к кратному большему числу.

В ячейке **B8** возвращается -12 , потому что с **mode=1** функция выполняет округление в сторону от нуля.

В Gnumeric, Kspread и Excel, **CEILING(-2.5;-1)** возвращает -3 , в OOo возвращает -2 . Поскольку функция должна выполнить округление, -2 похоже, правильный ответ. Эта функция придерживается странного *округления в сторону от нуля* Excel, только если задается третий параметр.

В Kspread 1.5.0 не проходит контрольный пример **CEILING(X; 0)** и выдает сообщение об ошибке.

COMBIN

Функция **COMBIN** возвращает число комбинаций подмножества элементов.

Синтаксис функции:

=COMBIN(n; k)

где:

- **n** — общее число элементов в наборе;
- **k** — выбранное число элементов из набора.

Функция **COMBIN** возвращает число вариантов выбора этих элементов. Например, если есть 3 элемента в наборе А, В и С, Вы можете выбрать 2 пункта 3 различными способами, а именно, АВ, АС и ВС.

Функция **COMBIN** реализует формулу:

$$\frac{n!}{(k!(n-k)!)}$$

	А	В	С	Д
1				
2				
3				

3 = COMBIN(3;2)

COMBINA

Функция **COMBINA** возвращает число комбинаций подмножества элементов.

Синтаксис функции:

=COMBINA(n; k)

где:

- **n** — общее число элементов в наборе;
- **k** — выбранное число элементов из набора.

Функция **COMBINA** возвращает число уникальных способов выбора элементов, когда порядок выбора является несущественным, и допускается повторение элементов. Например, если есть 3 элемента в наборе — А, В и С, Вы можете выбрать 2 элемента 6 различными способами, а именно, АА, АВ, АС, ВВ, ВС и СС; Вы можете выбрать 3 пункта 10 различными способами, а именно, ААА, ААВ, ААС, АВВ, АВС, АСС, ВВВ, ВВС, ВСС, ССС.

Функция **COMBINA** реализует формулу:

$$\frac{(n+k-1)!}{(k!(n-1)!)}$$

	A	B	C	D
1				
2		6	=COMBINA(3;2)	
3				
4		10	=COMBINA(3;3)	
5				

CONVERT

Функция **CONVERT** конвертирует наследуемые валюты разных европейских стран в и из Евро.

Синтаксис функции:

=CONVERT(value; currency1; currency2)

где:

- **value** — сумма валюты, которая будет преобразована;
- **currency1** и **currency2** — коды валюты для преобразования из и в соответственно. Они должны быть текстовым официальным сокращением для валюты (например, “EUR”), как показано в таблице ниже. Коэффициенты (показаны к Евро) были установлены Европейской Комиссией. Унаследованные валюты были заменены Евро в 2002 году.

“EUR”	“ATS”	13,7603	Австрийский шиллинг
“EUR”	“BEF”	40,3399	Бельгийский франк
“EUR”	“DEM”	1,95583	Немецкая марка
“EUR”	“ESP”	166,386	Испанская песета
“EUR”	“FIM”	5,94573	Финская марка
“EUR”	“FRF”	6,55957	Французский франк
“EUR”	“IEP”	0,787564	Ирландский фунт
“EUR”	“ITL”	1936,27	Итальянская лира
“EUR”	“LUF”	40,3399	Люксембургский франк
“EUR”	“NLG”	2,20371	Нидерландский гульден
“EUR”	“PTE”	200,482	Португальское эскудо
“EUR”	“GRD”	340,750	Греческая драхма

Или **currency1** или **currency2** должны быть определены как Евро.

	A	B	C	D	E
1					
2		7,27	=CONVERT(100;"ATS";"EUR")		
3					
4		195,58	=CONVERT(100;"EUR";"DEM")		
5					

В ячейке **B2** конвертируется 100 австрийских Шиллингов в Евро.

В ячейке **B4** конвертируется 100 Евро в Немецкие марки.

CONVERT_ADD

Функция **CONVERT** осуществляет преобразование значения из одних единиц измерения в другие.

Эта функция доступна, только если установлен пакет *Анализа*.

Синтаксис функции:

=CONVERT_ADD(number; originalunits; newunits)

где:

- **number** — значение, которое будет преобразовано;
- **originalunits** и **newunits** - текст, представляющий оригинальную и новую единицы измерения. Они являются регистрозависимыми, и должны быть выбраны из таблицы ниже. Дополнительная приставка указывает десятичный множитель из следующей таблицы.

Площадь	<p>“m2” — квадратный метр “mi2” — квадратная миля “Nmi2” — квадратная морская миля “in2” — квадратный дюйм “ft2” — квадратный фут “yd2” — квадратный ярд “ang2” — квадратный ангстрем “Pica2” — “Morgen” — “ar” — ар, 100 м² “acre” — акр, 1 акр = 4046,86 м² “ha” — гектар</p>
Длина	<p>“m” — метр “mi” — миля “Nmi” — морская миля “in” — дюйм “ft” — фут “yd” — ярд “ang” — ангстрем “Pica” “ell” “parsec” — парсек</p>
Энергия	<p>“J” — джоуль “e” — эрг “c” — Термодинамическая калория, 4.184 джоуля. Это не диетическая калория (килокалория). Для большей точности, используйте джоуль, из-за многих противоречивых определений калории. “cal” — Международная калория, 4.1868 джоуля. Это не диетическая калория (килокалория). Для большей точности, используйте джоуль, из-за многих противоречивых определений калории. “eV” — электронвольт “HPh” — лошадиная сила-час “Wh” — ватт-час “BTU” — британская тепловая единица (<i>составляет около 1060 джоулей</i>)</p>
Сила	<p>“N” — ньютон, единица измерения силы в системе СИ. “dyn” — дина, единица измерения силы в системе единиц СГС. “pond” — гравитационная сила действующая на массу в один грамм</p>

напряженность ЭМП	<p>“T” — тесла “ga” — гаусс</p>
Масса	<p>“g” — грамм “sg” “lbm” — фунт, международная единица веса (= 453,6 г) “u” — атомная единица массы “ozm” — унция (= 1/16 фунта, 28,3 г) “stone” — 14 фунтов “ton” — 2000 фунтов. Заметьте, что есть много других мер, также называемых “ton”; в частности это не метрическая тонна (тонна). “grain” — гран (1/7000 фунта, 0,0648 г) “pweight” “hweight” — английский центнер (= 50,8 кг) “shweight” — американский центнер (= 45,3 кг)</p>
Мощность	<p>“W” - ватт “HP” - лошадиная сила “PS” - немецкая “лошадиная сила”, почти, но не идентична “HP”</p>
Давление	<p>“Pa” — Паскаль “atm” — атмосфера “mmHg” — миллиметр ртутного столба “Torr” — торр, =101325/760 Па (близок, но не равен mmHg) “psi” — фунты на квадратный дюйм</p>
Скорость	<p>“m/s” — метр в секунду, м/с “m/h” — метр в час, м/ч “mph” — миля в час “kn” — Узел, равен одной миле в час или 1852/3600 м/с. “admkn” — Адмиралтейский узел (=6080 футов/ч)</p>
Температура	<p>“C” — градусы Цельсия “F” — градусы Фаренгейта “K” — градусы Кельвина “Reau” — градусы Реомюра “Rank” — градусы Ренкина</p>
Время	<p>“yr” — год “day” — день (точно 24 часа) “hr” — час (точно 60 минут) “mn” — минута (точно 60 секунд) “sec” — секунда</p>
Объем	<p>“l” — литр “tsp” — чайная ложка (американское общепринятое, традиционное значение), 1/6 жидкой унции в американской общепринятой системе измерений. “tbs” — столовая ложка (американское общепринятое традиционное значение). Она должна соответствовать 0,5 американской жидкой унции, а не 15 мл (обычно в США) или 20 мл (обычно в Австралии). “oz” — жидкая унция (американская общепринятая мера жидкости, равная 29,57 см³) “cup” — чашка (американская общепринятая мера жидкости) “pt” — американская пинта “qt” — кварта (американская общепринятая мера жидкости). Это — 0,946352946 литра, и таким образом не то же самое, что</p>

	<p>американская сухая кварта (1,101220 литра), и <i>при этом не</i> то же самое что английская кварта (которая используется в Великобритании и Канаде, и соответствует 1,1365225 литрам точно)</p> <p>“gal” — галлон (<i>американская общепринятая мера жидкости</i>)</p> <p>“m3” — кубический метр</p> <p>“mi3” — кубическая миля</p> <p>“Nmi3” — кубическая морская миля</p> <p>“in3” — кубический дюйм</p> <p>“ft3” — кубический фут</p> <p>“yd3” — кубический ярд</p> <p>“ang3” — кубический ангстрем</p> <p>“Pica3”</p> <p>“barrel” — американский нефтяной баррель, равен 42 американским галлонам (жидкости).</p> <p>“bushel” — американский бушель</p> <p>“regton” — зарегистрированная тонна Брутто, 100 кубических (международных) футов</p> <p>“Schooner”</p> <p>“Middy”</p> <p>“Glass”</p>
--	--

Вышеупомянутым единицам может предшествовать десятичный множитель. Например к (кило) — префикс для 10^3 — таким образом км означает километр. Если единица — мера площади, множитель возводится в квадрат, а если это мера объема, множитель возводится в куб. Например km² означает квадратный километр, что соответствует 10^6 квадратным метрам. Следующая таблицы содержит список префиксов:

префикс < 1	<p>y — йокто (10^{-24})</p> <p>z — зепто (10^{-21})</p> <p>a — атто (10^{-18})</p> <p>f — фемто (10^{-15})</p> <p>p — пико (10^{-12})</p> <p>n — нано (10^{-9})</p> <p>u — микро (10^{-6})</p> <p>m — милли (10^{-3})</p> <p>c — санти (10^{-2})</p> <p>d — деци (10^{-1})</p>
префикс > 1	<p>e — дека (10^1)</p> <p>h — гекто (10^2)</p> <p>k — кило (10^3)</p> <p>M — мега (10^6)</p> <p>G — гига (10^9)</p> <p>T — тера (10^{12})</p> <p>P — пета (10^{15})</p> <p>E — экса (10^{18})</p> <p>Z — зетта (10^{21})</p> <p>Y — йотта (10^{24})</p>

Это — стандартные десятичные приставки СИ, с двумя исключениями: микро и дека. Официально приставка для “микро” — μ , но на многих системах ее трудно ввести и/или увидеть, и она даже не доступна в некоторых системах, таким образом здесь используется традиционная (и универсально-используемая) замена “u”. Для “дека” (10^1) используется нестандартная приставка “e”, даже при том, что “da” легко поддерживается основными наборами символов.

Excel 2003, Lotus 1-2-3 и Corel Word Perfect (Quattro Pro) 12 не поддерживает стандартные приставки СИ **Z**, **Y**, **z** и **y**. OpenOffice.org 2.1 и Gnumeric 1.4 действительно поддерживают стандартные приставки СИ **Z**, **Y**, **z** и **y**. Они были включены как требуемые приставки, потому что они — стандартные приставки СИ, полезны, и уже поддерживаются некоторым приложениями. Короче говоря, поддерживаются стандартные приставки СИ в максимально возможной степени в пределах узкого набора символов.

	A	B	C	D	E
1					
2			10	=CONVERT_ADD(120;"in";"ft")	
3					
4			914,4	=CONVERT_ADD(10;"yd";"cm")	
5					

В ячейке **B2** возвращается 10, число футов в 120 дюймах.

В ячейке **B4** возвращается 914,4, что соответствует 10 ярдам, выраженным в сантиметрах.

По историческим причинам, многие из единиц не используют стандартные международные сокращения.

Эта функция в большинстве случаев не совместима с другими электронными таблицами.

COS

Функция **COS** возвращает косинус заданного угла (в радианах).

Синтаксис функции:

=COS(angle)

где:

- **angle** — угол в радианах.

Функция **COS** возвращает (тригонометрический) косинус угла, заданного в радианах. Если угол задан в градусах, умножьте его на **PI()/180** или используйте функцию **RADIANS**, чтобы преобразовать его в радианы.

	A	B	C	D
1				
2			0	=COS(PI()/2)
3				
4			0,5	=COS(RADIANS(60))
5				

В ячейке **B2** возвращается 0, косинус $\pi/2$ радиан. В ячейке **B4** возвращается 0,5, косинус 60 градусов.

COSH

Функция **COSH** возвращает гиперболический косинус числа.

Синтаксис функции:

=COSH(number)

где:

- **number** — любое действительное число, для которого требуется найти гиперболический косинус.

Функция **COSH** вычисляет гиперболический косинус заданного гиперболического угла. Гиперболический косинус — аналог обычного (тригонометрического) косинуса. Точки

($\cosh(t)$, $\sinh(t)$) определяют правую половину равносторонней гиперболы, так же, как точки ($\cos(t)$, $\sin(t)$) определяют точки круга. Формула для вычисления гиперболического косинуса имеет следующий вид:

$$\cosh(N) = \frac{e^N + e^{-N}}{2}$$

	A	B	C	D
1				
2			1=COSH(0)	
3				

В ячейке **B2** возвращается 1, гиперболический косинус 0.

COT

Функция **COT** возвращает котангенс заданного угла (в радианах).

Синтаксис функции:

=COT(angle)

где:

- **angle** — угол в радианах.

Функция **COT** возвращает (тригонометрический) котангенс угла, заданного в радианах. Если угол задан в градусах, умножьте его на **PI()/180** или используйте функцию **RADIANS**, чтобы преобразовать его в радианы.

Котангенс угла эквивалентен 1 деленной на тангенс этого угла.

	A	B	C	D
1				
2			1=COT(PI()/4)	
3				
4			1=COT(RADIANS(45))	
5				

В ячейке **B2** возвращается 1, котангенс $\pi/4$ радиан. В ячейке **B4** возвращается 1, котангенс 45 градусов.

COTH

Функция **COTH** возвращает гиперболический котангенс числа.

Синтаксис функции:

=COTH(number)

где:

- **number** — любое действительное число, для которого требуется найти гиперболический котангенс.

Функция **COTH** вычисляет гиперболический котангенс гиперболического угла. Гиперболический котангенс — аналог обычного (тригонометрического) котангенса. Формула для вычисления гиперболического котангенса имеет следующий вид:

$$\coth(N) = \frac{1}{\tanh(N)} = \frac{\cosh(N)}{\sinh(N)} = \frac{e^N + e^{-N}}{e^N - e^{-N}}$$

	A	B	C	D
1				
2		1,3130	=COTH(1)	
3				

В ячейке **B2** возвращается гиперболический котангенс 1, приблизительно 1,3130.

COUNTBLANK

Функция **COUNTBLANK** возвращает число пустых ячеек.

Синтаксис функции:

=COUNTBLANK(range)

где:

- **range** — диапазон, в котором требуется подсчитать пустые ячейки.

Функция **COUNTBLANK** возвращает число пустых ячеек в диапазоне ячеек **range**.

Ячейку, которая содержит пустой текст, такой как пробел, или даже текст с нулевой длиной, возвращающий "" (пустую строку), не считают пустой, даже при том, что она может казаться пустой.

	A	B	C	D	E	F	G
1							
2							
3							
4			4	=COUNTBLANK(A1:B2)			
5							
6			4	=SUMPRODUCT(TRIM(CLEAN(A1:B2))="")			
7							
8			0	=SUMPRODUCT(ISTEXT(A1:B2);TRIM(CLEAN(A1:B2))="")			
9							

В ячейке **B4** возвращается 4, так как ячейки A1, A2, B1 и B2 все пустые.

В ячейке **B6** возвращается количество ячеек, которые кажутся чистыми — то есть, которые действительно являются пустыми, или они содержат пустой текст или текст нулевой длины.

В ячейке **B8** возвращается количество ячеек, содержащих пустой текст и текст нулевой длины.

COUNTIF

Функция **COUNTIF** подсчитывает количество ячеек в диапазоне, которые удовлетворяют заданному условию.

Синтаксис функции:

=COUNTIF(range; condition)

где:

- **range** — диапазон, который будет проверен;
- **condition** — условие, может быть:
 - число, такое как 34,5
 - выражение, такое как 2/3 или SQRT (B5)
 - текстовая строка

Функция **COUNTIF** подсчитывает те ячейки в диапазоне **range**, которые равны условию,

если условие не текстовая строка, которая начинается с компаратора:

>, <, >=, <=, =, <>

В этом случае Функция **COUNTIF** сравнивает ячейки в диапазоне **range** с остатком от текстовой строки (интерпретируемом как число если возможно или как текст в противном случае).

Например условие “>4,5” проверяет, что содержимое каждой ячейки больше чем число 4,5, а условие “<собака” проверяет, что содержимое каждой ячейки располагается в алфавитном порядке перед текстом собака.

Может быть очень важно проверить параметры настройки в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**:

- Если флажок **Условия поиска = и <>** должен распространяться на всю ячейку установлен, то условие “red” будет соответствовать только **red**; если не установлен оно будет соответствовать **red, Fred, red herring**.
- Если флажок **Разрешить регулярные выражения в формулах** установлен, условие будет соответствовать использованию регулярных выражений — таким образом например “r.d” будет соответствовать **red, rod, rid**, а “red.*” будет соответствовать **red, redraw, redder**.
- Флажок **Учитывать регистр** не оказывает никакого влияния (не обращается внимание на регистр). См. примеры для понимания того, как добиться чувствительной к регистру проверки.

Пустые (не заполненные) ячейки в диапазоне **range** игнорируются (они никогда не удовлетворяют условию).

condition может определять только одно единственное условие. Смотри раздел *Условный подсчет и суммирование* для ознакомлением со способами определения множественных условий.

	A	B	C	D	E	F
1						>=20
2	3	1	=COUNTIF(A2:A8; ">=20")			20
3	5					
4	7	1	=COUNTIF(A2:A8; F1)			
5	25					
6	red	3	=COUNTIF(A2:A8; "<"&F2)			
7	Red					
8	S	3	=COUNTIF(A2:A8; ">=P")			
9						
10		2	=COUNTIF(A2:A8; "red")			
11						
12		1	=SUMPRODUCT(A2:A8="Red")			
13						

В ячейке **B2** возвращается количество ячеек в диапазоне **A2:A8**, содержимое которых в цифровой форме больше или равно 20.

В ячейке **B4** возвращается то же самое количество, так как ячейка **F1** содержит текст “>=20”.

В ячейке **B6** возвращается количество ячеек в диапазоне **A2:A8**, содержимое которых в цифровой форме меньше чем 20. Так как ячейка **F2** содержит число 20, выражение “<”&F2 преобразуется в условие “<20”.

В ячейке **B8** возвращается количество ячеек в диапазоне **A2:A8**, содержимое которых начинается с буквы P или далее по алфавиту.

В ячейке **B10** возвращается количество ячеек в диапазоне **A2:A8**, содержащих **red**, но это

значение может зависеть от параметров настройки, обсуждаемых выше.

В ячейке **B10** возвращается количество ячеек в диапазоне **A2:A8**, соответствующих **Red**, с учетом регистра.

DEGREES

Функция **DEGREES** преобразует радианы в градусы.

Синтаксис функции:

=DEGREES(radians)

где:

- **radians** — угол в радианах, который будет преобразован в градусы.

DEGREES(radians) эквивалентно **radians*180/PI()**.

	A	B	C	D
1				
2		180	=DEGREES(PI())	
3				

В ячейке **B2** возвращается 180 градусов.

DELTA

Функция **DELTA** возвращает 1 если два числа равны и 0 в противном случае.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=DELTA(number1; number2)

где:

- **number1** и **number2** — числа. Если **number2** опущено, оно предполагается равным 0.

Эта функция — реализация (математическая) дельта-функции Кронекера. Функция была введена Кронекером в 1866.

Выражение **number1=number2** возвращает **TRUE** или **FALSE** вместо 1 или 0, но во всем остальном идентично для числовых аргументов.

	A	B	C	D
1				
2	4	0	=DELTA(4; 5)	
3				
4		1	=DELTA(4; A2)	
5				
6		1	=DELTA(0)	
7				
8		ЛОЖЬ	=4=5	
9				

В ячейке **B2** возвращается 0.

В ячейке **B4** возвращается 1, так как ячейка **A2** содержит 4.

В ячейке **B6** возвращается 1.

В ячейке **B8** возвращается **ЛОЖЬ**, поскольку 4 не равно 5.

Функция **DELTA** должна использоваться только с числами.

Функция **DELTA** в настоящее время появляется в категории *Подключаемый модуль* Мастера функций.

ERF

Функция **ERF** вычисляет функцию ошибок (функция ошибок Гауса).

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=ERF(number1; number2)

где:

- **number1** и **number2** — числа.

Если **number2** опущено, функция **ERF** возвращает функцию ошибок, вычисленную между 0 и **number1**, в противном случае возвращает функцию ошибок, вычисленную между **number1** и **number2**.

Функция ошибок, также известная как функция ошибок Гауса, определяется для ERF(x) как:

$$\text{ERF}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

$$\text{ERF}(x_1; x_2) = \frac{2}{\sqrt{\pi}} \int_{x_1}^{x_2} e^{-t^2} dt = \text{ERF}(x_2) - \text{ERF}(x_1)$$

	A	B	C	D
1				
2		0,520499877813	=ERF(0,5)	
3				
4		0,297797288603	=ERF(0,2; 0,5)	
5				

Функция **ERF** в настоящее время появляется в категории *Подключаемый модуль* Мастера функций.

ERFC

Функция **ERFC** вычисляет дополнительную функцию ошибок (дополнительную функцию ошибок Гауса).

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=ERFC(number)

где:

- **number** — число, нижний предел интегрирования для дополнительной функции ошибок.

Функция **ERFC** возвращает функцию ошибок, вычисленную между заданным числом и бесконечностью, то есть, дополнительную функцию ошибок для числа.

$$\text{ERFC}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt = 1 - \text{ERF}(x)$$

	A	B	C	D
1				
2		0,479500122186953	=ERFC(0,5)	
3				

Функция **ERFC** в настоящее время появляется в категории *Подключаемый модуль* Мастера функций.

EVEN

Функция **EVEN** округляет число вверх, в сторону от нуля, до ближайшего четного целого числа.

Синтаксис функции:

=EVEN(number)

где:

- number** — заданное округляемое число.

Функция **EVEN** возвращает четное целое число, знак которого — тот же самый, что и у заданного числа и чье абсолютное значение больше или равно абсолютному значению заданного числа. Таким образом, если требуется округление, это округлено в сторону от нуля.

	A	B	C	D
1				
2		4	=EVEN(2,3)	
3				
4		2	=EVEN(2)	
5				
6		0	=EVEN(0)	
7				
8		-2	=EVEN(-0,5)	
9				

EXP

Функция **EXP** возвращает математическую константу e возведенную в степень числа.

Синтаксис функции:

=EXP(number)

где:

- number** — заданное число, для которого вычисляется экспоненциальная функция с основанием e .

Функция **EXP** возвращает e^{number} . Функция **EXP** является обратной к функции **LN** — натуральному логарифму числа. Вычисления производятся согласно следующей формуле:

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

	A	B	C	D
1				
2		2,71828182845904	=EXP(1)	
3				

В ячейке **B2** возвращается 2,71828182845904, математическая константа e с точностью Calc.

FACT

Функция **FACT** возвращает факториал числа.

Синтаксис функции:

=FACT(number)

где:

- **number** — число. **number** сначала преобразуется к целому числу с использованием INT.

Функция **FACT** возвращает **number!**, факториал числа вычисляемый как:

$$\text{FACT}(x) = \prod_{i=1}^x i$$

FACT(0) возвращает 1 по определению.

	A	B	C	D
1				
2		6	=FACT(3)	
3				
4		1	=FACT(0)	
5				

FACTDOUBLE

Функция **FACTDOUBLE** возвращает двойной факториал числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=FACTDOUBLE(number)

где:

- **number** — число. **number** сначала преобразуется к целому числу с использованием INT.

Функция **FACTDOUBLE** возвращает **number!!**, двойной факториал числа.

Для четных чисел **FACTDOUBLE(number)** возвращает:

$$(2n)!! = 2 \cdot 4 \cdot 6 \cdots 2n = \prod_{i=1}^n 2i = 2^n n!$$

Для нечетных чисел **FACTDOUBLE(number)** возвращает:

$$(2n+1)!! = 1 \cdot 3 \cdot 5 \cdots (2n+1) = \prod_{i=0}^n 2i+1 = \frac{(2n+1)!}{2^n n!}$$

FACTDOUBLE(0) возвращает 1 по определению.

	A	B	C	D
1				
2		15	=FACTDOUBLE(5)	
3				
4		48	=FACTDOUBLE(6)	
5				
6		1	=FACTDOUBLE(0)	
7				

Функция **FACTDOUBLE** в настоящее время появляется в категории *Подключаемый модуль*

Мастера функций.

FLOOR

Функция **FLOOR** возвращает число, округленное вниз до ближайшего числа, кратного другому числу.

Синтаксис функции:

=FLOOR(number; mult; mode)

где:

- **number** — округляемое число;
- **mult** — кратное, до которого требуется округлить;
- **mode** — режим округления. Если **mode** — ноль или опущен, функция **FLOOR** выполняет округление вниз к кратному меньшему (более отрицательному или равному) числу. Если **mode** — отличен от нуля, функция **FLOOR** выполняет округление к нулю. Это различие существенно только для отрицательных чисел. Используйте **mode=1** для совместимости, если Вы имеете отрицательные числа и хотите экспортировать в MS Excel. В MS Excel эта функция принимает только два аргумента.

Многие пользовательские приложения имеют функцию **FLOOR** только с двумя параметрами, и несколько отличной семантикой чем приведенная здесь (например, они работают, как будто была значение **mode** отлично от нуля). Эти функции **FLOOR** несовместимы со стандартным математическим определением **FLOOR**. Подобные приложения должны преобразовать такие формулы в формат с 3 параметрами при сохранении в формате OpenFormula, так, чтобы сохраненные формулы работали правильно.

	A	B	C	D	E
1					
2		6	=FLOOR(8; 3)		
3					
4		6	=FLOOR(6; 3)		
5					
6		-12	=FLOOR(-11; -2)		
7					
8		-12	=FLOOR(-11; -2; 0)		
9					
10		-10	=FLOOR(-11; -2; 1)		
11					

В ячейке **B2** возвращается 6, потому что $2 \cdot 3 = 6$ — первое кратное 3 число меньше 8.

В ячейках **B6** и **B8** возвращается -12, выполняется округление к кратному меньшему числу.

В ячейке **B10** возвращается -10, потому что с **mode=1** функция выполняет округление к нулю.

В Gnumeric и Excel, **FLOOR(-2.5;-1)**, возвращает -2, в OOo возвращает -3. Поскольку функция должна выполнять округление вниз, -3 кажется правильным ответом. В Kspread **FLOOR(-2.5)** возвращает -3. Эта функция придерживается странного *округления к нулю* Excel, только если задается третий параметр.

Реализация **FLOOR** в Kspread 1.5.0 принимает только один параметр.

GCD

Функция **GCD** возвращает наибольший общий делитель двух или более целых чисел.

Синтаксис функции:

=GCD(integer1; integer2; ... integer30)

где:

- **integer1, integer2, ... integer30** — до 30 целых чисел или диапазонов целых чисел, для которых должен быть вычислен наибольший общий делитель.

Наибольший общий делитель — наибольшее положительное целое число, на которое делится без остатка каждое из заданных целых чисел.

	A	B	C	D	E
1					
2			8	=GCD(16; 32; 24)	
3					
4	9		3	=GCD(A4:A6)	
5	12				
6	6		1	=GCD(3; 5)	
7					

В ячейке **C2** возвращается 8, потому что 8 — наибольшее число, на которое могут делиться без остатка 16, 24 и 32.

В ячейке **C4** возвращается 3, потому что ячейки A4, A5, и A6 содержат 9, 12, и 6.

В ячейке **C6** возвращается 1

Функция **GCD** в настоящее время возвращает результаты для отрицательных и дробных аргументов. Это не является документированной возможностью, и запланировано для OOo3.0, что использование отрицательных аргументов приведет к ошибке, а дробные аргументы будут обрезаться до обработки, чтобы соответствовать стандарту ODFD.

GCD_ADD

Функция **GCD_ADD** возвращает наибольший общий делитель двух или более целых чисел.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=GCD_ADD(integer1; integer2; ... integer30)

где:

- **integer1, integer2, ... integer30** — до 30 целых чисел или диапазонов целых чисел, для которых должен быть вычислен наибольший общий делитель.

Наибольший общий делитель — наибольшее положительное целое число, на которое делится без остатка каждое из заданных целых чисел.

	A	B	C	D	E	F
1						
2			8	=GCD_ADD(16; 32; 24)		
3						
4	9		3	=GCD_ADD(A4:A6)		
5	12					
6	6		1	=GCD_ADD(3; 5)		
7						

В ячейке **C2** возвращается 8, потому что 8 — наибольшее число, на которое могут делиться без остатка 16, 24 и 32.

В ячейке **C4** возвращается 3, потому что ячейки A4, A5, и A6 содержат 9, 12, и 6.

В ячейке **C6** возвращается 1.

Международный стандарт ODFD позволит обеим функциям **GCD** и **GCD_ADD** производить совершенно одинаковые результаты.

GESTEP

Функция **GESTEP** возвращает 1, если число больше или равно номеру шага, или 0 в противном случае.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=GESTEP(number; stepnumber)

где:

- **number** — проверяемое значение;
- **stepnumber** — пороговое значение.

Функция **GESTEP** возвращает 1, если **number** больше или равно **stepnumber** (оба являются числами). Если **stepnumber** опущен, оно, как предполагают, равняется 0.

Эта функция используется для фильтрации множества значений. Например, суммируя несколько функций **GESTEP**, можно определить количество значений, превосходящих некоторый порог.

	A	B	C	D	E
1					
2			0	=GESTEP(4; 5)	
3					
4	4		1	=GESTEP(4; A4)	
5					
6			1	=GESTEP(2)	
7					
8			ЛОЖЬ	=A4>=5	
9					

В ячейке **C2** возвращается 0.

В ячейке **C4** возвращается 1, так как ячейка **A4** содержит 4.

В ячейке **C6** возвращается 1, потому что 2 больше чем 0.

В ячейке **C8** возвращается ЛОЖЬ, так как ячейка **A4** содержит 4, а 4 не больше или равно 5. Этот подход очень подобен **GESTEP**, но не фильтрует нечисловые аргументы.

Функция **GESTEP** должна использоваться только с числами.

Функция **GESTEP** в настоящее время появляется в категории *Подключаемый модуль* Мастера функций.

INT

Функция **INT** округляет число вниз до ближайшего целого числа.

Синтаксис функции:

=INT(number)

где:

- **number** — округляемое число.

Функция **INT** возвращает число, округленное вниз до ближайшего целого числа. Отрицательные числа округляются до ближайшего меньшего целого числа: -1.3 округляется

до -2.

	A	B	C	D
1				
2		5	=INT(5,7)	
3				
4		-2	=INT(-1,3)	
5				

LCM

Функция **LCM** возвращает наименьшее общее кратное одного или более целых чисел.

Синтаксис функции:

=LCM(integer1; integer2; ... integer30)

где:

- **integer1, integer2, ... integer30** — до 30 целых чисел или диапазонов целых чисел, для которых должно быть вычислено наименьшее общее кратное.

Наименьшее общее кратное — наименьшее положительное целое число, которое делится на все заданные числа.

	A	B	C	D
1				
2		12	=LCM(4; 6)	
3				
4		3		
5		6		
6		6	=LCM(B4:B5)	
7				

В ячейке **B2** возвращается 12, которое является наименьшим положительным целым числом, кратным числам 4 ($12=4*3$) и 6 ($12=6*2$).

В ячейке **B6** возвращается 6, которое соответствует наименьшему общему кратному для чисел 3 и 6, содержащихся в диапазоне B4:B5.

В соответствии со стандартом ODF, в ODF 3.0 отрицательные аргументы функции **LCM** приведут к ошибке.

LCM_ADD

Функция **LCM_ADD** возвращает наименьшее общее кратное одного или более целых чисел.

Синтаксис функции:

=LCM_ADD(integer1; integer2; ... integer30)

где:

- **integer1, integer2, ... integer30** — до 30 целых чисел или диапазонов целых чисел, для которых должно быть вычислено наименьшее общее кратное.

Наименьшее общее кратное — наименьшее положительное целое число, которое делится на все заданные числа.

	A	B	C	D
1				
2		12	=LCM_ADD(4; 6)	
3				
4		3		
5		6		
6		6	=LCM_ADD(B4:B5)	
7				

В ячейке **B2** возвращается 12, которое является наименьшим положительным целым числом, кратным числам 4 ($12=4*3$) и 6 ($12=6*2$).

В ячейке **B6** возвращается 6, которое соответствует наименьшему общему кратному для чисел 3 и 6, содержащихся в диапазоне B4:B5.

Международный стандарт ODFD позволит обеим функциям **LCM / LCM_ADD** производить одинаковые результаты.

LN

Функция **LN** возвращает натуральный логарифм числа.

Синтаксис функции:

=LN(number)

где:

- **number** — положительное вещественное число, для которого вычисляется логарифм.

Функция **LN** возвращает натуральный логарифм (логарифм по основанию e) **number**, который соответствует показателю степени, в которую надо возвести число e чтобы получить **number**. Натуральный логарифм — обратная функция к экспоненциальной функции **EXP**.

Математическая константа e — приблизительно 2,71828182845904. Вычисление натурального логарифма осуществляется по следующей формуле:

$$\ln x = 2 \left[\frac{x-1}{x+1} + \frac{1}{3} \left(\frac{x-1}{x+1} \right)^3 + \frac{1}{5} \left(\frac{x-1}{x+1} \right)^5 + \dots \right]$$

	A	B	C	D
1				
2		1,09861	=LN(3)	
3				
4		321	=LN(EXP(321))	
5				

В ячейке **B2** возвращается натуральный логарифм 3 — приблизительно 1,09861.

В ячейке **B4** возвращается 321.

LOG

Функция **LOG** возвращает логарифм числа по заданному основанию.

Синтаксис функции:

=LOG(number; base)

где:

- **number** — положительное вещественное число, для которого вычисляется логарифм;
- **base** — основание логарифма.

Функция **LOG** возвращает логарифм по основанию **base** от числа **number**.

	A	B	C	D
1				
2		2,09590	=LOG(10; 3)	
3				
4		4	=LOG(7^4; 7)	
5				

В ячейке **B2** возвращается логарифм по основанию 3 от 10, приблизительно 2,0959.

В ячейке **B4** возвращается 4.

OpenOffice.org 2 требует указание основания логарифма, вместо того, чтобы считать его необязательным. Excel, и т.д. рассматривает основание как необязательный параметр, если оно не указывается, возвращается логарифм по основанию 10.

LOG10

Функция **LOG10** возвращает логарифм по основанию 10 от заданного числа.

Синтаксис функции:

=LOG10(number)

где:

- **number** — положительное вещественное число, для которого вычисляется логарифм.

Функция **LOG10** возвращает логарифм по основанию 10 от числа **number**.

	A	B	C	D
1				
2		0,69897	=LOG10(5)	
3				
4		2	=LOG10(100)	
5				

В ячейке **B2** возвращается логарифм по основанию 10 от 5, приблизительно 0,69897.

В ячейке **B4** возвращается 2, логарифм по основанию 10 от 100.

MOD

Функция **MOD** возвращает остаток от деления одного целого числа на другое.

Синтаксис функции:

=MOD(number; divisor)

где:

- **number** — число, остаток от деления которого определяется (делимое);
- **divisor** — число, на которое делится (делитель).

Для целочисленных аргументов функция **MOD** возвращает остаток от деления **number** на **divisor**.

Эта функция реализована как:

$$number - divisor * INT(number / divisor)$$

и дает результат, если аргументы не являются целыми числами.

	A	B	C	D	E
1					
2		1	=MOD(22; 3)		
3					
4		1,25	=MOD(11,25; 2,5)		
5					
6		1	=MOD(-2;3)		
7					
8		-1	=MOD(2;-3)		
9					
10		-2	=MOD(-2;-3)		
11					

В ячейке **B2** возвращается 1, остаток от деления 22 на 3.

В ячейке **B4** возвращается 1,25.

Функция **MOD** с отрицательными числами является коварной. Тесты на **MOD(-2; 3)**, **MOD(2;-3)**, и **MOD(-2;-3)** работают в Gnumeric, OOo2 и Excel 2002 как показано выше. KSpread 1.4.2 также вычисляет **MOD(-2; 3)** как 1. Однако, KSpread 1.4.2 (в Fedora Core 4) вычисляет **MOD(-2;-3)** как -5 (а не -1), а **MOD(2;-3)** — как 2 (а не -1). 9 января 2006, разработчик KSpread Томас Мекир объявил, что development версия KSpread была изменена, и следующий выпуск KSpread будет возвращать такие же результаты с этими и другими отрицательными значениями.

MROUND

Функция **MROUND** возвращает число, округленное до ближайшего числа, кратного другому числу.

Эта функция доступна, только если установлен пакет *Анализа*.

Синтаксис функции:

=MROUND(number; mult)

где:

- **number** — округляемое число;
- **mult** — кратное, до которого требуется округлить.

Функция **MROUND** возвращает число, округленное до ближайшего числа, кратного **mult**, которое содержит **mult** целое число раз.

Альтернативной реализацией будет **mult * ROUND(number/mult)**.

	A	B	C	D	E
1					
2		15	=MROUND(15,5; 3)		
3					
4		1,5	=MROUND(1,4; 0,5)		
5					

В ячейке **B2** возвращается 15, так как 15,5 ближе к 15 (= 3*5) чем к 18 (= 3*6).

В ячейке **B4** возвращается 1,5 (= 0,5*3).

Функция **MROUND** позволяет Вам выполнять округление с любой точностью, которую Вы хотите, и особенно полезна в финансовых/коммерческих приложениях, где требуется подобное округление. Она округляет заданное значение до любого кратного числа, которое Вы определите. Функция **MROUND** может использоваться для округления времени, например, к ближайшей четверти часа.

	A	B	C	D	E	F
1						
2		13:27:00	13:30:00	=MROUND(B2; TIME(0;15;0))		
3						

В отличие от эквивалентной функции Excel, **MROUND** в Calc допускает отрицательные числа.

MULTINOMIAL

Функция **MULTINOMIAL** возвращает факториал суммы аргументов, разделенный на произведение факториалов аргументов.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=MULTINOMIAL (number1; number2; ... number30)

где:

- **number1, number2, ... number30** — до 30 чисел или диапазонов чисел.

Эта функция возвращает:

$$\frac{(number1 + number2 + \dots + number30)!}{(number1! * number2! * \dots * number30!)}$$

	A	B	C	D
1				
2		3	=MULTINOMIAL(1; 2)	
3				
4		2		
5		3		
6		4		
7		1260	=MULTINOMIAL(B4:B6)	
8				

В ячейке **B2** возвращается 3, что соответствует $(1+2)!/(1!*2!)$.

В ячейке **B7** возвращается 1260, что соответствует $(2+3+4)!/(2!*3!*4!)$, так как диапазон **B4:B6** содержит числа 2, 3 и 4.

Мультиномиальное распределение — обобщение биномиального распределения на случай независимых испытаний случайного эксперимента с несколькими возможными исходами в теории вероятности. Функция **MULTINOMIAL()** в настоящее время относится к категории Математических функций, но была бы с большей готовностью отнесена к категории Статистических функций.

ODD

Функция **ODD** округляет число вверх, в сторону от нуля, до ближайшего нечетного целого числа.

Синтаксис функции:

=ODD(number)

где:

- **number** — заданное округляемое число.

Функция **ODD** возвращает нечетное целое число, знак которого — тот же самый, что и у заданного числа и чье абсолютное значение больше или равно абсолютному значению

заданного числа. Таким образом, если требуется округление, это округлено в сторону от нуля. По определению, **ODD(0) = 1**.

	A	B	C	D
1				
2		3	=ODD(1,2)	
3				
4		1	=ODD(1)	
5				
6		1	=ODD(0)	
7				
8		-5	=ODD(-3,1)	
9				

PI

Функция **PI** возвращает 3,14159265358979, величину математической постоянной π с точностью до 14 десятичных знаков.

Синтаксис функции:

=PI()

Эта функция не принимает никаких аргументов и возвращает (приблизительное) значение числа π . Число π — иррациональное число, и не может быть представлено конечным числом цифр.

	A	B	C	D
1				
2		3,1415926535898	=PI()	
3				

POWER

Функция **POWER** возвращает число возведенное в степень.

Синтаксис функции:

=POWER(number; power)

где:

- **number** — основание, может быть любым вещественным числом;
- **power** — показатель степени, в которую возводится основание.

Функция **POWER** возвращает $number^{power}$, т.е. **number** возведенное в степень **power**.

Тот же самый результат может быть достигнут при использовании оператора возведения в степень \wedge :

number^{power}

	A	B	C	D
1				
2		64	=POWER(4; 3)	
3				
4		64	=4^3	
5				

В ячейке **B2** возвращается 64, что соответствует 4 возведенному в степень 3.

В ячейке **B4** также возвращается 4 в степени 3.

PRODUCT

Функция **PRODUCT** возвращает произведение всех чисел, заданных в виде аргументов.

Синтаксис функции:

=PRODUCT(number1; number2; ... number30)

где:

- **number1, number2, ... number30** — до 30 чисел или диапазонов/массивов чисел, произведение которых должно быть вычислено.

Функция **PRODUCT** возвращает **number1 * number2 * number3 * ...**

Функция **PRODUCT** игнорирует любой текст или пустую ячейку в пределах диапазона или массива.

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					

В Excel **PRODUCT** игнорирует логические значения (то есть, это рассматривает их как 1). В Calc TRUE — 1, и FALSE — 0.

QUOTIENT

Функция **QUOTIENT** возвращает целую часть результата деления. Функция используется, когда нужно отбросить остаток от деления.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=QUOTIENT(numerator; denominator)

где:

- **numerator** — делимое;
- **denominator** — делитель.

Функция **QUOTIENT** возвращает целую часть от деления **numerator** на **denominator**. **QUOTIENT** эквивалентна **INT(numerator/denominator)**, за исключением того, что она может сообщать об ошибках с различными кодами ошибки.

	A	B	C	D
1				
2				
3				

В ячейке **B2** возвращается 3, целая часть от деления числа 11 на 3. Остаток 2 теряется.

RADIANS

Функция **RADIANS** преобразует градусы в радианы.

Синтаксис функции:

=RADIANS(degrees)

где:

- **degrees** - угол в градусах, который будет преобразован в радианы.

RADIANS(degrees) эквивалентно **degrees*PI()/180**.

	A	B	C	D
1				
2		1,5707963267949	=RADIANS(90)	
3				

В ячейке **B2** возвращается 1,5707963267949, что соответствует **PI/2** с точностью Calc.

RAND

Функция **RAND** возвращает случайное число между 0 и 1.

Синтаксис функции:

=RAND()

Эта функция генерирует новое случайное число большее или равное 0, и меньше 1 каждый раз, когда Calc выполняет перерасчет.

Чтобы заставить Calc выполнять перерасчет вручную нажмите *Shift+Ctrl+F9*.

Заметьте, что в отличие от большинства функций, эта функция будет возвращать *различные* значения каждый раз, когда вызывается с одними и теми же (пустой набор) параметрами.

Для генерации случайного числа, которое никогда не пересчитывается, скопируйте ячейку, содержащую **=RAND()**, и используйте **Правка — Вставить как...** (снимите выбор с **Вставить все** и **Формулы** и выберите **Числа**).

	A	B	C	D
1				
2		0,97	=RAND()	
3				
4		11,14	=RAND()*(12-6) + 6	
5				

В ячейке **B2** возвращается случайное число между 0 (включительно) и 1 (исключая).

В ячейке **B4** возвращается случайное вещественное число между 6 и 12.

Некоторые генераторы случайных чисел лучше чем другие. В настоящее время нет никакого особого требования для того, насколько случайным должен быть результат. Разработчики должны избегать использования встроенных генераторов случайных чисел, предоставляемых языковыми библиотеками, потому что они зачастую используют чрезвычайно ненадежные алгоритмы.

RANDBETWEEN

Функция **RANDBETWEEN** возвращает целое случайное число в указанном диапазоне.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=RANDBETWEEN(bottom; top)

где:

- **bottom** — целое число, нижняя граница диапазона;
- **top** — целое число, верхняя граница диапазона.

Функция **RANDBETWEEN** возвращает случайное целое число между **bottom** и **top** (оба включительно).

Эта функция генерирует новое случайное число каждый раз, когда Calc выполняет перерасчет. Чтобы заставить Calc выполнять перерасчет вручную нажмите *Shift+Ctrl+F9*.

Заметьте, что в отличие от большинства функций, эта функция будет возвращать *различные* значения каждый раз, когда вызывается с одними и теми же параметрами.

Для генерации случайного числа, которое никогда не перерасчитывается, скопируйте ячейку, содержащую эту функцию, и используйте **Правка — Вставить как...** (снимите выбор с **Вставить все** и **Формулы** и выберите **Числа**).

Альтернативой этой функции, не требующая *пакет Анализа* было бы:

$$\text{INT}(\text{RAND}() * (\text{top} - \text{bottom} + 1)) + \text{bottom}.$$

	A	B	C	D	E
1					
2		26	=RANDBETWEEN(20; 30)		
3					

В ячейке **B2** возвращается случайное число между 20 и 30 (включительно).

ROUND

Функция **ROUND** округляет число с заданной точностью.

Синтаксис функции:

=ROUND(number; places)

где:

- **number** — округляемое число;
- **places** — количество десятичных разрядов, до которого нужно округлить число.

Функция **ROUND** возвращает число, округленное до заданного количества десятичных разрядов. Если **places** больше 0, то число округляется до указанного количества десятичных разрядов справа от десятичного разделителя. Если **places** опущено или ноль, функция выполняет округление до ближайшего целого числа. Если **places** отрицательно, функция выполняет округление до ближайших 10, 100, 1000, и т.д.

Эта функция выполняет округление к самому близкому числу. Смотри функции **ROUNDDOWN** и **ROUNDUP** как альтернативу данной.

	A	B	C	D
1				
2		2,35	=ROUND(2,348; 2)	
3				
4		2	=ROUND(2,348; 0)	
5				
6		3	=ROUND(2,5)	
7				
8		1000	=ROUND(987,65; -2)	
9				

Если эти примеры, как Вам кажется, не работают, посмотрите диалоговое окно **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, чтобы проверить точность отображения.

ROUNDDOWN

Функция **ROUNDDOWN** округляет число вниз, к нулю, с заданной точностью.

Синтаксис функции:

=ROUNDDOWN(number; places)

где:

- **number** — округляемое число;
- **places** — количество десятичных разрядов, до которого нужно округлить число.

Функция **ROUNDDOWN** возвращает число, округленное вниз (к нулю) до заданного количества десятичных разрядов. Если **places** больше 0, то число округляется вниз до указанного количества десятичных разрядов справа от десятичного разделителя. Если **places** опущено или ноль, функция выполняет округление вниз до целого числа. Если **places** отрицательно, функция выполняет округление вниз до ближайших 10, 100, 1000, и т.д.

Эта функция выполняет округление к нулю. Смотри функции **ROUND** и **ROUNDUP** как альтернативу данной.

	A	B	C	D	E
1					
2			1,23	=ROUNDDOWN(1,234; 2)	
3					
4			45	=ROUNDDOWN(45,67; 0)	
5					
6			-45	=ROUNDDOWN(-45,67)	
7					
8			900	=ROUNDDOWN(987,65; -2)	
9					

Если эти примеры, как Вам кажется, не работают, посмотрите диалоговое окно **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, чтобы проверить точность отображения.

ROUNDUP

Функция **ROUNDUP** округляет число вверх, в сторону от нуля, с заданной точностью.

Синтаксис функции:

=ROUNDUP(number; places)

где:

- **number** — округляемое число;
- **places** — количество десятичных разрядов, до которого нужно округлить число.

Функция **ROUNDUP** возвращает число, округленное вверх (в сторону от нуля) до заданного количества десятичных разрядов. Если **places** больше 0, то число округляется вверх до указанного количества десятичных разрядов справа от десятичного разделителя. Если **places** опущено или ноль, функция выполняет округление вверх до целого числа. Если **places** отрицательно, функция выполняет округление вверх до ближайших 10, 100, 1000, и т.д.

Эта функция выполняет округление в сторону от нуля. Смотри функции **ROUND** и **ROUNDDOWN** как альтернативу данной.

	A	B	C	D	E
1					
2		1,12	=ROUNDUP(1,1111; 2)		
3					
4		1,3	=ROUNDUP(1,2345; 1)		
5					
6		46	=ROUNDUP(45,67; 0)		
7					
8		-46	=ROUNDUP(-45,67)		
9					
10		1000	=ROUNDUP(987,65; -2)		
11					

Если эти примеры, как Вам кажется, не работают, посмотрите диалоговое окно **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, чтобы проверить точность отображения.

SERIESSUM

Функция **SERIESSUM** возвращает сумму первых членов степенного ряда.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=SERIESSUM(x; n; m; ar)

где:

- **x** — значение переменной степенного ряда;
- **n** — показатель степени x для первого члена степенного ряда;
- **m** — шаг, на который увеличивается показатель степени n для каждого следующего члена степенного ряда;
- **ar** — ссылается на диапазон, содержащий коэффициенты при соответствующих степенях x. Количество значений в аргументе **ar** определяет количество членов степенного ряда.

Степенной ряд может быть представлен в виде:

$$f(x) = a_0 x^n + a_1 x^{n+m} + a_2 x^{n+2m} + a_3 x^{n+3m} + \dots$$

Например, следующий степенной ряд может использоваться для выражения математической константы *e*, возведенной в степень:

$$e^x = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

Когда $x = 1$, суммируемые элементы ряда приближаются к *e*. Чтобы суммировать первые 5 элементов, используя **SERIESSUM**, мы должны установить: **x = 1**; **n = 0**; **m = 1**; **ar** в B1:B5

где B1, B2, B3, B4, B5 содержат соответственно: **=1/FACT(0)**, **=1/FACT(1)**, **=1/FACT(2)**, **=1/FACT(3)**, **=1/FACT(4)**.

Теперь, используя эти значения в функции **SERIESSUM**:

	A	B	C	D	E
1		1,00000000000	=1/FACT(0)		
2		1,00000000000	=1/FACT(1)		
3		0,50000000000	=1/FACT(2)		
4		0,16666666667	=1/FACT(3)		
5		0,04166666667	=1/FACT(4)		
6					
7		2,70833333333	=SERIESSUM(1; 0; 1; B1:B5)		
8					

В ячейке **B7** получаем 2,70833333333, приближенное значение числа e (= 2,71828182845904 ...). Используя большее количество элементов дало бы более точное приближение.

SIGN

Возвращает знак числа: +1, если число положительное, -1 если отрицательное и 0 если ноль.

Синтаксис функции:

=SIGN(number)

где:

- number** — число, знак которого должен быть определен.

	A	B	C	D
1				
2		1	=SIGN(3,4)	
3				
4		-1	=SIGN(-4,5)	
5				
6		0	=SIGN(0)	
7				

SIN

Функция **SIN** возвращает синус заданного угла (в радианах).

Синтаксис функции:

=SIN(angle)

где:

- angle** — угол в радианах.

Функция **SIN** возвращает (тригонометрический) синус угла, заданного в радианах. Если угол задан в градусах, умножьте его на **PI()/180** или используйте функцию **RADIANS**, чтобы преобразовать его в радианы.

	A	B	C	D
1				
2		1	=SIN(PI()/2)	
3				
4		0,5	=SIN(RADIANS(30))	
5				

В ячейке **B2** возвращается 1, синус $\pi/2$ радиан. В ячейке **B4** возвращается 0,5, синус 30 градусов.

SINH

Функция **SINH** возвращает гиперболический синус числа.

Синтаксис функции:

=SINH(number)

где:

- **number** — любое действительное число, для которого требуется найти гиперболический синус.

Функция **SINH** вычисляет гиперболический синус заданного гиперболического угла. Гиперболический синус — аналог обычного (тригонометрического) синуса. Точки ($\cosh(t)$, $\sinh(t)$) определяют правую половину равносторонней гиперболы, так же, как точки ($\cos(t)$, $\sin(t)$) определяют точки круга. Формула для вычисления гиперболического синуса имеет следующий вид:

$$\sinh(N) = \frac{e^N - e^{-N}}{2}$$

	A	B	C	D
1				
2		0	=SINH(0)	
3				

В ячейке **B2** возвращается 0, гиперболический синус 0.

SQRT

Функция **SQRT** возвращает положительное значение квадратного корня числа.

Синтаксис функции:

=SQRT(number)

где:

- **number** — положительное число, для которого вычисляется квадратный корень.

Если **number** отрицательно, то функция **SQRT** возвращает значение ошибки **Ошибка:502, недопустимый параметр**.

	A	B	C	D
1				
2		4	=SQRT(16)	
3				
4		Ошибка:502	=SQRT(-16)	
5				

SQRTPI

Функция **SQRTPI** возвращает квадратный корень из произведения π на число.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=SQRTPI(number)

где:

- **number** — положительное число.

Функция **SQRTPI** возвращает положительный квадратный корень из (**PI**) умноженного на **number**). Эта функция эквивалентна **SQRT(PI()*number)**.

	A	B	C	D
1				
2		2,50662827463	=SQRTPI(2)	
3				

В ячейке **B2** возвращается квадратный корень из **2π**, приблизительно 2,50662827463.

SUBTOTAL

Функция **SUBTOTAL** возвращает результат **SUM**, **AVERAGE**, **STDEV** и т.д. для фильтрованных данных.

Синтаксис функции:

=SUBTOTAL(function; range)

где:

- **function** — число, которое определяет одну из следующих функций:
 - 1 — **AVERAGE**
 - 2 — **COUNT**
 - 3 — **COUNTA**
 - 4 — **MAX**
 - 5 — **MIN**
 - 6 — **PRODUCT**
 - 7 — **STDEV**
 - 8 — **STDEVP**
 - 9 — **SUM**
 - 10 — **VAR**
 - 11 — **VARP**
- **range** — полный диапазон, из которого выбираются отфильтрованные ячейки для вычисления.

Функция **SUBTOTAL** дает возможность Вам выполнять основные операции электронной таблицы на видимых ячейках в режиме Быстрого фильтра. Если диапазон содержит другие функции **SUBTOTAL**, они игнорируются, чтобы избежать двойного подсчета.

Вы имеете таблицу в диапазоне ячеек A1:B5, содержащую название города в столбце A и сопровождающееся данными в столбце B. Вы используете Быстрый фильтр таким образом, чтобы видеть только те строки, которые содержат город Гамбург. Вы хотите увидеть сумму данных, которые отображаются; то есть, только сумму для отфильтрованных строк.

	A	B	C	D
1	Город	Данные		
2	Гамбург	20		
4	Гамбург	15		
5	Гамбург	5		
7				
8		40	=SUBTOTAL(9; B2:B6)	
9				

В ячейке **B8** возвращается желаемый результат.

SUM

Функция **SUM** суммирует содержимое ячеек.

Синтаксис функции:

=SUM(number1; number2; ... number30)

где:

- **number1, number2, ... number30** — до 30 чисел или диапазонов/массивов чисел, сумма которых должна быть вычислена.

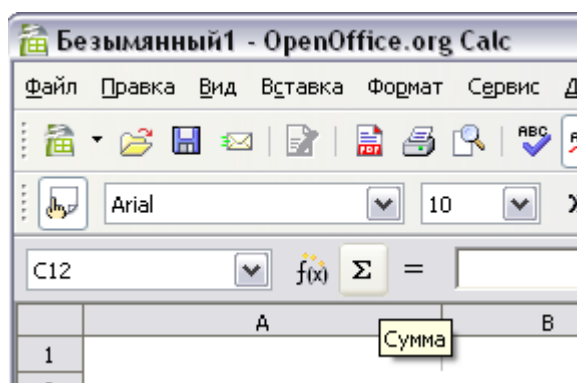
Функция **SUM** игнорирует любые текстовые или пустые ячейки в пределах диапазона или массива.

Функция **SUM** может также использоваться для суммирования или подсчета ячеек, для которых верно заданное условие — смотри *Условный подсчет и суммирование*.

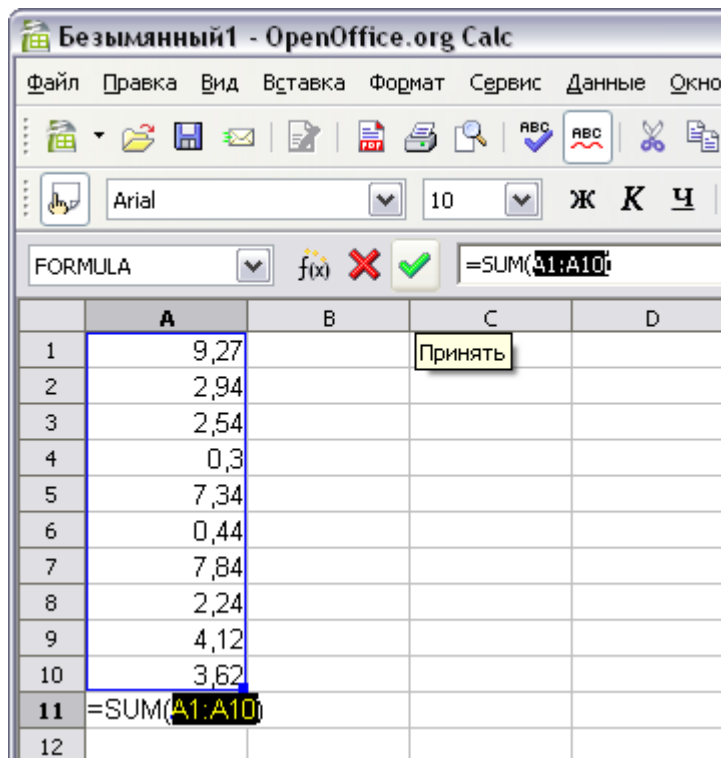
	А	В	С	Д	Е
1		Продажи по регионам			
2		2002	2003	2004	
3	Северо-Восток	14 223	16 774	17 429	
4	Центрально-Атлантический	8 282	9 327	10 043	
5	Южный	11 283	14 634	14 844	
6	Средне-Западный	98 321	112 937	109 385	
7	Юго-Западный	1 143	1 438	1 573	
8	Западный	36 509	41 738	44 808	
9		169 761	196 848	198 082	
10		=SUM(B3:B8)	=SUM(C3:C8)	=SUM(D3:D8)	
11					

Здесь мы имеем шесть регионов продаж — и чтобы определить полные продажи в течение каждого года в каждом из регионов, мы используем функцию **SUM**, как проиллюстрировано выше. Следует отметить, что диапазон ячеек, к которым применяется функция **SUM**, не обязательно должен быть одним столбцом или строкой.

ООо Calc предлагает простой способ суммирования ряда чисел. Существует значок **Сумма** на панели формул (см. ниже)



Сначала, выберите ячейку, а затем нажмите на значок **Сумма**. Если выбранная ячейка будет у основания столбца чисел, то ООо Calc сделает интеллектуальное предположение о диапазоне суммирования — который Вы должны принять или отвергнуть.



Зачастую, нам нужно подвести итог ряда чисел, которые занимают одинаковую позицию на различных листах. Это очень просто, как показано ниже.

	A	B	C	D	E
1	4	=Январь.A1			
2	3	=Февраль.A1			
3	5	=Март.A1			
4	7	=Апрель.A1			
5	19	=SUM(A1:A4)			
6	19	=SUM(Январь.A1:Апрель.A1)			
7					

Январь / Февраль / Март / Апрель / **Итоги**

Мы имеем по листу на каждый месяц, а также лист **Итоги**. Для иллюстрации мы складываем ячейки **A1** для каждого листа с **января** по **апрель** формулой **=SUM(Январь.A1:Апрель.A1)**. Заметьте, что это немного отличается от соответствующей формулы Excel — **=SUM(Январь:Апрель! A1)**

В Excel **SUM** игнорирует логические значения (то есть, это рассматривает их как 0). В Calc TRUE — 1, и FALSE — 0.

SUMIF

Функция **SUMIF** суммирует содержимое ячеек в диапазоне, которые удовлетворяют заданному условию.

Синтаксис функции:

=SUMIF(test_range; condition; sum_range)

где:

- **test_range** — диапазон, который будет проверен;
- **condition** — условие, может быть:
 - число, такое как 34,5;
 - выражение, такое как 2/3 или **SQRT(B5)**;

- текстовая строка;
- **sum_range** — диапазон, фактические ячейки для суммирования.

Функция **SUMIF** определяет те ячейки в диапазоне **test_range**, которые удовлетворяют условию, и суммирует соответствующие ячейки в диапазоне **sum_range**. Если **sum_range** опущен, суммируются ячейки в диапазоне **test_range**.

SUMIF ищет ячейки в диапазоне **test_range**, которые равны условию, если условие не текстовая строка, которая начинается с компаратора:

>, <, >=, <=, =, <>

В этом случае **SUMIF** сравнивает ячейки в диапазоне **test_range** с остатком от текстовой строки (интерпретируемом как число, если возможно, или как текст в противном случае).

Например условие “>4,5” проверяет, что содержимое каждой ячейки больше чем число 4,5, а условие “<собака” проверяет, что содержимое каждой ячейки располагается в алфавитном порядке перед словом **собака**.

Может быть очень важно проверить параметры настройки в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**:

- Если флажок **Условия поиска = и <> должен распространяться на всю ячейку** установлен, то условие “red” будет соответствовать только **red**; если не установлен оно будет соответствовать **red, Fred, red herring**.
- Если флажок **Разрешить регулярные выражения в формулах** установлен, условие будет соответствовать использованию регулярных выражений — таким образом например “r.d” будет соответствовать **red, rod, rid**, а “red.*” будет соответствовать **red, redraw, redder**.
- Флажок **Учитывать регистр** не оказывает никакого влияния (не обращается внимание на регистр). См. примеры для понимания того, как добиться чувствительной к регистру проверки.

Пустые (не заполненные) ячейки в диапазоне **test_range** игнорируются (они никогда не удовлетворяют условию).

Строка **condition** может определять только одно единственное условие. Смотри раздел *Условный подсчет и суммирование* для ознакомлением со способами определения множественных условий.

	A	B	C	D	E	F	G
1	1			яблоки		>=0	
2	-2	7	3	груши		10	
3	3	9	5	Яблоки			
4	-4	11	7	яблоки			
5	5			груши			
6	-6			яблоки			
7	7			груши			
8	-8			яблоки			
9	9			Яблоки			
10							
11	-20	=SUMIF(A1:A9;"<0")					
12							
13	25	=SUMIF(A1:A9; F1)					
14							
15	8	=SUMIF(B2:B4; "<"&F2; C2:C4)					
16							
17	-5	=SUMIF(D1:D9; "яблоки"; A1:A9)					
18							
19	12	=SUMPRODUCT(D1:D9="Яблоки"; A1:A9)					
20							

В ячейке **B11** возвращается сумма отрицательных чисел в диапазоне **A1:A9**.

В ячейке **B13** возвращается сумма положительных чисел в диапазоне **A1:A9**, если **F1** содержит текст “>=0” (без двойных кавычек).

В ячейке **B15** возвращается сумма ячеек **C2** и **C3**, потому что ячейки **B2** и **B3** — меньше чем 10 (так как ячейка **F2** содержит 10, условие преобразуется к виду “<10”).

В ячейке **B17** возвращается сумма ячеек из диапазона **A1:A9**, соответствующие которым ячейки из диапазона **D1:D9** содержат текст “яблоки” (без учета регистра).

В ячейке **B19** возвращается сумма записей из диапазона **A1:A9**, соответствующий которым элементы из диапазона **D1:D9** содержат текст “Яблоки”, с учетом регистра (записи “яблоки”, и “ЯБЛОКИ” не соответствуют условию).

SUMSQ

Функция **SUMSQ** возвращает сумму квадратов аргументов.

Синтаксис функции:

=SUMSQ(number1; number2; number30)

где:

- **number1, number2, ... number30** — до 30 чисел или диапазонов чисел, сумма квадратов которых должна быть вычислена.

	A	B	C	D
1				
2		29	=SUMSQ(2; 3; 4)	
3				
4		1		
5		2		
6		5	=SUMSQ(B4:B5)	
7				

В ячейке **B2** возвращается 29, что соответствует $2*2 + 3*3 + 4*4$.

В ячейке **B6** возвращается 5, что соответствует $1*1 + 2*2$, поскольку **B4** содержит 1, а **B5** — 2.

TAN

Функция **TAN** возвращает тангенс заданного угла (в радианах).

Синтаксис функции:

=TAN(angle)

где:

- **angle** — угол в радианах.

Функция **TAN** возвращает (тригонометрический) тангенс угла, заданного в радианах. Если угол задан в градусах, умножьте его на $\text{PI}()/180$ или используйте функцию **RADIANS**, чтобы преобразовать его в радианы.

	A	B	C	D
1				
2		1	=TAN(PI()/4)	
3				
4		1	=TAN(RADIANS(45))	
5				

В ячейке **B2** возвращается 1, тангенс $\pi/4$ радиан. В ячейке **B4** возвращается 1, тангенс 45 градусов.

TANH

Функция **TANH** возвращает гиперболический тангенс числа.

Синтаксис функции:

=TANH(number)

где:

- **number** — любое действительное число, для которого требуется найти гиперболический тангенс.

Функция **TANH** вычисляет гиперболический тангенс заданного гиперболического угла. Гиперболический тангенс — аналог обычного (тригонометрического) тангенса. Формула для вычисления гиперболического тангенса имеет следующий вид:

$$\tanh(N) = \frac{\sinh(N)}{\cosh(N)} = \frac{e^N - e^{-N}}{e^N + e^{-N}}$$

	A	B	C	D
1				
2		0	=TANH(0)	
3				

В ячейке **B2** возвращается 0, гиперболический тангенс 0.

TRUNC

Функция **TRUNC** усекает число, удаляя десятичные разряды.

Синтаксис функции:

=TRUNC(number; places)

где:

- **number** — любое действительное число;
- **places** — число, определяющее точность усечения. Значение по умолчанию — 0 (нуль).

Функция **TRUNC** возвращает число с указанным количеством десятичных разрядов. Лишние десятичные разряды просто удаляются, независимо от знака. **TRUNC(number; 0)** ведет себя как **INT(number)** для положительных чисел, но эффективно округляет к нулю для отрицательных чисел.

	A	B	C	D
1				
2		1,23	=TRUNC(1,239; 2)	
3				
4		-1,234	=TRUNC(-1,234999; 3)	
5				
6		5550	=TRUNC(5555;-1)	
7				

В ячейке **B2** возвращается 1,23, 9 теряется. В ячейке **B4** возвращается -1.234. Все девятки теряются. В ячейке **B6** возвращается 5550, последняя пятерка теряется.

Условный подсчет и суммирование

Далее приведен обзор различных способов условного подсчета и суммирования содержимого ячеек, в зависимости от результата некоторой проверки.

Условия в диапазонах ячеек

Один простой метод для подсчета или суммирования, использующий несколько условий — ввести условия в новую строку или столбец. Например, если диапазон **A1:A6** содержит список цветов, а диапазон **B1:B6** — список размеров, то мы можем ввести в ячейку **D1** формулу **=(A1=“красный”)**, которая возвращается **TRUE** или **FALSE** в зависимости от того, содержит ячейка **A1** текст **красный** или нет. В качестве альтернативы, мы можем ввести в ячейку **D1** формулу **=AND(A1=“красный”; B1=“большой”)**, которая возвращает **TRUE**, если ячейка **A1** содержит **красный** И ячейка **B1** содержит **большой** и **FALSE** в противном случае. Скопируйте и вставьте эту формулу в ячейки **D2:D6**, и мы имеем диапазон ячеек, которые содержат **TRUE**, если условия выполняются и **FALSE** в противном случае.

	A	B	C	D	E	F	G
1	красный	большой		ИСТИНА	=AND(A1="красный";B1="большой")		
2	синий	маленький		ЛОЖЬ			
3	зеленый	большой		ЛОЖЬ			
4	синий	маленький		ЛОЖЬ			
5	красный	большой		ИСТИНА			
6	красный	маленький		ЛОЖЬ			
7					2 =SUM(D1:D6)		
8							

В числовых вычислениях, **TRUE** рассматривается как 1, а **FALSE** — как 0. Таким образом введенная формула **=SUM(D1:D6)** просто суммирует эти единицы и нули, и дает нам количество элементов, которые соответствуют одновременно условиям **красный** И **большой**.

На самом деле, так как **TRUE** и **FALSE** оцениваются как 1 и 0, мы не нуждаемся в функции **AND** — в **D1** мы можем просто записать **=(A1=“красный”)*(B1=“большой”)**, и скопировать/вставить вниз в ячейки **D2:D6**.

Теперь пойдем дальше. Предположим, что диапазон **C1:C6** содержит список весов этих элементов, и мы желаем знать суммарную массу для всех больших и красных элементов. В ячейке **D1** мы запишем **=(A1=“красный”)*(B1=“большой”)*C1**, и скопируем/вставим эту формулу вниз в ячейки **D2:D6**. Ячейка **D1** будет содержать вес в **C1**, если условия будут выполняться (и ноль в противном случае) и так далее для ячеек **D2:D6**, следовательно **=SUM(D1:D6)** теперь даст нам суммарный вес.

	A	B	C	D	E	F	G
1	красный	большой	8	8	=(A1="красный")*(B1="большой")*C1		
2	синий	маленький	5	0			
3	зеленый	большой	2	0			
4	синий	маленький	11	0			
5	красный	большой	6	6			
6	красный	маленький	4	0			
7					14 =SUM(D1:D6)		
8							

В качестве альтернативы, можно заполнить диапазон **D1:D6** формулой массива. В ячейке **D1**, напишите **=(A1:A6=“красный”)*(B1:B6=“большой”)*C1:C6**, и введите, нажав **Ctrl-Shift-Enter**. Все ячейки в диапазоне **D1:D6** теперь показывают требуемый вес как ранее.

Функция **SUMPRODUCT**

Функция **SUMPRODUCT** может использоваться для выполнения вычислений подсчета и суммирования в предыдущем разделе, не используя дополнительные столбцы. Необходимо понять формулы массива, чтобы понять ее.

Используя пример суммирования из предыдущего раздела, **A1:A6**="красный", **B1:B6**="большой" и **C1:C6** можно рассмотреть как 3 отдельных массива, не отображаемых, но внутренне рассчитываемых.

=SUMPRODUCT(A1:A6="красный"; B1:B6="большой"; C1:C6), перемножит соответствующие элементы массивов вместе и возвратит их сумму, то есть:

(A1="red")*(B1="big")*C1 + (A2="red")*(B2="big")*C2 + ...

	А	В	С	Д	Е	F	G	
1	красный	большой	8					
2	синий	маленький	5					
3	зеленый	большой	2					
4	синий	маленький	11					
5	красный	большой	6					
6	красный	маленький	4					
7			14					
8			=SUMPRODUCT(A1:A6="красный"; B1:B6="большой"; C1:C6)					
9								

Это так-же дает нам суммарный вес, однако, дополнительный столбец не требуется.

Для переносимости в Excel, используйте формулу **=SUMPRODUCT((A1:A6="красный")*(B1:B6="большой")*C1:C6)**, поскольку Calc и Excel рассматривают логические результаты по-разному.

Заметьте, что формулы с использованием функции **SUMPRODUCT** просто вводятся нажатием клавиши *Enter* — они не требуют нажатия *Ctrl-Shift-Enter* даже при том, что связаны с массивами.

Кроме того, знайте, что вычисления, использующие очень большие массивы берут много компьютерного процессорного времени, и могут сильно замедлить обработку электронных таблиц.

Функция **SUM** с формулами массива

В качестве альтернативы функции **SUMPRODUCT** можно использовать функцию **SUM**. Предыдущий пример был бы записан следующим образом:

=SUM((A1:A6="красный")*(B1:B6="большой")*C1:C6)

и введен как формула массива, нажатием *Ctrl-Shift-Enter*. Как и в случае с функцией **SUMPRODUCT**, это работает, перемножая соответствующие элементы массивов вместе и возвращая их сумму.

Суммирование элементов с определенным форматированием

Функция **CELL** возвращает информацию о ячейках, например, что данные отображаются в формате числа или даты и ширину столбца. Не существует функции, которая возвращает цвет или шрифт ячейки. Если не существует самостоятельной формулы, имеющая отношение к цвету или шрифту, то необходимо использовать макрос.

Суммирование элементов, соответствующих элементам в отдельном списке

Предположим, что диапазон **A1:A9** содержит список дат, диапазон **B1:B9** — телефонные номера, а диапазон **C1:C9** — затраты на телефонные звонки. **F1:F5** — список определенных телефонных номеров, и Вы хотите знать общую стоимость звонков по этим номерам.

В ячейке D1 введем:

=ISNUMBER(MATCH(B1; F\$1:F\$5; 0))*C1

и скопируем и вставим в ячейки **D2:D9**. **=SUM(D1:D9)** теперь дает общую стоимость звонков.

	A	B	C	D	E	F	G
1	1 авг 08	720 528-1700	3,90	3,90		602 280-8600	
2	10 авг 08	616 975-0148	1,40	1,40		720 528-1700	
3	12 авг 08	901 271-3704	2,60	0,00		801 257-6400	
4	30 авг 08	602 280-8600	1,50	1,50		502 657-6033	
5	3 сен 08	614 719-5900	0,80	0,00		616 975-0148	
6	16 сен 08	801 257-6400	3,70	3,70			
7	18 сен 08	602 280-8600	5,20	5,20	=ISNUMBER(MATCH(B7; F\$1:F\$5; 0))*C7		
8	2 окт 08	614 719-5900	1,60	0,00			
9	8 окт 08	801 257-6400	1,20	1,20			
10				16,90	=SUM(D1:D9)		
11							

Чтобы выполнять это вычисление без дополнительного столбца, можно использовать:

=SUMPRODUCT(NOT(ISERROR(MATCH(B1:B9; F\$1:F\$5; 0))); C1:C9)

Мы обсуждали выше, как может быть важно проверить параметры настройки Calc; это — хороший пример — если пользователь имеет включенные регулярные выражения (по умолчанию), номер телефона, записанный как (720) 528-1700 интерпретируется как регулярное выражение и может соответствовать записям иным, чем (720) 528-1700.

Функции комплексных чисел

Комплексное число (иногда также называемое мнимым числом) — пара вещественных чисел, включающая действительную и мнимую части. В математике, комплексные числа часто записываются в виде $x + iy$, где x (действительная часть), а y (мнимая часть) — вещественные числа, а i — $\sqrt{-1}$. Такая запись называется *алгебраической формой* комплексного числа.

Комплексное число может быть также записано в виде (в *тригонометрической форме*)

$$z = r e^{i\theta} = r(\cos \theta + i \sin \theta)$$

где: r — модуль комплексного числа (вещественное число), а θ — аргумент или фаза (вещественное число, представляющее угол в радианах).

Если на плоскости по оси абсцисс расположить действительную часть, а по оси ординат — мнимую, то комплексному числу будет соответствовать точка с декартовыми координатами x и y (или её радиус-вектор, что то же самое), а модуль и аргумент будут полярными координатами этой точки. Такая плоскость называется комплексной.

Во многих технических областях вместо i используется буква j ; j легче отличить от цифры 1. Комплексные числа широко используются в математике, математической физике, в естественных науках — гидродинамике, картографии, квантовой механике, теории колебаний, а также в инженерных расчетах.

Следующие функции обрабатывают комплексные числа. Функция **COMPLEX** действует как доступный конструктор комплексных чисел; другие функции принимают и/или создают комплексные числа. Заметим, что **SQRT** возвращает ошибку, когда получает отрицательное число, в то время как **IMSQRT** вернет комплексное число с учетом отрицательного числа.

История

Впервые, по-видимому, мнимые величины появились в известном труде «Великое искусство, или об алгебраических правилах» Кардано (1545), который счёл их непригодными к употреблению. Пользу мнимых величин, в частности, при решении кубического уравнения, в так называемом неприводимом случае (когда вещественные корни многочлена выражаются через кубические корни из мнимых величин), впервые оценил Бомбелли (1572). Он же дал некоторые простейшие правила действий с комплексными числами.

Выражения вида $a + b\sqrt{-1}$, появляющиеся при решении квадратных и кубических уравнений, стали называть «мнимыми» в XVI-XVII веках, однако даже для многих крупных ученых XVII века алгебраическая и геометрическая сущность мнимых величин представлялась неясной. Лейбниц, например, писал: «Дух божий нашел тончайшую отдушину в этом чуде анализа, уроде из мира идей, двойственной сущности, находящейся между бытием и небытием, которую мы называем мнимым корнем из отрицательной единицы»¹.

Долгое время было неясно, все ли операции над комплексными числами приводят к комплексным результатам, или, например, извлечение корня может привести к открытию какого-то нового типа чисел. Задача о выражении корней степени n из данного числа была решена в работах Муавра (1707) и Котса (1722).

Символ $i = \sqrt{-1}$ предложил Эйлер (1777, опубл. 1794), взявший для этого первую букву слова *imaginarium*. Он же распространил все стандартные функции, включая логарифм, на комплексную область. Эйлер также высказал в 1751 году мысль об алгебраической замкнутости поля комплексных чисел. К такому же выводу пришел Д'Аламбер (1747), но

¹ Клайн М. Математика. Утрата определённости. — М.: Мир, 1984. — С. 139.

первое строгое доказательство этого факта принадлежит Гауссу (1799). Гаусс и ввёл в широкое употребление термин «комплексное число» в 1831 году, хотя этот термин ранее использовал в том же смысле французский математик Лазар Карно в 1803 году.

Геометрическое истолкование комплексных чисел и действий над ними появилось впервые в работе Весселя (англ.), (1799). Первые шаги в этом направлении были сделаны Валлисом (Англия) в 1685 году. Современное геометрическое представление, иногда называемое «диаграммой Аргана», вошло в обиход после опубликования в 1806-м и 1814-м годах работы (Аргана (фр.)), повторявшей независимо выводы Весселя.

Арифметическая модель комплексных чисел как пар вещественных чисел была построена Гамильтоном (1837); это доказало непротиворечивость их свойств. Гамильтон предложил и обобщение комплексных чисел — кватернионы, алгебра которых некоммутативна.

COMPLEX

Функция **COMPLEX** возвращает комплексное число, с учетом действительной и мнимой частей.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=COMPLEX(realpart; imaginarypart; suffix)

где:

- **realpart** — действительная часть комплексного числа;
- **imaginarypart** — мнимая часть комплексного числа;
- **suffix** — необязательный текст **i** или **j** (в нижнем регистре) для указания мнимой части комплексного числа; по умолчанию — **i**. Имейте в виду, что **I** или **J**, не принимаются для суффикса.

Функция возвращает комплексное число в виде текста, в форме **a + bi** или **a + bj**.

	A	B	C	D	E
2					
3		3+4i	=COMPLEX(3; 4; "i")		
4		2-3j	=COMPLEX(2; -3; "j")		
5					

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMABS

Функция **IMABS** возвращает абсолютное значение комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMABS(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

IMABS возвращает абсолютное значение или модуль комплексного числа **complexnumber** — то есть, в полярном представлении, расстояние от начала координат. Если **complexnumber** — **a + bi**, то абсолютное значение — $\sqrt{a^2 + b^2}$; если **complexnumber** — $r(\cos \varphi + i \sin \varphi)$, то

абсолютное значение — r .

	A	B	C	D
1				
2		5	=IMABS("4+3i")	
3				

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

Excel использует **COMPLEX(0;0)** как значение по умолчанию для пустой ячейки, в то время как Calc в таких случаях возвращает ошибку.

IMAGINARY

Функция **IMAGINARY** возвращает мнимую часть комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMAGINARY(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a+bi** или **a+bj**, где **a** — действительная часть, **b** — мнимая часть.

	A	B	C	D
1				
2		3	=IMAGINARY("4+3i")	
3				

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

Excel использует **COMPLEX(0;0)** как значение по умолчанию для пустой ячейки, в то время как Calc в таких случаях возвращает ошибку.

IMARGUMENT

Функция **IMARGUMENT** возвращает аргумент комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMARGUMENT(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

IMARGUMENT возвращает аргумент комплексного числа **complexnumber** в радианах — то есть, в полярном представлении, угол относительно горизонтальной оси. Для комплексного числа $a + bi = r(\cos \varphi + i \sin \varphi)$ аргумент — φ .

	A	B	C	D	E
1					
2		0,7853982	=IMARGUMENT("3+3i")		
3					

Возвращается **0,785398163397** — что соответствует **$2\pi/8$** радиана.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера

функции.

И Excel и Calc возвращают ошибку, когда **complexnumber** ссылается на пустую ячейку.

IMCONJUGATE

Функция **IMCONJUGATE** возвращает комплексно-сопряженное комплексное число.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMCONJUGATE(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

Функция **IMCONJUGATE** возвращает комплексно-сопряженное комплексное число для **complexnumber** — то есть, если **complexnumber** — $a + bi$, она возвращает $a - bi$, в виде текста.

	A	B	C	D	E
1					
2		2-3i	=IMCONJUGATE("2+3i")		
3					

Возвращается **2-3i** в виде текста.

Эта функция в настоящее время появляется в категории Подключаемый модуль мастера функции.

Excel использует **COMPLEX(0;0)** как значение по умолчанию для пустой ячейки, в то время как Calc в таких случаях возвращает ошибку.

IMCOS

Функция **IMCOS** возвращает косинус комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMCOS(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

IMCOS возвращает косинус **complexnumber** в виде текста — то есть, если **complexnumber** — $a + ib$, она возвращает $\cos(a)\cosh(b) - i\sin(a)\sinh(b)$.

	A	B	C	D
1				
2		-4.18962569096881-9.10922789375534i	=IMCOS("2+3i")	
3				

Возвращается **-4.18962569096881-9.10922789375534i** в виде текста.

Эта функция в настоящее время появляется в категории Подключаемый модуль мастера функции.

Excel использует **COMPLEX(0;0)** как значение по умолчанию для пустой ячейки, в то время как Calc в таких случаях возвращает ошибку.

IMDIV

Функция **IMDIV** возвращает результат от деления одного комплексного числа на другое.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMDIV(complexnum1; complexnum2)

где:

- **complexnum1** и **complexnum2** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

IMDIV возвращает **complexnum1/complexnum2** в виде текста. Частное двух комплексных чисел $X = a + bi$ и $Y = c + di$, определяется следующим образом:

$$\text{IMDIV}(X; Y) = \frac{a + bi}{c + di} = \frac{(ac + bd) + (bc - ad)i}{(c^2 + d^2)}$$

Деление на ноль возвращает ошибку.

	A	B	C	D	E
1					
2			5-2i	=IMDIV("9+8i"; "1+2i")	
3					

Возвращается **5-2i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMEXP

Функция **IMEXP** возвращает экспоненту комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMEXP(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

IMEXP возвращает математическую константу e возведенную в степень **complexnumber**. Результат — комплексное число, представленное в виде текста.

Если **complexnumber** — **a + bi**, экспонента комплексного числа определяется следующим образом:

$$e^{a+bi} = e^a e^{bi} = e^a (\cos(b) + i \sin(b))$$

	A	B	C	D
1				
2			-1.13120438375681+2.47172667200482i	=IMEXP("1+2i")
3				

Возвращается **-1.13120438375681+2.47172667200482i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMLN

Функция **IMLN** возвращает натуральный логарифм комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMLN(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

	A	B	C	D
1				
2		0.80471895621705+1.10714871779409i =IMLN("1+2i")		
3				

Возвращается **0.80471895621705+1.10714871779409i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMLOG10

Функция **IMLOG10** возвращает десятичный логарифм комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMLOG10(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

	A	B	C	D
1				
2		0.349485002168009+0.480828578784234i =IMLOG10("1+2i")		
3				

Возвращается **0.349485002168009+0.480828578784234i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMLOG2

Функция **IMLOG2** возвращает двоичный логарифм комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMLOG2(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

	A	B	C	D
1				
2		1.16096404744368+1.59727796468811i =IMLOG2("1+2i")		
3				

Возвращается **1.16096404744368+1.59727796468811i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMPOWER

Функция **IMPOWER** возвращает комплексное число возведенное в степень.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMPOWER(complexnumber; number)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**;
- **number** — число.

Если **complexnumber** — $a+ib=r(\cos\varphi+i\sin\varphi)$, а **number** — n , то **IMPOWER(complexnumber; number)** возвращает $(a+ib)^n=r^n(\cos(n\varphi)+i\sin(n\varphi))$.

	A	B	C	D
1				
2		-3+4i =IMPOWER("1+2i"; 2)		
3				

Возвращается $(1 + 2i)^2 = -3 + 4i$ в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMPRODUCT

Функция **IMPRODUCT** возвращает произведение комплексных чисел.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMPRODUCT(complexnum1; complexnum2; ... complexnum29)

где:

- **complexnum1, complexnum2 ...** — комплексные числа в виде текста, или ссылки на ячейки/диапазоны ячеек, содержащие комплексные числа в виде текста.

Заданы два комплексных числа $X = a + bi$ и $Y = c + di$, произведение двух комплексных чисел определяется следующим образом:

$$X * Y = (ac - bd) + (ad + bc)i$$

	A	B	C	D	E
1					
2		14-8i =IMPRODUCT("2-3i"; "4+2i")			
3					

Возвращается **14-8i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMREAL

Функция **IMREAL** возвращает действительную часть комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMREAL(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**, где **a** - действительная часть, а **b** — мнимая часть.

	A	B	C	D
1				
2		4	=IMREAL("4+3i")	
3				

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMSIN

Функция **IMSIN** возвращает синус комплексного числа.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMSIN(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

IMSIN возвращает синус комплексного числа **complexnumber**, в виде текста — то есть, если **complexnumber** — **a+bi**, она возвращает $\sin(a) \cosh(b) - \cos(a) \sinh(b)i$.

	A	B	C	D
1				
2		9.15449914691143-4.16890695996656i	=IMSIN("2+3i")	
3				

Возвращается **9.15449914691143-4.16890695996656i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

Excel использует **COMPLEX(0;0)** как значение по умолчанию для пустой ячейки, в то время как Calc в таких случаях возвращает ошибку.

IMSUB

Функция **IMSUB** возвращает разность между 2 комплексными числами.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMSUB(complexnum1; complexnum2)

где:

- **complexnum1** и **complexnum2** — текст, представляющий комплексные числа в виде текста, или ссылки на ячейки, содержащие комплексные числа в виде текста.

Разность двух комплексных чисел $X = a + bi$ и $Y = c + di$, определяется следующим образом:

$$X - Y = (a + bi) - (c + di) = (a - c) + (b - d)i$$

	A	B	C	D	E
1					
2		4+3i	=IMSUB("7+5i"; "3+2i")		
3					
4		3+4i			
5		1+2i			
6		2+2i	=IMSUB(B4;B5)		
7					

В строке 2 возвращается **4+3i** в виде текста. В строке 6 возвращается **2+2i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

IMSUM

Функция **IMSUM** возвращает сумму комплексных чисел.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=IMSUM(complexnum1; complexnum2; ... complexnum29)

где:

- **complexnum1**, **complexnum2** ... — комплексные числа в виде текста, или ссылки на ячейки/диапазоны ячеек, содержащие комплексные числа в виде текста.

Текст, который не может быть преобразован к комплексному числу, игнорируется.

Сумма двух комплексных чисел $X = a + bi$ и $Y = c + di$, определяется следующим образом:

$$X + Y = (a + bi) + (c + di) = (a + c) + (b + d)i$$

	A	B	C	D	E
1					
2		4+6i	=IMSUM("1+2i"; "3+4i")		
3					
4					
5		1+2i			
6		3+4i			
7		4+6i	=IMSUM(B5;B6)		
8					

В обоих случаях возвращается **4+6i** в виде текста.

Эта функция в настоящее время появляется в категории *Подключаемый модуль* мастера функции.

И Excel и Calc используют **COMPLEX(0;0)** в качестве значения по умолчанию для пустой ячейки.

IMSQRT

Функция **IMSQRT** возвращает квадратный корень комплексного числа.

Эта функция доступна, только если установлен пакет *Анализа*.

Синтаксис функции:

=IMSQRT(complexnumber)

где:

- **complexnumber** — текст, представляющий комплексное число, например в виде **a + bi** или **a + bj**.

IMSQRT возвращает квадратный корень комплексного числа **complexnumber** в виде текста. Таким образом, если **complexnumber** — $a_0 + b_0i$, она возвращает $a_1 + b_1i$, такое что $(a_1 + b_1i) * (a_1 + b_1i) = a_0 + b_0i$.

Если **complexnumber** представить в виде $X = r(\cos \varphi + i \sin \varphi)$, то квадратный корень определяется следующим образом:

$$\text{IMSQRT}(X) = \sqrt{r} \left(\cos\left(\frac{\varphi}{2}\right) + i \sin\left(\frac{\varphi}{2}\right) \right)$$

где $\varphi \in [-\pi; \pi]$

	A	B	C	D
1				
2		5+3i	=IMSQRT("16+30i")	
3				

Эта функция в настоящее время появляется в категории *Подключаемый модуль мастера функции*.

Статистические функции

Статистические функции выполняют статистический анализ данных, они сообщают информацию о наборах чисел. Некоторые функции, которые также можно было бы считать статистическими, такие как **SUM**, приведены в других разделах.

AVEDEV

Функция **AVEDEV** возвращает среднее (арифметическое) абсолютных отклонений значений от их среднего.

Синтаксис функции:

=AVEDEV(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

Функция **AVEDEV** вычисляет средние чисел, потом абсолютное (положительный знак) отклонение каждого числа от этого среднего. Она возвращает среднее число тех отклонений. Используется формула:

$$\frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$$

	A	B	C	D	E
1					
2					
3					

2 = AVEDEV(1; 4; 7)

В ячейке **B2** возвращается **2**. Среднее — **4**, а абсолютные отклонения от среднего — **3**, **0** и **3**, соответственно.

AVERAGE

Функция **AVERAGE** возвращает среднее (арифметическое) аргументов, игнорируя текст.

Синтаксис функции:

=AVERAGE(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

Среднее, также называемое *средним арифметическим*, а также называемое *средним числом*, вычисляется математически сложением значений, а затем делением на количество значений (размер выборки n).

$$\frac{1}{n} \sum_{i=1}^n x_i$$

Сумма данных $\sum x_i$ может быть определена с использованием функции **=SUM(data)**. Размер выборки n может быть определен с использованием функции **=COUNT(data)**. Таким образом **=SUM(data)/COUNT(data)** вычислит среднее.

	A	B	C	D	E	F
1						
2		1	4	=AVERAGE(2; 6; 4)		
3		3				
4	яблоко		2	=AVERAGE(A2:A4)		
5						

В ячейке **C2** возвращается **4**, среднее арифметическое этих трех чисел в списке.

В ячейке **C4** возвращается **2, 2**, среднее арифметическое значений диапазона **A2:A4** — чисел **1** и **3**. Текст игнорируется.

Calc предполагает, что логические значения **1 (TRUE)** и **0 (FALSE)**. Excel игнорирует логические значения.

AVERAGEA

Функция **AVERAGEA** возвращает среднее (арифметическое) аргументов, включая текст (оцениваемый как 0).

Синтаксис функции:

=AVERAGEA(value1; value2; ... value30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, которые могут включить числа, текст и логические значения. Текст оценивается как **0**. Логические значения оцениваются как **1 (TRUE)** и **0 (FALSE)**.

	A	B	C	D	E	F
1						
2		4	4	=AVERAGEA(2; 6; 4)		
3		2				
4	яблоко		2	=AVERAGEA(A2:A4)		
5						
6			1,5	=AVERAGEA(2; TRUE())		
7						

В ячейке **C2** возвращается **4**, среднее арифметическое этих трех чисел в списке.

В ячейке **C4** возвращается **2**, среднее арифметическое значений диапазона **A2:A4** — чисел **4**, **2** и **0**.

В ячейке **C6** возвращается **1,5**, среднее арифметическое чисел **2** и **1**.

B

Функция **B** вычисляет вероятности для биномиального распределения².

Синтаксис функции:

=B(n; p; k; k2)

где:

- **n** — число независимых случайных экспериментов;
- **p** — вероятность “успеха” в каждом из случайных экспериментов;
- **k; k2** — нижний и верхний пределы диапазона распределения для **n** независимых

² *Биномиальное распределение* — (распределение Бернулли), распределение вероятностей числа появлений некоторого события при повторных независимых испытаниях, если вероятность появления этого события в каждом испытании равна *p*.

случайных экспериментов, $k2 = k$ если $k2$ опущен.

Для n независимых случайных экспериментов, каждый с вероятностью успеха p , функция **B** возвращает вероятность того, что число успехов будет между k и $k2$ включительно, или если $k2$ опущен, вероятность того, что число успехов будет точно k .

Вероятность вычисляется как:

$$\sum_{i=k}^{k2} \frac{n!}{i!(n-i)!} p^i (1-p)^{n-i}$$

где $k2 = k$ если $k2$ опущен.

	A	B	C	D	E
1					
2		0,29	=B(10; 1/6; 2)		
3					

В ячейке **B2** возвращается приблизительно **0,29** (29 %), вероятность того, что номер четыре будет выброшен точно два раза за 10 бросков игральной кости.

BETADIST

Функция **BETADIST** вычисляет интегральную функцию бета-распределения³.

Синтаксис функции:

=BETADIST(n; a; b; a; b)

где:

- **n** — значение в интервале между **a** и **b**, для которого вычисляется функция;
- **a** и **b** — параметры распределения;
- **a** — необязательная нижняя граница интервала изменения **n** (по умолчанию, если опущена, равна 0);
- **b** — необязательная верхняя граница интервала изменения **n** (по умолчанию, если опущена, равна 1).

Бета распределение — семейство непрерывных распределений вероятности, определенных для интервала от **a** до **b**, где **a** и **b** — параметры, управляющие формой распределения, а **n** находится между **a** и **b**.

Функция **BETADIST** для $n \geq a$ и $n \leq b$ вычисляет интегральную функцию распределения:

$$\int_a^n \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot \left(\frac{x-a}{b-a}\right)^{\alpha-1} \left(1-\left(\frac{x-a}{b-a}\right)\right)^{\beta-1} dx$$

	A	B	C	D	E
1					
2		0,96	=BETADIST(0,75; 3; 4)		
3					

BETAINV

Функция **BETAINV** вычисляет обратную интегральную функцию указанного бета-распределения, инверсию функции **BETADIST**.

Синтаксис функции:

³ *Бета распределение* в теории вероятностей и статистике — двухпараметрическое семейство абсолютно непрерывных распределений.

=BETAINV(p; α; β; a; b)

где:

- **p** — вероятность, связанная с бета-распределением;
- **α** и **β** — параметры распределения;
- **a** — необязательная нижняя граница интервала изменения **x** (по умолчанию, если опущена, равна 0);
- **b** — необязательная верхняя граница интервала изменения **x** (по умолчанию, если опущена, равна 1).

Бета распределение — семейство непрерывных распределений вероятности, определенных для интервала от **a** до **b**, где **α** и **β** — параметры, управляющие формой распределения, а **p** находится между **a** и **b**.

Функция **BETAINV** возвращает уникальное значение **p**, лежащее в закрытом интервале от **a** до **b**, из условия что **BETADIST(n; α; β; a; b) = p**.

	A	B	C	D	E
1					
2			0,75	=BETAINV(0,96; 3; 4)	
3					

BINOMDIST

Функция **BINOMDIST** вычисляет вероятности для биномиального распределения.

Синтаксис функции:

=BINOMDIST(k; n; p; mode)

где:

- **k** — количество успешных независимых случайных экспериментов;
- **n** — число независимых случайных экспериментов;
- **p** — вероятность “успеха” в каждом из случайных экспериментов;
- **mode** — логическое значение, определяющее форму функции.

Для **n** независимых случайных экспериментов, каждый с вероятностью успеха **p**, функция **BINOMDIST** возвращает вероятность того, что число успехов будет:

- точно **k**, если **mode** — 0.
- до (и включая) **k**, если **mode** — 1.

BINOMDIST(k; n; p; 0) эквивалентно **B(n; p; k)**; **BINOMDIST(k; n; p; 1)** эквивалентно **B(n; p; 0; k)**. См. **B** для получения более подробной информации.

	A	B	C	D	E
1					
2			0,05	=BINOMDIST(3; 12; 0,5; 0)	
3					
4			0,07	=BINOMDIST(3; 12; 0,5; 1)	
5					

В ячейке **C2** возвращается приблизительно **0,05** (5 %), вероятность того, что орел выпадет точно 3 раза при 12 бросках монеты.

В ячейке **C4** возвращается приблизительно **0,07** (7 %), вероятность того, что орел выпадет 0, 1, 2 или 3 раза при 12 бросках монеты.

CHIDIST

Функция **CHIDIST** вычисляет значения для χ^2 -распределения⁴.

Синтаксис функции:

=CHIDIST(x; k)

где:

- **x** — число ≥ 0 , для которого требуется вычислить распределение;
- **k** — (положительное целое число) степени свободы для χ^2 -распределения.

Функция **CHIDIST** возвращает вероятность правого “шлейфа” для χ^2 -распределения:

$$\int_x^{\infty} \frac{t^{\frac{k}{2}-1} e^{-\frac{t}{2}}}{2^{\frac{k}{2}} \Gamma\left(\frac{k}{2}\right)} dt$$

	A	B	C	D	E
1					
2			0,5724	=CHIDIST(2; 3)	
3					

Проект международного стандарта ODFD называет эту функцию **LEGACY.CHIDIST**, и определяет новую функцию **CHISQDIST**. Функция **CHISQDIST** ожидается в OOo 3.1.

Проект международного стандарта ODFD не имеет ограничений для **x**, но определяет, что функция **LEGACY.CHIDIST** возвращает 1 для **x** ≤ 0 .

CHIINV

Функция **CHIINV** вычисляет инверсию функции **CHIDIST**.

Синтаксис функции:

=CHIINV(p; k)

где:

- **p** — вероятность, связанная с χ^2 -распределением;
- **k** — (положительное целое число) степени свободы для χ^2 -распределения.

Функция **CHIINV(p; k)** возвращает значение **x**, такое что **CHIDIST(x; k) = p**.

	A	B	C	D	E
1					
2			2,00	=CHIINV(0,5724; 3)	
3					

Проект международного стандарта ODFD называет эту функцию **LEGACY.CHIINV**, а также определяет новую функцию **CHISQINV**.

CHISQDIST

Функция **CHISQDIST** вычисляет значения для χ^2 -распределения. Эта функция еще не реализована, но ожидается в OpenOffice.org версии 3.1.

Синтаксис функции:

⁴ *Распределение χ^2 (хи-квадрат) с n степенями свободы — это распределение суммы квадратов n независимых стандартных нормальных случайных величин.*

=CHISQDIST(x; k; Cumulative)

где:

- **x** — число ≥ 0 , для которого требуется вычислить χ^2 -распределение;
- **k** — (положительное целое число) степени свободы для χ^2 -распределения;
- **Cumulative** — логическое значение.

В случае **Cumulative=TRUE()** используется интегральная функция распределения, в случае **Cumulative=FALSE()** — функция плотности вероятности. Этот параметр является необязательным, он устанавливается в **TRUE()**, если отсутствует.

CHISQDIST(x;k;FALSE()) возвращает значение функции плотности вероятности для χ^2 -распределения:

$$= \begin{cases} 0 & \text{если } x < 0 \\ \frac{\frac{k}{2} x^{\frac{k}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{k}{2}} \Gamma(\frac{k}{2})} & \text{если } x \geq 0 \end{cases}$$

CHISQDIST(x;k;TRUE()) возвращает вероятность левого “шлейфа” для χ^2 -распределения:

$$= \begin{cases} 0 & \text{если } x < 0 \\ \int_0^x \frac{t^{\frac{k}{2}-1} e^{-\frac{t}{2}}}{2^{\frac{k}{2}} \Gamma(\frac{k}{2})} dt & \text{если } x \geq 0 \end{cases}$$

	A	B	C	D	E
1					
2		0,1699395239	=CHISQDIST(2,7;3;FALSE())		
3					
4		0,5597727056	=CHISQDIST(2,7;3;TRUE())		
5					

Если вам требуется **CHISQDIST(x; k; TRUE())** с не целым параметром **k**, используйте вместо нее **GAMMADIST(x; k/2; 2)**.

В случае плотности внутренние расчеты используют логарифмические и показательные функции, если $x > 1425$ или $x \cdot k > 1\,391\,000$. Полученные результаты являются менее точными в этих случаях.

Эта функция не доступна в версии 3.0 и ранее.

CHITEST

Функция **CHITEST** возвращает меру “критерия согласия” χ^2 .

Синтаксис функции:

=CHITEST(observed; expected)

где:

- **observed** — диапазон или множество наблюдаемых значений;
- **expected** — диапазон или множество ожидаемых значений.

Диапазоны **observed** и **expected** должны иметь одинаковое количество строк и столбцов и должно быть по крайней мере 2 значения в каждом.

Функция **CHITEST** вычисляет статистику χ^2 и степени свободы, а затем вызывает функцию **CHIDIST**.

	A	B	C	D	E
1					
2			0,969	=CHITEST({8;9;7;8}; {8;8;8;8})	
3					

Проект международного стандарта ODFD называет эту функцию **LEGACY.CHITEST**, и указывает, что она на то, что она имеет ограниченное математическое действие. Функция **CHITEST** совместима с Excel функцией **CHITEST**.

CONFIDENCE

Функция **CONFIDENCE** возвращает доверительный интервал.

Доверительный интервал в математической статистике — это интервал, построенный с помощью случайной выборки из распределения с неизвестным параметром, такой, что он содержит данный параметр с заданной вероятностью.

Синтаксис функции:

=CONFIDENCE(α ; sd; size)

где:

- **α** — уровень значимости ($0 < \alpha < 1$), который определяет требуемый уровень доверия **=(1 - α)*100 %**. Таким образом, **$\alpha = 0,05$** дает 95%-ый уровень доверия;
- **sd** — (известное) среднееквадратичное отклонение нормального распределения;
- **size** — размер выборки из этого распределения.

Функция **CONFIDENCE** возвращает значение, которое когда добавляется и вычитается из выборочного среднего дает интервал, в пределах которого, как ожидают, будет располагаться среднее значение генеральной совокупности с указанным уровнем доверия.

Функция **CONFIDENCE** вычисляется следующим образом:

$$\frac{\text{NORMINV}(1 - \alpha/2; 1; 0) * sd}{\sqrt{size}}$$

	A	B	C	D	E
1					
2			0,4383	=CONFIDENCE(0,05;1;20)	
3					

В ячейке **C2** возвращается приблизительно **0,4383**. С 95%-ым уровнем доверия, среднее значение генеральной совокупности находится в пределах $\pm 0,4383$ от выборочного среднего.

CORREL

Функция **CORREL** возвращает коэффициент корреляции Пирсона двух наборов данных.

Синтаксис функции:

=CORREL(x; y)

где:

- **x** и **y** — диапазоны или массивы, содержащие два набора данных.

Любой текст или пустые записи игнорируются.

Функция **CORREL** вычисляет:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

где \bar{x} и \bar{y} — среднее x и y .

Параметры x и y всегда оцениваются как формулы массива.

=CORREL(A1:A30; B1:B30)

возвращает коэффициент корреляции Пирсона для двух наборов данных в диапазонах **A1:A30** и **B1:B30**.

Функция **CORREL** идентична функции **PEARSON**.

COUNT

Функция **COUNT** подсчитывает количество числовых значений в списке аргументов, игнорируя текстовые элементы.

Синтаксис функции:

=COUNT(value1; value2; ... value30)

где:

- **number1; number2; ... number30** — до 30 значений или диапазонов, представляющих значения, которые будут подсчитываться.

Подсчитываются только числа в ссылках; все другие типы данных игнорируются. Ошибки не размножаются.

	A	B	C	D	E	F
1						
2		4	3	=COUNT(2; 4; 6; "восемь")		
3		2				
4	яблоко		2	=COUNT(A2:A4)		
5						

В ячейке **C2** возвращается **3**, потому что **2**, **4** и **6** — числа, а **“восемь”** — текст.

В ячейке **C4** возвращается **2**.

COUNTA

Функция **COUNTA** подсчитывает количество непустых значений в списке аргументов.

Синтаксис функции:

=COUNTA(value1; value2; ... value30)

где:

- **number1; number2; ... number30** — до 30 значений или диапазонов, представляющих значения, которые будут подсчитываться.

Подсчитывает количество непустых значений в списке. Значение считается непустым, если оно содержит что-либо любого типа, включая ошибку. В ссылке, подсчитывается каждая ячейка, которая не является пустой. Значение пустая строка (“”) не считается пустым. Ошибки, содержащиеся в диапазоне считают непустым значением при подсчете; ошибки не размножаются. Постоянные выражения или формулы допускаются; они вычисляются и если они производят значение ошибки, значение ошибки подсчитывается как одно непустое значение (и не размножается как ошибка).

	A	B	C	D	E	F
1	1,1					
2	#1/Д		3	=COUNTA(A1:A3)		
3	яблоко					
4			1	=COUNTA(D1)		
5						

В ячейке **C2** возвращается **3**, потому что ни одна из ячеек в диапазоне **A1:A3** не пуста.

В ячейке **C4** возвращается **1**, потому что ячейка **D1** содержит “”, а пустая строка является строкой (хотя и нулевой длины).

COVAR

Функция **COVAR** возвращает ковариацию⁵ двух наборов данных.

Синтаксис функции:

=COVAR(x; y)

где:

- **x** и **y** — диапазоны или массивы, содержащие два набора данных.

Функция **COVAR** вычисляет:

$$\sum_i (x - \bar{x})(y - \bar{y})$$

где \bar{x} и \bar{y} — среднее **x** и **y**.

Параметры **x** и **y** всегда оцениваются как формулы массива.

=COVAR(A1:A30; B1:B30)

возвращает ковариацию двух наборов данных в диапазонах **A1:A30** и **B1:B30**.

DEVSQ

Функция **DEVSQ** возвращает сумму квадратов отклонений от среднего.

Синтаксис функции:

=DEVSQ(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

Функция **DEVSQ** вычисляет средние всех чисел, затем суммирует квадраты отклонений каждого числа от этого среднего. Для **N** значений, формула вычисления:

$$\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$

	A	B	C	D	E
1					
2			8	=DEVSQ(1; 3; 5)	
3					

В ячейке **B2** возвращается **8**, вычисляемое как $(-2)^2 + 0 + (2)^2$.

⁵ Ковариация в теории вероятностей — это мера линейной зависимости случайных величин.

EXPONDIST

Функция **EXPONDIST** вычисляет значения для экспоненциального распределения.

Экспоненциальное распределение — абсолютно непрерывное распределение, моделирующее время между двумя последовательными свершениями одного и того же события.

Синтаксис функции:

=EXPONDIST(x; λ; mode)

где:

- **x** — значение, для которого вычисляется функция;
- **λ** — интенсивность или обратный коэффициент масштаба ($\lambda > 0$);
- **mode** — логическое значение, которое указывает, какую форму экспоненциальной функции использовать.

Если **mode** = 0, **EXPONDIST** вычисляет функцию плотности вероятности экспоненциального распределения:

$$\lambda e^{-\lambda x}$$

Если **mode** = 1, **EXPONDIST** вычисляет интегральную функцию распределения показательного распределения:

$$1 - e^{-\lambda x}$$

	A	B	C	D	E
1					
2			1	=EXPONDIST(0; 1; 0)	
3					
4			0	=EXPONDIST(0; 1; 1)	
5					

FDIST

Функция **FDIST** вычисляет значения для F-распределения.

Синтаксис функции:

=FDIST(x; r1; r2)

где:

- **x** — значение, для которого вычисляется функция, должно быть больше или равно 0;
- **r1** и **r2** — степени параметров свободы для F-распределения, положительные целые числа.

Функция **FDIST** возвращает область правого “шлейфа” функции плотности вероятности для F-распределения, вычисляя:

$$\frac{\Gamma\left(\frac{r_1+r_2}{2}\right)\left(\frac{r_1}{r_2}\right)^{\frac{r_1}{2}}}{\Gamma\left(\frac{r_1}{2}\right)\Gamma\left(\frac{r_2}{2}\right)} \int_x^{\infty} \frac{t^{\frac{r_1}{2}-1}}{\left(1+\frac{r_1}{r_2}t\right)^{\frac{(r_1+r_2)}{2}}} dt$$

	A	B	C	D	E
1					
2			0,485657	=FDIST(1; 4; 5)	
3					

Разрабатываемый международный стандартный ODFB называет эту функцию **LEGACY.FDIST**, и определяет новую функцию **FDIST**.

FINV

Функция **FINV** вычисляет инверсию функции **FDIST**.

Синтаксис функции:

=FINV(p; r1; r2)

где:

- **p** — вероятность, связанная с F-распределением, должна быть больше чем 0 и меньше или равна 1;
- **r1** и **r2** — степени параметров свободы для F-распределения, положительные целые числа.

Функция **FINV** возвращает уникальное неотрицательное значение **x**, такое что:

FDIST(x; r1; r2) = p.

	A	B	C	D	E
1					
2			1	=FINV(0,485657; 4; 5)	
3					

Разрабатываемый международный стандартный ODFB называет эту функцию **LEGACY.FINV**, и определяет новую функцию **FINV**.

FISHER

Функция **FISHER** вычисляет значения для преобразования Фишера⁶.

Синтаксис функции:

=FISHER(r)

где:

- **r** — числовое значение, которое требуется преобразовать, $(-1 < r < 1)$.

Функция **FISHER** возвращает значение преобразования Фишера для аргумента **r**. Эта функция вычисляет:

$$\frac{1}{2} \ln \left(\frac{1+r}{1-r} \right)$$

	A	B	C	D	E
1					
2			0	=FISHER(0)	
3					

FISHERINV

Функция **FISHERINV** вычисляет обратное преобразования Фишера.

Синтаксис функции:

=FISHERINV(z)

⁶ В статистике, гипотезы о значении коэффициента корреляции в совокупности ρ между переменными X и Y основной генеральной совокупности, могут быть проверены с использованием *преобразования Фишера*, примененного к выборочной корреляции r .

где:

- **z** — значение, для которого выполняется обратное преобразование.

Функция **FISHERINV** возвращает значение **r**, такое что **FISHER(r) = z**. Эта функция вычисляет:

$$\frac{e^{2x}-1}{e^{2x}+1}$$

	A	B	C	D	E
1					
2			0	=FISHERINV(0)	
3					

FORECAST

Функция **FORECAST** подбирает прямую линию к данным, используя линейную регрессию и возвращает точку на этой линии.

=FORECAST(new_xvalue; yvalues; xvalues)

где:

- **new_xvalue** — координата x точки данных, для которой возвращается значение;
- **yvalues** и **xvalues** — одностроковые или одностолбцовые диапазоны, определяющие точки в наборе данных.

Диапазоны **yvalues** и **xvalues** должны быть одинакового размера.

Функция **FORECAST** возвращает значение y на найденной прямой линии, соответствующее значению x в **new_xvalue**. Это эквивалентно функции **TREND(yvalues; xvalues; new_xvalue)**.

Параметры **xvalues** и **yvalues** всегда оцениваются как формулы массива. Заметьте, что функция **TREND** не оценивает свои параметры как формулы массива.

	A	B	C	D	E
1					
2		1	2		
3		2	4		
4		3	7		
5		4	8		
6		5	11		
7					
8			13	=FORECAST(6; B2:B6; A2:A6)	
9					
10			13	=TREND(B2:B6; A2:A6; 6)	
11					

FTEST

Функция **FTEST** возвращает результат вычисления F-критерия (вероятность того, что два набора данных получены из одной и той же генеральной совокупности).

Синтаксис функции:

=FTEST(data1; data2)

где:

- **data1** и **data2** — диапазоны или массивы (возможно различного размера) содержащие числа, для которых вычисляется F-критерий.

F-критерий определяет вероятность того, что два образца имеют одинаковую дисперсию.

На самом деле, вычисляются выборочные дисперсии **data1** и **data2**. Диапазоны **data1** и **data2** реорганизуются в случае необходимости таким образом, чтобы **data1** имел большую дисперсию (σ_1), а **data2** — меньшую (σ_2), и вычисляется **F_value** как σ_1/σ_2 . Результат, возвращаемый **FTEST** — **2*FDIST(F_value; COUNT(data1)-1; COUNT(data2)-1)**. Это двусторонняя вероятность того, что дисперсии в **data1** и **data2** отличаются незначительно. (Внутренние алгоритмы, используемые Calc, более сложны чем приведенный).

Параметры **data1** и **data2** всегда оцениваются как формулы массива.

	A	B	C	D	E
1					
2		0,8194	=FTEST({9;8;6;8}; {5;6;7})		
3					

В ячейке **B2** возвращается приблизительно **0,8194**. Вероятность того, что дисперсии этих двух выборок отличаются незначительно - приблизительно 81,94 %. Заметьте: реальный пример нуждается в большем количестве данных чем приведены здесь.

Ранние версии Excel и спецификации проекта ODFD неправильно утверждали, что это был односторонний критерий.

GAMMADIST

Функция **GAMMADIST** вычисляет значения функции плотности вероятности или интегральной функции распределения для Гамма-распределения.

Синтаксис функции:

=GAMMADIST(x; α ; β ; mode)

где:

- **x** — значение, для которого требуется вычислить распределение;
- **α** — параметр распределения (форма);
- **β** — параметр распределения (масштаб);
- **mode** — логическое значение, определяющее форму функции.

Гамма-распределение — двухпараметрическое семейство непрерывных распределений вероятности, с параметрами **α** (форма) и **β** (масштаб). Эту функцию можно использовать для изучения переменных, которые имеют асимметричное распределение. Гамма-распределение обычно используется в теории очередей.

Если **mode = 0**, **GAMMADIST** вычисляет функцию плотности вероятности Гамма-распределения:

$$\frac{\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\frac{x}{\beta}}}{\beta \Gamma(\alpha)}$$

Если **mode = 1**, **GAMMADIST** вычисляет интегральную функцию распределения Гамма-распределения:

$$\int_0^x \frac{\left(\frac{t}{\beta}\right)^{\alpha-1} e^{-\frac{t}{\beta}}}{\beta \Gamma(\alpha)} dt$$

Параметр **x** должен быть больше или равен 0.

	A	B	C	D	E
1					
2		0,864665	=GAMMADIST(2; 1; 1; 1)		
3					

GAMMAINV

Функция **GAMMAINV** вычисляет инверсию интегральной функции **GAMMADIST**.

Синтаксис функции:

=GAMMAINV(p; α; β)

где:

- **p** — вероятность, связанная с гамма-распределением;
- **α** — параметр распределения (форма);
- **β** — параметр распределения (масштаб).

Функция **GAMMAINV** возвращает значение **x**, такое что **GAMMADIST(x; α; β; 1) = p**.

	A	B	C	D	E
1					
2		1,966113	=GAMMAINV(0,86; 1; 1)		
3					

GAMMALN

Функция **GAMMALN** возвращает натуральный логарифм гамма-функции⁷.

Синтаксис функции:

=GAMMALN(x)

где:

- **x** — значение, для которого требуется вычислить гамма-функцию, должно быть больше 0.

Функция **GAMMALN** возвращает натуральный логарифм **Γ(x)**, гамма-функции.

	A	B	C	D	E
1					
2		1,791759	=GAMMALN(4)		
3					

Разрабатываемый международный стандартный ODFD категоризирует ее как математическую функцию, и включает функцию **GAMMA**, которая в настоящее время не реализована в Calc.

GAUSS

Функция **GAUSS** возвращает значение в стандартном нормальном интегральном распределении.

Синтаксис функции:

=GAUSS(x)

где:

⁷ *Гамма-функция* — математическая функция, которая расширяет понятие факториала на поле комплексных чисел. Была введена Леонардом Эйлером (1729), а своим обозначением обязана Лежандру.

- x — число.

Функция **GAUSS** возвращает **NORMSDIST(x) – 0,5**.

	A	B	C	D	E
1					
2			0	=GAUSS(0)	
3					

GEOMEAN

Функция **GEOMEAN** возвращает среднее геометрическое аргументов.

Синтаксис функции:

=GEOMEAN(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа. Числа не должны быть нулем.

Среднее геометрическое a_1, a_2, \dots, a_n определяется как:

$$\left(\prod_{i=1}^n a_i \right)^{\frac{1}{n}}$$

	A	B	C	D	E
1					
2			2	=GEOMEAN(1; 2; 4)	
3					

В ячейке **B2** возвращается **2**, вычисляемое как $(1 \cdot 2 \cdot 4)^{\frac{1}{3}}$.

Calc предполагает, что логические значения **1 (TRUE)** и **0 (FALSE)**. Excel игнорирует логические значения.

HARMEAN

Функция **HARMEAN** возвращает среднее гармоническое аргументов.

Синтаксис функции:

=HARMEAN(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа. Числа не должны быть нулем.

Среднее гармоническое a_1, a_2, \dots, a_n определяется как:

$$\frac{n}{\sum_{i=1}^n \frac{1}{a_i}}$$

	A	B	C	D	E
1					
2			3	=HARMEAN(2; 4; 4)	
3					

В ячейке **B2** возвращается **3**, вычисляемое как $\frac{3}{\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{4}\right)}$.

Calc предполагает, что логические значения **1 (TRUE)** и **0 (FALSE)**. Excel игнорирует логические значения.

HYPGEOMDIST

Функция **HYPGEOMDIST** вычисляет значения для гипергеометрического распределения.

Синтаксис функции:

=HYPGEOMDIST(x; n; M; N)

где:

- **x** — количество успешных испытаний в выборке;
- **n** — размер выборки;
- **M** — количество успешных испытаний в генеральной совокупности;
- **N** — размер генеральной совокупности.

Гипергеометрическое распределение — дискретное распределение вероятности, дающее **x** вероятность успешных испытаний в последовательности жеребьевок **n** (без повторений) из конечной генеральной совокупности размера **N**, которая содержит **M** успешных испытаний.

Функция **HYPGEOMDIST** вычисляет функцию плотности вероятности гипергеометрического распределения:

$$\frac{\text{кол-во путей для } x \text{ успехов} \times \text{кол-во путей для } (n - x) \text{ неудач}}{\text{общее кол-во путей для выбора}}$$

Уравнение для гипергеометрического распределения имеет следующий вид:

$$\frac{\binom{M}{x} \binom{N-M}{n-x}}{\binom{N}{n}}$$

	A	B	C	D	E
1					
2		0,45	=HYPGEOMDIST(2; 3; 3; 6)		
3					

В ячейке **B2** возвращается **0,45**. Если урна содержит 3 красных шара и 3 зеленых шара, вероятность того, что 2 красных шара будут выбраны после 3 попыток без повторений — $27/60 = 0.45$.

Разрабатываемый международный стандарт ODFD добавляет дополнительный параметр этой функции, предоставляя возможность вычислять интегральную функцию распределения.

INTERCEPT

Функция **INTERCEPT** подбирает прямую линию к данным, используя линейную регрессию и возвращает ее точку пересечения с осью **Y**.

Синтаксис функции:

=INTERCEPT(yvalues; xvalues)

где:

- **yvalues** и **xvalues** — однострочковые или одностолбцовые диапазоны, определяющие точки в наборе данных.

Диапазоны **yvalues** и **xvalues** должны быть одинакового размера.

Функция **INTERCEPT** подбирает прямую линию через эти точки данных, используя метод линейной регрессии (наименьших квадратов⁸). Она возвращает значение y , где эта прямая линия пересекает ось Y .

Уравнение прямой линии можно задать как $y = a + bx$. Метод линейной регрессии вычисляет:

$$b = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

и

$$a = \bar{y} - b\bar{x}$$

точка пересечения с осью Y , возвращаемая этой функцией.

Параметры **xvalues** и **yvalues** всегда оцениваются как формулы массива.

	A	B	C	D	E
1					
2		1	2		
3		2	4		
4		3	7		
5		4	8		
6		5	10		
7					
8					
9					

0,2 = INTERCEPT(B2:B6; A2:A6)

В ячейке **B8** возвращается **0,2**. Найденное уравнение прямой линии — почти $y = 2x$ (и было бы им, если бы ячейка **B4** содержала 6) — таким образом, прямая линия проходит почти (но не совсем) через начало координат.

KURT

Функция **KURT** возвращает меру того, как достигло максимума или минимума распределение.

Синтаксис функции:

=KURT(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов/массивов, содержащих числа.

Функция **KURT** возвращает эксцесс, меру того, как достигло максимума или минимума распределение, относительно нормального распределения. Положительные значения указывают относительно остроконечное распределение, а отрицательные — относительно сглаженное распределение. Функция **KURT** вычисляет:

$$\frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}$$

⁸ *Метод наименьших квадратов* — один из методов регрессионного анализа для оценки неизвестных величин по результатам измерений, содержащим случайные ошибки. Метод наименьших квадратов применяется также для приближённого представления заданной функции другими (более простыми) функциями и часто оказывается полезным при обработке наблюдений.

для $n \geq 4$ чисел, имеющих среднееквадратичное отклонение $s > 0$.

Если имеется менее четырех точек данных, или если среднееквадратичное отклонение выборки равняется нулю, функция **KURT** возвращает значение ошибки #VALUE.

=KURT(A1:A30)

возвращает эксцесс чисел в диапазоне **A1:A30**.

	A	B	C	D	E
1					
2		0,2	=KURT(1; 3; 4; 5; 7)		
3					

В ячейке **B2** возвращается **0,2**, указывающее, что это (слишком небольшое, чтобы быть полезным) распределение медленно достигает максимума.

LARGE

Функция **LARGE** возвращает n -ое наибольшее значение в списке чисел.

Синтаксис функции:

=LARGE(numberlist; n)

где:

- **numberlist** — диапазон или массив чисел (возможно неупорядоченный);
- **n** — целое число, позиция (начиная с наибольшей) в массиве или диапазоне ячеек данных.

Функция **LARGE** возвращает n -ое наибольшее число в пределах (неупорядоченного) диапазона или массива чисел **numberlist**.

	A	B	C	D	E
1		4			
2		7			
3		5			
4		8			
5					
6		5	=LARGE(B1:B4; 3)		
7					

В ячейке **B6** возвращается **5**. Самое большое число — 8, потом 7, затем на третьем месте 5.

LOGINV

Функция **LOGINV** вычисляет инверсию функции **LOGNORMDIST**.

Синтаксис функции:

=LOGINV(p; μ ; σ)

где:

- **p** — вероятность, связанная с нормальным логарифмическим распределением;
- **μ** — среднее;
- **σ** — среднееквадратичное отклонение.

Переменная имеет логарифмически нормальное (логнормальное) распределение, если ее натуральный логарифм имеет нормальное распределение. Параметры распределения - **μ** (среднее) и **σ** (среднееквадратичное отклонение).

Функция **LOGINV** возвращает значение x , такое что **LOGNORMDIST**(x ; μ ; σ) = p .

	A	B	C	D	E
1					
2			1	=LOGINV(0,5; 0; 1)	
3					
4			0,5	=LOGNORMDIST(1; 0; 1)	
5					

LOGNORMDIST

Функция **LOGNORMDIST** вычисляет значения для интегральной функции логнормального распределения.

Синтаксис функции:

=LOGNORMDIST(x ; μ ; σ)

где:

- p — вероятность, связанная с нормальным логарифмическим распределением;
- μ — среднее;
- σ — среднеквадратичное отклонение.

Переменная имеет логарифмически нормальное (логнормальное) распределение, если ее натуральный логарифм имеет нормальное распределение. Параметры распределения - μ (среднее) и σ (среднеквадратичное отклонение).

Логнормальное распределение обычно используется для общего анализа надежности, числа циклов до отказа при усталости, сопротивлении материалов и переменных нагрузках в вероятностном проектировании. Другое преимущество логнормального распределения состоит в том, что оно является положительно-определенным, таким образом оно часто полезно для представления параметров, которые не могут иметь отрицательных значений. Логнормальные распределения оказались полезными как распределения для количества осадков, для распределения размера частиц аэрозоля или капелек, и для многих других случаев.

Функция **LOGNORMDIST** вычисляет интегральную функцию плотности для логнормального распределения.

LOGNORMDIST(x ; μ ; σ) эквивалентно **NORMDIST**((LN(x) - μ)/ σ ; 0; 1; 1); она также может быть вычислена как $0,5 + 0,5 * \text{ERF}((\text{LN}(x) - \mu)/(\sigma * \text{SQRT}(2)))$

Функция **LOGNORMDIST** при $x > 0$ возвращает значение

$$\int_0^x \frac{e^{-\frac{1}{2} \left(\frac{\ln(t) - \mu}{\sigma} \right)^2}}{t \sqrt{2\pi} \sigma} dt$$

и 0 в противном случае.

	A	B	C	D	E
1					
2			0,5	=LOGNORMDIST(1; 0; 1)	
3					

В разрабатываемом международном стандартном ODFFF эта функция имеет дополнительный параметр, позволяющий также вычислять функцию плотности вероятности.

MAX

Функция **MAX** возвращает наибольшее значение из списка аргументов, игнорируя текстовые записи.

Синтаксис функции:

=MAX(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

	A	B	C	D	E
1					
2			6	=MAX(2; 6; 4)	
3					
4			1,1		
5			2,2		
6			Яблоко		
7					
8			2,2	=MAX(B4:B6)	
9					

Calc предполагает, что логические значения **1 (TRUE)** и **0 (FALSE)**. Excel игнорирует логические значения.

MAXA

Функция **MAXA** возвращает наибольшее значение из списка аргументов, включая текстовые и логические записи.

Синтаксис функции:

=MAXA(value1; value2; ... value30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, которые могут включать числа, текст и логические значения. Текст оценивается как 0. Логические значения оцениваются как **1 (TRUE)** и **0 (FALSE)**.

	A	B	C	D	E
1					
2			6	=MAXA(2; 6; 4)	
3					
4			-2		
5			-1		
6			Яблоко		
7					
8			0	=MAXA(B4:B6)	
9					
10			1	=MAXA(0; TRUE())	
11					

MEDIAN

Функция **MEDIAN** возвращает медиану множества чисел.

Синтаксис функции:

=MEDIAN(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

Функция **MEDIAN** возвращает медиану (среднее значение) чисел. Если количество чисел нечетно, это — точное среднее значение. Если количество чисел четно, возвращается среднее двух средних значений.

Медиана выборки (термин был впервые введен Гальтоном, 1882) — это значение, которое разбивает выборку на две равные части (при ранжировании). Половина значений лежит ниже медианы, а половина — выше медианы.

	A	B	C	D	E
1					
2			9	=MEDIAN(1; 5; 9; 20; 21)	
3					
4			7	=MEDIAN(1; 5; 9; 20)	
5					

В ячейке **B2** возвращается **9**, число, лежащее точно в середине набора чисел.

В ячейке **B4** возвращается 7, которое является средним **5** и **9**, эти два числа лежат в середине набора чисел.

MIN

Функция **MIN** возвращает наименьшее значение из списка аргументов, игнорируя текстовые записи.

Синтаксис функции:

=MIN(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

	A	B	C	D	E
1					
2			2	=MIN(2; 6; 4)	
3					
4			1,1		
5			2,2		
6			Яблоко		
7					
8			1,1	=MIN(B4:B6)	
9					

Calc предполагает, что логические значения **1 (TRUE)** и **0 (FALSE)**. Excel игнорирует логические значения.

MINA

Функция **MINA** возвращает наименьшее значение из списка аргументов, включая текстовые и логические записи.

Синтаксис функции:

=MINA(value1; value2; ... value30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, которые могут

включать числа, текст и логические значения. Текст оценивается как 0. Логические значения оцениваются как **1 (TRUE)** и **0 (FALSE)**.

	A	B	C	D	E
1					
2			2	=MINA(2; 6; 4)	
3					
4			3		
5			4		
6			Яблоко		
7					
8			0	=MINA(B4:B6)	
9					
10			1	=MINA(2; TRUE())	
11					

MODE

Функция **MODE** возвращает наиболее часто встречающееся значение множества чисел.

Синтаксис функции:

=MODE(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов/массивов, содержащих числа.

Функция **MODE** возвращает число, которое появляется чаще всего, или если более чем одно число появляется так много раз, наименьшее из этих чисел. Если нет числа, которое появляется более одного раза, функция **MODE** возвращает ошибку.

Функция **MODE** оценивает свои параметры, как формулы массива — смотри примеры.

	A	B	C	D	E	F
1	1					
2	3		3	=MODE(A1:A4)		
3	2					
4	3		3	=MODE(3; 7; 3; 9; 9; 3; 5; 1; 9; 5)		
5						
6			2	=MODE(ABS({1;1;1;2;2;-2;-2}))		
7						

В ячейке **B2** возвращается **3**, число, встречающееся чаще всего в диапазоне **A1:A4**.

В ячейке **B4** возвращается **3**. Число **9** встречается такое-же количество раз, но **3** является меньшим.

В ячейке **B6** возвращается **2**. Функция **ABS** обычно принимает одно число, но здесь ему передается массив. **MODE** заставляет формулу быть формулой массива, так, что находится абсолютное значение каждого числа в массиве, и передается **MODE** как массив {1; 1; 1; 2; 2; 2; 2}. Самая общее значение теперь **2**.

Для данных в диапазоне **A1:A10**, формула **=MODE(-A1:A10)=MODE(A1:A10)** возвращает **TRUE**, если есть только одно самое общее значение, и **FALSE** в противном случае.

NEGBINOMDIST

Функция **NEGBINOMDIST** вычисляет вероятности для отрицательного биномиального распределения.

Синтаксис функции:

=NEGBINOMDIST(x; r; p)

где:

- **x** — количество неудачных испытаний;
- **r** — пороговое значение количества успешных испытаний;
- **p** — вероятность “успеха”.

Для независимых испытаний каждый с вероятностью успеха **p**, функция **NEGBINOMDIST** возвращает вероятность того, что будет точно **x** неудач прежде чем будет достигнуто **r** успехов. Используемая формула:

$$\binom{x+r-1}{r-1} p^r (1-p)^x = \frac{(x+r-1)!}{x!(r-1)!} p^r (1-p)^x$$

	A	B	C	D	E
1					
2		0,25	=NEGBINOMDIST(1; 1; 0,5)		
3					

В ячейке **B2** возвращается **0,25** (25 %), вероятность того, что орел выпадет точно один раз перед решкой при бросках монеты.

NORMDIST

Функция **NORMDIST** вычисляет значения функции плотности вероятности или совокупной функции распределения для нормального распределения с заданными средним и среднеквадратичным отклонением.

Синтаксис функции:

=NORMDIST(x; μ; σ; mode)

где:

- **x** — значение, для которого вычисляется распределение;
- **μ** — среднее арифметическое распределения (математическое ожидание);
- **σ** — среднеквадратичное отклонение распределения (разброс);
- **mode** — логическое значение, определяющее форму функции.

Нормальное распределение, также называемое *распределением Гаусса* — семейство непрерывных распределений вероятности с двумя параметрами управления **α** и **λ**. Физическая величина подчиняется нормальному распределению, когда она подвержена влиянию огромного числа случайных помех.

Если **mode** = 0, функция **NORMDIST** вычисляет плотность вероятности нормального распределения:

$$\frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\sqrt{2\pi}\sigma}$$

Если **mode** = 1, функция **NORMDIST** вычисляет интегральную функцию нормального распределения:

$$\int_{-\infty}^x \frac{e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}}{\sqrt{2\pi}\sigma} dt$$

	A	B	C	D	E
1					
2		0,03	=NORMDIST(70; 63; 5; 0)		
3					
4		0,92	=NORMDIST(70; 63; 5; 1)		
5					

NORMINV

Функция **NORMINV** вычисляет инверсию интегральной функции нормального распределения **NORMDIST**.

Синтаксис функции:

=NORMINV(p; μ ; σ)

где:

- **p** — вероятность, связанная с нормальным распределением;
- **μ** — среднее арифметическое распределения (математическое ожидание);
- **σ** — среднеквадратичное отклонение распределения (разброс).

Нормальное распределение, также называемое *распределением Гаусса* — семейство непрерывных распределений вероятности с двумя параметрами управления **α** и **λ** . Физическая величина подчиняется нормальному распределению, когда она подвержена влиянию огромного числа случайных помех.

Функция **NORMINV** возвращает значение **x**, такое что **NORMDIST(x; μ ; σ ; 1) = p**.

	A	B	C	D	E
1					
2		70,03	=NORMINV(0,92; 63; 5)		
3					

NORMSDIST

Функция **NORMSDIST** вычисляет значения для интегральной функции стандартного нормального распределения.

Синтаксис функции:

=NORMSDIST(x)

где:

- **x** — значение, для которого вычисляется вероятность.

Стандартное нормальное распределение — нормальное распределение со средним **$\mu = 0$** и среднеквадратичным отклонением **$\sigma = 1$** .

Функция **NORMSDIST** вычисляет интегральную функцию стандартного нормального распределения. Она эквивалентна **NORMDIST(x; 0; 1; 1)**.

	A	B	C	D	E
1					
2		0,5	=NORMSDIST(0)		
3					
4		0,5	=NORMDIST(0; 0; 1; 1)		
5					

В ячейке **B2** возвращается **0,5**. Кривая стандартного нормального распределения сосредоточена у 0, следовательно половина кривой - ниже 0. В ячейке **B4** также

возвращается **0,5**.

Разрабатываемый международный стандартный ODFF расценивает ее как унаследованную функцию, предпочитая **NORMDIST**.

NORMSINV

Функция **NORMSINV** вычисляет инверсию функции **NORMSDIST**.

Синтаксис функции:

=NORMSINV(p)

где:

- **p** — вероятность, связанная со стандартным нормальным распределением.

Функция **NORMSINV** возвращает значение **x**, такое что **NORMSDIST(x) = p**.

Она эквивалентна **NORMINV(p; 0; 1)**.

	A	B	C	D	E
1					
2			0	=NORMSINV(0,5)	
3					
4			0	=NORMINV(0,5; 0; 1)	
5					

В ячейке **B2** возвращается **0**. Половина стандартного нормального распределения располагается ниже 0. В ячейке **B4** также возвращается **0**.

Разрабатываемый международный стандартный ODFF расценивает ее как унаследованную функцию, предпочитая **NORMINV**.

PEARSON

Функция **PEARSON** возвращает коэффициент корреляции Пирсона двух наборов данных.

Синтаксис функции:

=PEARSON(x; y)

где:

- **x** и **y** — диапазоны или массивы, содержащие два набора данных.

Любой текст или пустые записи игнорируются.

Функция **PEARSON** вычисляет:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

где \bar{x} и \bar{y} — среднее **x** и **y**.

Параметры **x** и **y** всегда оцениваются как формулы массива.

=PEARSON(A1:A30; B1:B30)

возвращает коэффициент корреляции Пирсона для двух наборов данных в диапазонах **A1:A30** и **B1:B30**.

PERCENTILE

Функция **PERCENTILE** возвращает указанную процентилю⁹ в списке чисел.

Синтаксис функции:

=PERCENTILE(numberlist; fraction)

где:

- **numberlist** — диапазон или массив чисел, не обязательно упорядоченный;
- **fraction** — значение процентиля в интервале от 0 до 1 включительно.

Функция **PERCENTILE** возвращает значение ниже которого лежат **fraction** чисел из **numberlist**.

Функция **PERCENTILE** внутренне назначает разряд каждому числу в **numberlist**, где 0 - разряд самого наименьшего числа, 1 — следующее из самых наименьших чисел и так далее. Разряд r значения, которое будет найдено — $fraction * (N - 1)$, где N — количество чисел в **numberlist**. Если r — целое число, возвращена соответствующее значение из **numberlist**; в противном случае значение вычисляется пропорционально между значениями с разрядом $|r|$ и $|r| + 1$ (см. примеры).

Эта функция может быть полезной, например, для определения оценки, ниже которой лежит определенный процент из всех оценок теста. Она имеет ограниченное использование с небольшим списком чисел.

Функция **PERCENTILE** возвращает максимум, медиану, минимальное значение когда **fraction** — 1, 0,5 и 0 соответственно.

PERCENTILE(A2:A99; 0.6)

Ячейки, **A2:A99** содержат массив оценок теста, формула возвращает оценку, ниже которой лежат 60 % массива.

Следующие два примера иллюстрируют вычисление, но не предназначены для отображения практического применения:

	A	B	C	D	E
1					
2		9	=PERCENTILE({3;4;9;11;12}; 0,5)		
3					
4		6	=PERCENTILE({9;3;7;8}; 0,25)		
5					

В ячейке **B2** возвращается **9**, медиана списка.

В ячейке **B4** возвращается **6**. В списке есть 4 числа. Разряд значения, которое будет найдено — $0,25 * (4 - 1) = 0,75$. Самому малому числу **3** назначен разряд 0; следующему большему числу **7** назначен разряд 1. Значение с разрядом **0,75** лежит между 3 и 7, и вычисляется как $3 + (7 - 3) * 0,75 = 6$.

Несмотря на свое название, функция **PERCENTILE** требует дробную часть, а не процент.

PERCENTRANK

Функция **PERCENTRANK** возвращает процентный рейтинг числа в списке чисел.

⁹ *Процентиль* — мера расположения данных выборки или распределения. Говорят, что n -й процентиль - это такое значение, ниже которого расположено n процентов наблюдений данной переменной. Следовательно, 40-й процентиль — это значение, ниже которого расположено 40% результатов наблюдений; 50-й процентиль называется медианой, а 25-й и 75-й процентиля — нижним и верхним квартилями соответственно.

Синтаксис функции:

=PERCENTRANK(numberlist; number)

где:

- **numberlist** — диапазон или массив чисел, не обязательно упорядоченный;
- **number** — число, процентный рейтинг которого должен быть найден.

Функция **PERCENTRANK** внутренне назначает рейтинг r числу, где 0 — рейтинг самого наименьшего числа, 1 — следующего из самых наименьших и так далее. Если число отсутствует в **numberlist**, ему присваивается дробный рейтинг пропорционально рейтингам чисел с обеих сторон (см. примеры). Если N — количество чисел в **numberlist**, функция **PERCENTRANK** возвращает $r/(n-1)$, дробное число между 0 и 1 включительно.

Эта функция имеет ограниченное использование с небольшим списком чисел. Она может быть полезна например, для определения положения балла в пределах всех баллов для теста, и в этом случае формула возвращает, как заявляют, $L/(n-1)$, где L — количество баллов, которое меньше чем **number**.

PERCENTRANK(A2:A99; A7)

Когда ячейки **A2:A99** содержат оценки теста, возвращает процентный рейтинг (в виде дроби) для студента, экзаменационная отметка которого находится в ячейке **A7**.

Следующие два примера иллюстрируют вычисление, но имеют малое практическое значение:

	A	B	C	D	E
1					
2		0,25	=PERCENTRANK({3;4;9;11;12}; 4)		
3					
4		0,25	=PERCENTRANK({9;3;7;8}; 6)		
5					

В ячейке **B2** возвращается **0,25**.

В ячейке **B2** возвращается **0,25**. Самому наименьшему числу 3 назначается рейтинг “0”; следующему по порядку числу 7 назначается рейтинг “1”. Числу 6 назначается рейтинг пропорционально между ними — “0,75”. В списке есть 4 числа, таким образом результат вычисляется как $0,75 / (4-1) = 0,75/3 = 0,25$.

Несмотря на свое название, функция **PERCENTRANK** возвращает дробную часть, а не процент.

PERMUT

Функция **PERMUT** возвращает число упорядоченных перестановок для заданного числа объектов.

Синтаксис функции:

=PERMUT(n; k)

где **n** и **k** — целые числа.

Функция **PERMUT** возвращает число упорядоченных путей, которыми **k** объектов могут быть выбраны из набора **n** объектов, когда объект может быть выбран только однажды. Например, из набора из 3 объектов А, В, С, мы можем выбрать 2 следующим образом: АВ, АС, ВА, ВС, СА, СВ.

Функция **PERMUT** вычисляет:

$${}_n P_k = \frac{n!}{(n-k)!}$$

	A	B	C	D	E
1					
2		6	=PERMUT(3; 2)		
3					

В ячейке **B2** возвращается 6, как в примере, описанном выше.

Разрабатываемый международный стандартный ODFD ожидает, что **n** и **k** — целые числа. И Calc и Excel отсекают любой не целочисленный параметр.

PERMUTATIONA

Функция **PERMUTATIONA** возвращает число упорядоченных перестановок для заданного числа объектов, разрешая повторение.

Синтаксис функции:

=PERMUTATIONA(n; k)

где **n** и **k** — целые числа.

Функция **PERMUTATIONA** возвращает число упорядоченных путей, которыми **k** объектов могут быть выбраны из набора **n** объектов, когда объект может быть выбран неоднократно. Например, из набора из 3 объектов A, B, C, мы можем выбрать 2 следующим образом: AA, AB, AC, BA, BB, BC, CA, CB, CC.

Функция **PERMUTATIONA** просто вычисляет n^k .

	A	B	C	D	E
1					
2		9	=PERMUTATIONA(3; 2)		
3					

В ячейке **B2** возвращается 9, как в примере, описанном выше.

PHI

Функция **PHI** вычисляет значения для стандартного нормального распределения.

Синтаксис функции:

=PHI(x)

где:

- **x** — значение, для которого вычисляется вероятность.

Стандартное нормальное распределение — нормальное распределение со средним $\mu = 0$ и среднеквадратичным отклонением $\sigma = 1$.

Функция **PHI** вычисляет функцию плотности вероятности стандартного нормального распределения; другими словами она возвращает **NORMDIST(x; 0; 1; 0)**.

	A	B	C	D	E
1					
2		0,35207	=PHI(0,5)		
3					
4		0,35207	=NORMDIST(0,5; 0; 1; 0)		
5					

POISSON

Функция **POISSON** вычисляет значения для распределения Пуассона.

Синтаксис функции:

=POISSON(x; λ; mode)

где:

- **x** — значение, для которого вычисляется вероятность;
- **λ** — ожидаемое численное значение (математическое ожидание);
- **mode** — логическое значение, определяющее форму возвращаемого распределения вероятностей.

Распределение Пуассона — дискретное распределение вероятности, дающее вероятность того, что **x** событий происходят за определенное время, где события происходят независимо, и где ожидаются в среднем **λ** события.

Если **mode** = 0, функция **POISSON** вычисляет функцию плотности вероятности распределения Пуассона:

$$\frac{e^{-\lambda} \lambda^x}{x!}$$

Если **mode** = 1, функция **POISSON** вычисляет интегральную функцию распределения Пуассона:

$$\sum_{k=0}^x \frac{e^{-\lambda} \lambda^k}{k!}$$

	A	B	C	D	E
1					
2			0,13176	=POISSON(8; 9; 0)	
3					
4			0,45565	=POISSON(8; 9; 1)	
5					

В ячейке **B2** возвращается приблизительно **0,13176**, вероятность того, что точно 8 событий происходят за период времени, за который Вы ожидаете 9 событий.

В ячейке **B4** возвращается приблизительно **0,45565**, вероятность того, что до 8 событий включительно происходит за период времени, за который Вы ожидаете 9 событий.

PROB

Функция **PROB** возвращает результат из списка вероятностей.

Синтаксис функции:

=PROB(values; probabilities; start; end)

где:

- **values** — диапазон или массив чисел (возможно неупорядоченный);
- **probabilities** — диапазон или массив чисел того же самого размера что и **values**, указывающий соответствующую вероятность (>0 и <=1) того, что произойдет каждое значение в **values**. Числа в **values** должны в сумме давать точно 1;
- **start** — нижняя граница значения, для которого вычисляется вероятность;
- **end** — верхняя граница значения, для которого требуется вычислить вероятность.

Функция **PROB** находит все значения, которые располагаются между **start** и **end** включительно, и возвращает сумму соответствующих вероятностей. Параметр **end** может быть опущен, в этом случае функция **PROB** возвращает вероятность, соответствующую **start** (или 0, если **start** не присутствует в **values**).

Функция **PROB** оценивает параметры **values** и **probabilities** как формулы массива — смотри примеры.

	A	B	C	D	E	F
1						
2		0,8	=PROB({3;4;5;6}; {0,2;0,4;0,3;0,1}; 4; 6)			
3						
4		0,7	=PROB({2,2;5;1}; {0,5;0,3;0,2}; 0; 3)			
5						
6		0,4	=PROB({3;4;5;6}; {0,2;0,4;0,3;0,1}; 4)			
7						

В ячейке **B2** возвращается **0,8**, сумма вероятностей для 4, 5 и 6.

В ячейке **B4** возвращается **0,7**, сумма вероятностей для 1 и 2,2.

В ячейке **B6** возвращается **0,4**, вероятность для 4.

	A	B	C	D	E
1	1	0,3			
2	4	0,7			
3					
4		0,3	=PROB(A1:A2*2; B1:B2; 2)		
5					

В ячейке **B4** возвращается **0,3**. Функция **PROB** вызывает оценку значений как формулу массива, таким образом, чтобы формула **A1:A2*2** приводила к массиву {2|8}. Таким образом, фактически вычисляется формула **PROB({2|8}; B1:B2; 2)**, которая возвращает результат.

	A	B	C	D	E
1	1	30			
2	4	70			
3					
4		0,7	=PROB(A1:A2; B1:B2/100;4)		
5					

В ячейке **B4** возвращается **0,7**. Функция **PROB** вызывает оценку вероятностей как формулу массива, таким образом, чтобы формула **B1:B2/100** приводила к массиву {0,3|0,7}. Ячейки **B1** и **B2** здесь содержат проценты.

QUARTILE

Функция **QUARTILE** возвращает указанный квартиль¹⁰ в списке чисел.

Синтаксис функции:

=QUARTILE(numberlist; q)

где:

- **numberlist** — диапазон или массив чисел, не обязательно упорядоченный;
- **q** — число, может быть 0 (чтобы вернуть минимальное значение), 1 (чтобы вернуть значение ниже который располагается четверть значений), 2 (чтобы

¹⁰ *Квартиль* — Указывает на место расположения данных выборки или распределения. Когда значение находится в зоне, где расположено нижние 25% наблюдаемых значений переменной, то говорят, что оно расположено в нижнем квартиле. Если же оно расположено там, где находятся верхние 25% значений, то говорят, что оно расположено в верхнем квартиле.

возвратить медиану выборки), 3 (чтобы вернуть значение, ниже которого располагается три четверти значений) или 4 (чтобы вернуть максимальное значение).

Функция **QUARTILE** возвращает значение, ниже которого лежит **q** четвертей чисел в **numberlist**. Функция **QUARTILE** вычисляет возвращаемое значение таким же образом как функция **PERCENTILE**.

QUARTILE(A2:A99; 3)

Когда ячейки **A2:A99** содержат оценки теста, возвращает оценку, ниже которой лежит 75 % оценок.

Следующие два примера иллюстрируют вычисление, но имеют малое практическое значение:

	A	B	C	D	E
1					
2		9	=QUARTILE({3;4;9;11;12}; 2)		
3					
4		6	=QUARTILE({9;3;7;8}; 1)		
5					

В ячейке **B2** возвращается **9**, медиана списка.

В ячейке **B4** возвращается **6**. Вычисление выполняется так же, как для **PERCENTILE({9; 3; 7; 8}; 0,25)**.

RANK

Функция **RANK** возвращает ранг числа в списке чисел.

Синтаксис функции:

=RANK(number; numberlist; order)

где:

- **number** — число, для которого определяется ранг;
- **numberlist** — диапазон или массив чисел, не обязательно упорядоченный;
- **order** — число, определяющее характер упорядочивания;

Функция **RANK** возвращает ранг числа **number** в пределах неупорядоченного диапазона или массива чисел **numberlist**. Если какое либо число в списке **numberlist** встречается несколько раз, ему присваивают один тот же ранг, но при этом увеличивается ранг для последующих отличающихся чисел.

Если **order** — ноль или опущен, числа в списке **numberlist** сортируются в порядке убывания, т. е. наибольшие числа располагаются сначала; если **order** — отличен от нуля, числа сортируются в порядке убывания, т. е. в начале располагаются наименьшие числа.

Возвращается ошибка, если число не присутствует в списке **numberlist**.

	A	B	C	D	E
1		4			
2		7			
3		5			
4		8			
5					
6		3	=RANK(5; B1:B4)		
7					
8		4	=RANK(5; {3;5;2;6;1}; 1)		
9					

В ячейке **B6** возвращается **3**. Самое максимальное число — 8, за ним следует 7, затем на третьем месте 5.

В ячейке **B6** возвращается **4**, поскольку другие три числа меньше чем 5.

В соответствии с проектом, функция **RANK** возвращает тот же самый ранг для любого повторяющегося значения в **numberlist**. Если Вы хотите, чтобы для каждого числа в диапазоне (скажем) **A1:A4** был уникальный ранг, введите в ячейке **B1** формулу **=RANK(A1; \$A\$1:\$A\$4) + COUNTIF(\$A\$1:A1; A1) – 1** и скопируйте/вставьте ее в ячейки **B2:B4**.

Calc предполагает, что любые логические значения соответствуют **1 (TRUE)** и **0 (FALSE)**. Поведение Excel с логическими значениями может быть “неожиданным”.

RSQ

Функция **RSQ** возвращает квадрат коэффициента корреляции Пирсона двух наборов данных.

Синтаксис функции:

=RSQ(x; y)

где:

- **x** и **y** — диапазоны или массивы, содержащие два набора данных.

Любой текст или пустые записи игнорируются.

Функция **RSQ** вычисляет квадрат коэффициента корреляции Пирсона (которому традиционно обозначают буквой *r*, следовательно “*r* в квадрате”). См. функцию **PEARSON** для получения дополнительной информации.

Функция **RSQ** показывает, как соотносятся дисперсия для **y** к дисперсии для **x**.

Параметры **x** и **y** всегда оцениваются как формулы массива.

=RSQ(A1:A30; B1:B30)

возвращает квадрат коэффициента корреляции Пирсона для двух наборов данных в диапазонах **A1:A30** и **B1:B30**.

SKEW

Функция **SKEW** оценивает перекося распределения, используя выборку множества чисел.

Синтаксис функции:

=SKEW(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов/массивов, содержащих числа.

Распределение должно содержать по крайней мере три числа.

Функция **SKEW** возвращает меру того, насколько перекошено распределение относительно нормального распределения — то есть, насколько оно асимметрично. Асимметрия характеризует степень несимметричности распределения относительно его середины. Положительное значение указывает на отклонение распределения в сторону положительных значений, а отрицательное — на отклонение распределения в сторону отрицательных значений. Функция **SKEW** вычисляется следующим образом:

$$\frac{n}{(n-1)(n-2)} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s} \right)^3$$

для $n \geq 3$ чисел, имеющих среднеквадратичное отклонение $s > 0$.

Если имеется менее трех чисел, функция **SKEW** возвращает значение ошибки **#VALUE**.

=SKEW(A1:A30)

возвращает оценку перекоса распределения чисел в диапазоне **A1:A30**.

	A	B	C	D	E
1					
2		0,885	=SKEW(1; 3; 4; 5; 9)		
3					
4		0,000	=SKEW(1; 3; 4; 5; 7)		
5					

В ячейке **B2** возвращается приблизительно **0,885**, указывая, что хвост этого (слишком маленького, чтобы быть полезным) распределения склоняется в положительную сторону.

В ячейке **B4** возвращается **0**; распределение является симметричным.

SLOPE

Функция **SLOPE** подбирает прямую линию к данным, используя линейную регрессию и возвращает его наклон.

Синтаксис функции:

=SLOPE(yvalues; xvalues)

где:

- **yvalues** и **xvalues** — одностроковые или одностолбцовые диапазоны, определяющие точки в наборе данных.

Диапазоны **yvalues** и **xvalues** должны быть одинакового размера.

Функция **SLOPE** подбирает прямую линию через эти точки данных, используя метод линейной регрессии (наименьших квадратов¹¹). После чего она возвращает наклон этой линии.

Уравнение прямой линии можно задать как $y = a + bx$. Метод линейной регрессии вычисляет:

$$b = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

и

¹¹ *Метод наименьших квадратов* — один из методов регрессионного анализа для оценки неизвестных величин по результатам измерений, содержащим случайные ошибки. Метод наименьших квадратов применяется также для приближённого представления заданной функции другими (более простыми) функциями и часто оказывается полезным при обработке наблюдений.

$$a = \bar{y} - b\bar{x}$$

b - наклон, возвращаемый этой функцией.

Параметры **xvalues** и **yvalues** всегда оцениваются как формулы массива.

	A	B	C	D	E
1					
2		1	2		
3		2	4		
4		3	6		
5		4	8		
6		5	11		
7					
8			2,2	=SLOPE(B2:B6; A2:A6)	
9					

В ячейке **B8** возвращается **2,2**. Уравнение найденной прямой линии — почти соответствует $y = 2x$ — таким образом, наклон — почти (но не совсем) 2.

SMALL

Функция **SMALL** возвращает n -ое наименьшее значение в списке чисел.

Синтаксис функции:

=SMALL(numberlist; n)

где:

- **numberlist** — диапазон или массив чисел (возможно неупорядоченный);
- **n** — целое число, позиция (начиная с наибольшей) в массиве или диапазоне ячеек данных.

Функция **SMALL** возвращает n -ое наименьшее число в пределах (неупорядоченного) диапазона или массива чисел **numberlist**.

	A	B	C	D	E
1		4			
2		7			
3		5			
4		8			
5					
6		7	=SMALL(B1:B4; 3)		
7					

В ячейке **B6** возвращается **7**. Самое наименьшее число — 4, потом 5, потом третье самое наименьшее — 7.

STANDARDIZE

Функция **STANDARDIZE** преобразует значение в нормальном распределении к его эквиваленту в стандартном нормальном распределении.

Синтаксис функции:

=STANDARDIZE(x; μ; σ)

где:

- **x** — число в нормальном распределении с известными средним μ и среднеквадратичным отклонением σ ;

- μ — среднее арифметическое распределения (математическое ожидание);
- σ — среднеквадратичное отклонение распределения (разброс).

Функция **STANDARDIZE** преобразует значение нормального распределения **n** к соответствующему значению в стандартном нормальном распределении, которое имеет среднее $\mu = 0$ и среднеквадратичное отклонение $\sigma = 1$.

Функция **STANDARDIZE** вычисляет и возвращает

$$\frac{(n - \mu)}{\sigma}$$

	A	B	C	D	E
1					
2		1	=STANDARDIZE(14; 10; 4)		
3					

В ячейке **B2** возвращается 1. Оригинальная значение 14 располагается на 1 среднеквадратичное отклонение выше первоначального среднего 10; новое значение располагается на 1 среднеквадратичное отклонение выше нового среднего 0.

STDEV

Функция **STDEV** возвращает среднеквадратичное отклонение по выборке. Среднеквадратичное отклонение часто используется для определения меры разброса значений относительно их среднего.

Синтаксис функции:

=STDEV(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

Функция **STDEV** возвращает среднеквадратичное отклонение, где **number1 ... number30** — выборка из генеральной совокупности. С **N** значениями в выборке, формула для вычисления выглядит следующим образом:

$$\sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Если данные представляют всю генеральную совокупность, то стандартное отклонение следует вычислять с помощью функции **STDEVP**.

	A	B	C	D	E
1					
2		2	=STDEV(2; 6; 4)		
3					

Calc предполагает, что логические значения **1 (TRUE)** и **0 (FALSE)**. Excel игнорирует логические значения.

STDEVA

Функция **STDEVA** возвращает среднеквадратичное отклонение по выборке.

Синтаксис функции:

=STDEVA(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа. Могут также быть включены логические значения и текст.

Функция **STDEVA** возвращает среднеквадратичное отклонение, где **number1 ... number30** — выборка из генеральной совокупности. Логические значения оцениваются как **1 (TRUE)** и **0 (FALSE)**. Текстовые значения всегда оцениваются как ноль.

С **N** значениями в выборке, формула для вычисления выглядит следующим образом:

$$\sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

	A	B	C	D	E	F
1						
2	красный		2	=STDEVA(2; 6; 4)		
3	ИСТИНА					
4	2		1	=STDEVA(A2:A4)		
5						

В ячейке **B4** возвращается **1**, среднеквадратичное отклонение **0, 1** и **2**.

Calc всегда оценивает текстовые значения как 0. Разрабатываемый международный стандарт ODFD позволяет числам в текстовой форме произвольно преобразовываться в числовую форму. Поэтому переносимость между электронными таблицами соответствующими ODFD может быть проблемой.

STDEVP

Функция **STDEVP** возвращает среднеквадратичное отклонение по генеральной совокупности¹².

Синтаксис функции:

=STDEVP(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

Функция **STDEVP** возвращает среднеквадратичное отклонение, где **number1 ... number30** - вся генеральная совокупность. Если Вы имеете только выборку из генеральной совокупности, используйте вместо нее функцию **STDEV**. Для генеральной совокупности из **N** значений, формула для вычисления выглядит следующим образом:

$$\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

	A	B	C	D	E
1					
2			2	=STDEVP(3; 3; 7; 7)	
3					

Calc предполагает, что логические значения **1 (TRUE)** и **0 (FALSE)**. Excel игнорирует логические значения.

¹² *Генеральная совокупность* (в англ. — *population*) — совокупность всех объектов (единиц), относительно которых учёный намерен делать выводы при изучении конкретной проблемы. Генеральная совокупность состоит из всех объектов, которые подлежат изучению. Состав генеральной совокупности зависит от целей исследования. Иногда генеральная совокупность - это все население определённого региона (например, когда изучается отношение потенциальных избирателей к кандидату), чаще всего задаётся несколько критериев, определяющих объект исследования. Например, женщины 18-29 лет, использующие крем для рук определённых марок не реже раза в неделю, и имеющие доход не ниже 5 000 руб. на одного члена семьи. *Выборка* — множество случаев (испытуемых, объектов, событий, образцов), с помощью определённой процедуры выбранных из генеральной совокупности для участия в исследовании.

STDEVPA

Функция **STDEVPA** возвращает среднее квадратичное отклонение по генеральной совокупности (допускающей текстовые и логические значения).

Синтаксис функции:

=STDEVPA(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа. Могут также быть включены логические значения и текст.

Функция **STDEVPA** возвращает среднее квадратичное отклонение, где **number1 ... number30** - вся генеральная совокупность. Логические значения оцениваются как **1 (TRUE)** и **0 (FALSE)**. Текстовые значения всегда оцениваются как ноль.

Если Вы имеете только выборку из генеральной совокупности, используйте вместо нее функцию **STDEVA**. Для генеральной совокупности из **N** значений, формула для вычисления выглядит следующим образом:

$$\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

	A	B	C	D	E	F
1	Красный					
2	ЛОЖЬ		2	=STDEVPA(3; 3; 7; 7)		
3		4				
4		4		2	=STDEVPA(A1:A4)	
5						

В ячейке **B4** возвращается **2**, среднее квадратичное отклонение по генеральной совокупности **0, 0, 4 и 4**.

STEYX

Функция **STEYX** подбирает прямую линию к данным, используя линейную регрессию и возвращает стандартную ошибку значений **y**.

Синтаксис функции:

=STEYX(yvalues; xvalues)

где:

- **yvalues** и **xvalues** — одностроковые или одностолбцовые диапазоны, определяющие точки в наборе данных.

Диапазоны **yvalues** и **xvalues** должны быть одинакового размера.

Функция **SLOPE** подбирает прямую линию через эти точки данных, используя метод линейной регрессии (наименьших квадратов). После чего она возвращает стандартную ошибку фактических значений **y** по сравнению с значениями **y** на найденной прямой линии.

Уравнение прямой линии можно задать как $y = a + bx$. Метод линейной регрессии вычисляет:

$$b = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

и

$$a = \bar{y} - b\bar{x}$$

Функция **STEYX** вычисляет:

$$\sqrt{\frac{\sum_i (y_i - y_i')^2}{n-2}}$$

где y_i — фактические значения, и y_i' - вычисленные значения на найденной линии.

Параметры **xvalues** и **yvalues** всегда оцениваются как формулы массива.

	A	B	C	D	E
1					
2		1		2	
3		2		4	
4		3		7	
5		4		8	
6		5		10	
7					
8			0,52	=STEYX(B2:B6; A2:A6)	
9					

В ячейке **B8** возвращается приблизительно **0,52**. Уравнение найденной прямой линии — почти соответствует $y = 2x$, и было бы им, если бы ячейка **B4** содержала 6. Стандартная ошибка поэтому весьма небольшая.

TDIST

Функция **TDIST** вычисляет значения для t-распределения Стьюдента.

Синтаксис функции:

=TDIST(x; r; mode)

где:

- **x** — значение, для которого вычисляется вероятность;
- **r** — целое число ≥ 1 , число степеней свободы для t-распределения;
- **mode** — может быть 1 или 2, соответствует числу возвращаемых хвостов распределения.

Распределение Стьюдента в теории вероятностей — это однопараметрическое семейство абсолютно непрерывных распределений. Распределение Стьюдента используется в статистике для точечного оценивания, построения доверительных интервалов и тестирования гипотез, касающихся неизвестного среднего статистической выборки из нормального распределения.

Функция **TDIST** возвращает функцию плотности вероятности для t-распределения, вычисляя:

$$\frac{\text{mode} \Gamma\left(\frac{r+1}{2}\right)}{\sqrt{\pi r} \Gamma\left(\frac{r}{2}\right)} \left(1 + \frac{x^2}{2}\right)^{-\frac{r+1}{2}}$$

	A	B	C	D	E
1					
2			0,6667	=TDIST(0,5; 2; 2)	
3					

TINV

Функция **TINV** вычисляет инверсию двусторонней функции **TDIST**.

Синтаксис функции:

=TINV(p; r)

где:

- **p** — вероятность, связанная с t-распределением Стьюдента, должна быть ≥ 0 и ≤ 1 ;
- **r** — целое число ≥ 1 , число степеней свободы для t-распределения.

Функция **TINV** возвращает значение **x**, такое что **TDIST(x; r; 2) = p**. Чтобы вычислить инверсию односторонней функции **TDIST**, используйте формулу **TINV(p*2; r)**, которая возвращает значение **x**, такое что **TDIST(x; r; 1) = p**.

	A	B	C	D	E
1					
2			0,5	=TINV(0,6667; 2)	
3					

TRIMMEAN

Функция **TRIMMEAN** возвращает среднее арифметическое множества чисел, игнорируя часть максимальных и минимальных значений.

Синтаксис функции:

=TRIMMEAN(numberlist; fraction)

где:

- **numberlist** — диапазон или массив чисел, не обязательно упорядоченных;
- **fraction** — доля точек данных, исключаемых из вычислений.

Функция **TRIMMEAN** возвращает среднее, игнорируя **fraction** ($0 \leq \text{fraction} < 1$) наиболее удаленных от середины значений в **numberlist**. Более точно, если **n** — количество чисел в **numberlist**, игнорируются **INT(n * fraction/2)** самых больших и самых малых значений.

=TRIMMEAN(A1:A100; 0,1)

возвращает среднее арифметическое значений в диапазоне ячеек **A1:A100**, игнорируя 5 самых больших и 5 самых малых значений.

TTEST

Функция **TTEST** возвращает вероятность, соответствующую t-критерию Стьюдента.

Синтаксис функции:

=TTEST(data1; data2; mode; type)

где:

- **data1** и **data2** — диапазоны или массивы (возможно различного размера) содержащие числа, для которых выполняется вычисление t-критерия;
- **mode** — режим: 1 — для одностороннего t- критерия, 2 — для двустороннего t- критерия;
- **type** — тип вычисляемого t- критерия: 1 — для парных выборок, 2 — для двух выборок с равной дисперсией, 3 — для двух образцов с неравной дисперсией.

Параметры **data1** и **data2** всегда оцениваются как формулы массива.

t-критерий Стьюдента — общее название для класса методов статистической проверки гипотез (статистических критериев), основанных на сравнении с распределением Стьюдента. Наиболее частые случаи применения t-критерия связаны с проверкой равенства средних значений в двух выборках.

Данный критерий был разработан Уильямом Госсетом для оценки качества пива в компании Гиннесс. В связи с обязательствами перед компанией по неразглашению коммерческой тайны (а руководство Гиннеса считало таковой использование статистического аппарата в своей работе), статья Госсетта вышла в журнале «Биометрика» под псевдонимом «Student» (Студент).

Для применения данного критерия необходимо, чтобы исходные данные имели нормальное распределение. В случае применения двухвыборочного критерия для независимых выборок также необходимо соблюдение условия равенства дисперсий. Существуют, однако, альтернативы критерию Стьюдента для ситуации с неравными дисперсиями.

TTEST(A2:A20; B2:20; 2; 3)

Формула возвращает результат вычисления двустороннего t-критерия Стьюдента для двух выборок с неравной дисперсией, используя данные в диапазонах **A2:A20** и **B2:20**.

VAR

Функция **VAR** возвращает дисперсию¹³ по выборке.

Синтаксис функции:

=VAR(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

Функция **VAR** возвращает выборочную дисперсию, где **number1 ... number30** - выборка из генеральной совокупности. Должны быть включены по крайней мере два числа. Возвращает ошибку, если передано менее чем два числа.

Для **N** значений в выборке, формула для вычисления выглядит следующим образом:

$$\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

	A	B	C	D	E
1					
2					
3					

4 =VAR(2; 6; 4)

VARA

Функция **VARA** возвращает дисперсию по выборке (допускаются текстовые и логические значения).

Синтаксис функции:

=VARA(number1; number2; ... number30)

где:

¹³ *Дисперсия случайной величины* — мера разброса данной случайной величины, т.е. её отклонения от математического ожидания.

Выборочная дисперсия в математической статистике — это оценка теоретической дисперсии распределения на основе выборки.

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа. Могут также быть включены логические значения и текст.

Функция **VARA** возвращает дисперсию распределения, где **number1 ... number30** — выборка из генеральной совокупности. Выборка должна содержать по крайней мере два числа. Возвращает ошибку, если передано менее чем два числа. Логические значения оцениваются как **1 (TRUE)** и **0 (FALSE)**. Текстовые значения всегда оцениваются как ноль.

Для **N** значений в выборке, формула для вычисления выглядит следующим образом:

$$\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

	A	B	C	D	E	F
1						
2	Красный		4	=VARA(2; 6; 4)		
3	ИСТИНА					
4	2		1	=VARA(A2:A4)		
5						

В ячейке **B4** возвращается **2**, дисперсия выборки **0, 1 и 2**.

Calc всегда оценивает текстовые значения как 0. Разрабатываемый международный стандарт ODFD позволяет числам в текстовой форме произвольно преобразовываться в числовую форму. Поэтому переносимость между электронными таблицами соответствующими ODFD может быть проблемой.

VARP

Функция **VARP** возвращает дисперсию генеральной совокупности.

Синтаксис функции:

=VARP(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа.

Функция **VARP** возвращает дисперсию распределения, где **number1 ... number30** - вся генеральная совокупность. Если Вы имеете только выборку из генеральной совокупности, используйте вместо нее функцию **VAR**. Для генеральной совокупности из **N** значений, формула для вычисления выглядит следующим образом:

$$\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$

	A	B	C	D	E
1					
2			4	=VARP(3; 3; 7; 7)	
3					

Calc предполагает, что логические значения **1 (TRUE)** и **0 (FALSE)**. Excel игнорирует логические значения.

VARPA

Функция **VARPA** возвращает дисперсию генеральной совокупности.

Синтаксис функции:

=VARPA(number1; number2; ... number30)

где:

- **number1; number2; ... number30** — до 30 чисел или диапазонов, содержащих числа. Могут также быть включены логические значения и текст.

Функция **VARP** возвращает дисперсию распределения, где **number1 ... number30** - вся генеральная совокупность. Логические значения оцениваются как **1 (TRUE)** и **0 (FALSE)**. Текстовые значения всегда оцениваются как ноль.

Если Вы имеете только выборку из генеральной совокупности, используйте вместо нее функцию **VARA**. Для генеральной совокупности из **N** значений, формула для вычисления выглядит следующим образом:

$$\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$

	A	B	C	D	E	F
1	Красный					
2	ЛОЖЬ		4	=VARPA(3; 3; 7; 7)		
3		4				
4		4		4	=VARPA(A1:A4)	
5						

В ячейке **B4** возвращается **2**, дисперсия генеральной совокупности **0, 0, 4 и 4**.

Calc всегда оценивает текстовые значения как 0. Разрабатываемый международный стандарт ODFD позволяет числам в текстовой форме произвольно преобразовываться в числовую форму. Поэтому переносимость между электронными таблицами соответствующими ODFD может быть проблемой.

WEIBULL

Функция **WEIBULL** вычисляет значения для распределения Вейбулла.

Синтаксис функции:

=WEIBULL(x; k; λ; mode)

где:

- **x** — значение, для которого вычисляется вероятность, $x \geq 0$;
- **k** — параметр распределения (коэффициент формы);
- **λ** — параметр распределения (коэффициент масштаба);
- **mode** — логическое значение, определяющее форму возвращаемого распределения вероятностей.

Распределение Вейбулла — непрерывное распределение вероятности, с параметрами **k > 0** (форма) и **λ > 0** (масштаб). Этому распределению хорошо подчиняется распределение отказов в объектах, содержащих большое количество однотипных неремонтируемых элементов (полупроводниковых приборов, микромодулей и т. д.). Названо в честь шведского инженера Валодди Вейбулла (Waloddi Weibull, 1887—1979).

Если **mode = 1**, функция **WEIBULL** вычисляет плотность вероятности распределения Вейбулла:

$$\frac{k}{\lambda} \left(\frac{x}{\lambda} \right)^{k-1} e^{-(x/\lambda)^k}$$

Если **mode = 0**, функция **WEIBULL** вычисляет совокупную функцию распределения для распределения Вейбулла:

$$1 - e^{-(x/\lambda)^k}$$

	A	B	C	D	E
1					
2		0,216623	=WEIBULL(2,5; 3; 4; 1)		
3					
4		0,229505	=WEIBULL(2,5; 3; 4; 0)		
5					

В ячейке **B2** возвращается приблизительно **0,216623**, значение функции плотности вероятности с $k = 3$ и $\lambda = 4$ для $x = 2,5$.

В ячейке **B4** возвращается приблизительно **0,229505**, значение функции интегрального распределения с $k = 3$ и $\lambda = 4$ для $x = 2,5$.

ZTEST

Функция **ZTEST** возвращает результат вычисления z-критерия.

Синтаксис функции:

=ZTEST(data; μ ; σ)

где:

- **data** — диапазон или массив, содержащее случайную выборку из генеральной совокупности (предполагается, что генеральная совокупность имеет нормальное распределение);
- **μ** — (известное) среднее генеральной совокупности;
- **σ** — (известное) среднеквадратичное отклонение генеральной совокупности. Если опущено, оно вычисляется исходя из данных выборки с использованием **STDEV(data)**.

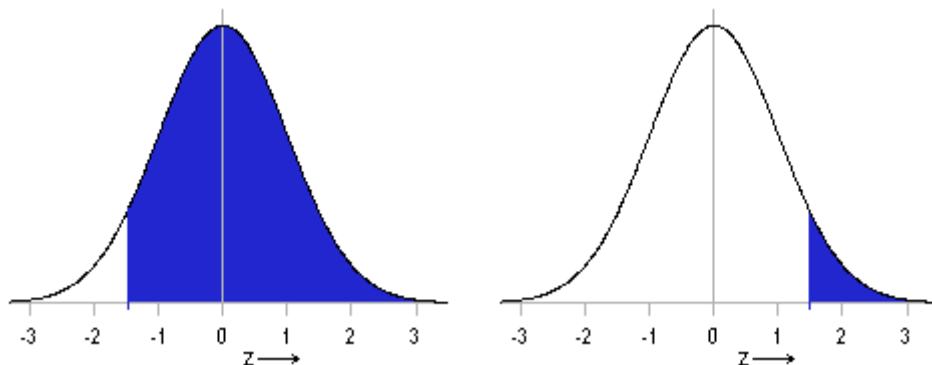
Функция **ZTEST** предназначена для проверки нулевой гипотезы состоящей в том, что выборка **data** получена из генеральной совокупности имеющей нормальное распределение со средним **μ** и средним квадратическим отклонением **σ** . Проверка нулевой гипотезы выполняется на основе статистики **Z**.

Функция **ZTEST** вычисляет z статистику:

$$z = \frac{m - \mu}{\sigma / \sqrt{n}}$$

где **m** — выборочное среднее, а **n** — объем выборки. Когда среднее и среднеквадратичное отклонение генеральной совокупности известно, z статистика строит стандартное нормальное распределение — то есть, нормальное распределение с средним =0 и среднеквадратичным отклонением =1.

Функция **ZTEST** возвращает одностороннюю интегральную вероятность — область под кривой стандартного нормального распределения справа от значения z (здесь заштриховано синим):



Выполняется проверка нуль-гипотезы, для **data** — выборки нормально распределенной случайной переменной с μ средним и σ среднеквадратичным отклонением. Возвращаемое значение 1 указывает, что нуль-гипотеза отклонена, то есть выборка не является случайной выборкой нормального распределения.

Выборка **data** должна содержать по крайней мере два числа.

ZTEST(A2:A20; 9; 2)

возвращает результат вычисления z-критерия для выборки в диапазоне **A2:A20**, извлеченной из генеральной совокупности с известными средними 9 и известным среднеквадратичным отклонением 2.

Calc вызывал неверные результаты для версии этой функции с 3-мя параметрами.

Как и со всей статистикой, необходимо хорошее понимание для получения надежных результатов.

Финансовые функции

Группа финансовых функций может успешно использоваться для решения экономических и финансовых задач.

Финансовые системы дат

В те дни, когда компьютеры еще не стали широко распространены, были разработаны различные системы для более простого ручного вычисления даты. Некоторые из них продолжают использоваться, и доступны в Calc, особенно в финансовых функциях. Это, к сожалению, делает сегодня жизнь немного сложнее.

Одна из проблем здесь заключается в том, чтобы определить количество дней между двумя учетными датами, *date1* и *date2* (*date2* после *date1*), соответственно *day1*, *month1*, *year1* и *day2*, *month2*, *year2*, согласно базовым вычислениям определяются следующим образом:

0: Американский метод (NASD), 12 месяцев 30 по дней в каждом (30US/360)

NASD была предшественником FINRA (оба — американские учреждения). Предполагается, что каждый месяц имеет 30 дней, и общее количество дней в году — 360. В этой системе:

- Если и *date1* и *date2* — обе — последний день февраля, *day2* изменяется на 30.
- Если *day1* — 31, или *date1* — последний день в феврале, *day1* изменяется на 30.
- Если *day1* — теперь 30, а *day2* — 31, *day2* изменяется на 30.

Каждый месяц, как теперь предполагают, имеет 30 дней и вычисляют результат. К сожалению Excel реализует эту систему неправильно — он опускает шаг 1. Для совместимости Calc теперь дает тот же самый результат, что и Excel.

1: Фактическое число дней в месяцах, точное число дней в году

Эта система - просто нормальный календарь; 28 дней в феврале и 365 дней в году, или в високосном году — 29 дней в феврале и 366 дней в году. Результат — **DAYS(date2; date1)**.

2: Фактическое число дней в месяце, год имеет 360 дней

Целые года рассчитываются как по 360 дней каждый, и фактические дни в остающейся части года добавляются.

3: Фактическое число дней в месяце, год имеет 365 дней

Целые года рассчитываются как по 365 дней каждый, и фактические дни в остающейся части года добавляются.

4: Европейский метод, 12 месяцев по 30 дней каждый (30E/360)

Предполагается, что каждый месяц имеет 30 дней, и общее количество дней в году — 360. Если или *day1* или *day2* — 31, они изменяются на 30. Каждый месяц, как теперь предполагают, имеет 30 дней, и вычисляют результат.

Отметьте, что даты февраля никогда не изменяются, потому что не бывает 31 февраля.

Используются и другие финансовые системы дат, но они не были реализованы в Excel или в Calc.

Другая проблема заключается в определении даты, которая отстоит от другой даты на несколько месяцев или даже лет (до или после). Это важно например в функции **COUPDAYBS**.

Правильным видится следующее — если первоначальная дата последний день месяца, то новая дата также последний день месяца, в противном случае используется тот же день в месяце (или ближайшей по возможности). Таким образом:

- За 6 месяцев до 2008-09-15 — 2008-03-15;
- За 6 месяцев до 2008-08-31 — 2008-2-29 (високосный год);
- За 6 месяцев до 2008-2-29 — 2007-08-31;
- За 6 месяцев до 2007-08-29 — 2007-2-28 (не високосный год);
- за 1 год до 2009-02-28 — 2008-02-29.

ACCRINT

Функция **ACCRINT** Вычисляет начисленные проценты по ценным бумагам с периодическими выплатами процентов.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции

=ACCRINT(issue; first_interest; settlement; rate; par; frequency; basis)

где:

- **issue** — дата выпуска ценных бумаг;
- **first_interest** — первый срок выплаты процентов по ценным бумагам;
- **settlement** — дата, когда подсчитываются проценты, накопившиеся до этого момента;
- **rate** — ежегодная номинальная процентная ставка (процентная ставка купона);
- **par** — номинальная стоимость ценных бумаг;
- **frequency** — ежегодное число выплат процентов (1, 2 или 4);
- **basis** — выбирается из списка возможных вариантов и указывает, как должен рассчитываться год:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый;

	A	B	C	D	E	F
1						
2		Дата выпуска	28.02.01			
3		Первый срок выплаты	31.08.01			
4		Дата подсчета	01.05.01			
5		Процентная ставка	0,1			
6		Номинальная стоимость	1000			
7		Число выплат процентов в год	2			
8						
9						
10						
11						

16,94444=ACCRINT(C2;C3;C4;C5;C6;C7)

ACCRINTM

Функция **ACCRINTM** вычисляет начисленные проценты по ценным бумагам, которые выплачиваются в подошедший срок платежа.

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=ACCRINTM(issue; settlement; rate; par; basis)

где:

- **issue** — дата выпуска ценных бумаг;
- **settlement** — дата, когда подсчитываются проценты, накопившиеся до этого момента;
- **rate** — ежегодная номинальная процентная ставка (процентная ставка купона);
- **par** — номинальная стоимость ценных бумаг;
- **basis** — выбирается из списка возможных вариантов и указывает, как должен рассчитываться год:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

	A	B	C	D	E	F
1						
2		Дата выпуска	01.04.01			
3		Дата подсчета	15.06.01			
4		Процентная ставка	0,1			
5		Номинальная стоимость	1000			
6						
7				20,54795	=ACCRINTM(C2;C3;C4;C5;3)	
8						

AMORDEGRC

Функция **AMORDEGRC** возвращает сумму амортизации за период с использованием пропорционально уменьшающейся амортизации (французская система).

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=AMORDEGRC(cost; purchase_date; first_period_end; salvage; period; rate; basis)

где:

- **cost** — стоимость приобретения;
- **purchase_date** — дата приобретения;
- **first_period_end** — дата завершения первого периода амортизации;
- **salvage** — ликвидационная стоимость;
- **period** — период для расчета суммы амортизации. 0 — начальный период (от **purchase_date** до **first_period_end**).

- **rate** — норма амортизации;
- **basis** — выбирается из списка возможных вариантов и указывает, как должен рассчитываться год:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Вычисляет сумму амортизационных отчислений за период, как пропорционально уменьшающуюся амортизацию, как она используется во французской системе бухгалтерского учета.

Срок службы **t** имущества — **1/rate**. Коэффициент амортизации **f** устанавливается исходя из:

- $0 \leq t < 3$: **f**=1.0
- $3 \leq t < 5$: **f**=1.5
- $5 \leq t < 6$: **f**=2.0
- $t \geq 6$: **f**=2.5

Сумма амортизация за период рассчитывается как:

- период 0: **cost * rate * YEARFRAC(purchase_date; first_period_end; basis)**
- период с 1 по t-3 : **rate * f * сумма_амортизации_за_предыдущий_период**
- период t-2 : **0.5 * сумма_амортизации_за_предыдущий_период**
- период t-1 : **сумма_амортизации_за_период_t-2**
- период $\geq t$: 0

	A	B	C	D	E	F
1						
2		Стоимость приобретения	1000			
3		Дата приобретения	01.02.06			
4		Дата завершения первого периода	31.12.06			
5		Ликвидационная стоимость	10			
6		Период	0			
7		Норма амортизации	0,1			
8						
9						
10						

228 =AMORDEGRC(C2;C3;C4;C5;C6;C7;1)

AMORLINC

Функция **AMORLINC** возвращает сумму амортизации за период с использованием линейной амортизации (французская система).

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=AMORLINC(cost; purchase_date; first_period_end; salvage; period; rate; basis)

где:

- **cost** — стоимость приобретения;

- **purchase_date** — дата приобретения;
- **first_period_end** — дата завершения первого периода амортизации;
- **salvage** — ликвидационная стоимость;
- **period** — период для расчета суммы амортизации. 0 — начальный период (от **purchase_date** до **first_period_end**).
- **rate** — норма амортизации;
- **basis** — выбирается из списка возможных вариантов и указывает, как должен рассчитываться год:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Вычисляет сумму амортизационных отчислений за период, с использованием линейного метода амортизации, как во французской системе бухгалтерского учета.

Учетный срок службы актива - $1/\text{rate}$, который состоит из начального неполного периода до завершения первого периода, потом некоторое количество полных периодов, затем неполный период до конца срока службы актива.

Амортизация в начальный период 0 (период с **purchase_date** до **first_period_end**) рассчитывается пропорционально и определяется как:

$= \text{cost} * \text{rate} * \text{YEARFRAC}(\text{purchase_date}; \text{first_period_end}; \text{basis})$.

Амортизации в последующие **полные** периоды определяется как:

$= \text{cost} * \text{rate}$.

В окончательном (неполном) периоде срока службы, амортизация уменьшает остаточную стоимость актива до ликвидационной стоимости, другими словами это:

cost - salvage - общая_сумма_амортизации_начисленная_за_предыдущие_периоды.

AMORLINC возвращает 0 после этого.

	A	B	C	D	E	F
1						
2		Стоимость приобретения	1000			
3		Дата приобретения	01.02.04			
4		Дата завершения первого периода	31.12.04			
5		Ликвидационная стоимость	10			
6		Период	8			
7		Норма амортизации	0,1			
8						
9						
10						

100 = AMORLINC(C2;C3;C4;C5;C6;C7;1)

Calc и Excel осуществляют вычисления по-разному, когда дата закупки — конец периода. Excel считает в качестве начального периода 0 первый полный период. Calc (обоснованно) принимает начальный период 0 нулевой длины и таким образом правильно возвращает нулевую сумму амортизации. Пример:

$=\text{AMORLINC}(1000; "2008-12-31"; "2008-12-31"; 100; 0; 0.25; 1)$

возвращает 0 в Calc и 250 в Excel. В этом примере окончательным неполным периодом

(возвращающим сумму амортизации 150) является 3-й период в Excel и 4-й период в Calc.

COUPDAYBS

Функция **COUPDAYBS** возвращает число дней с предыдущей даты выплаты процентов по ценным бумагам до даты соглашения (даты покупки ценных бумаг).

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=COUPDAYBS(settlement; maturity; frequency; basis)

где:

- **settlement** — дата покупки ценных бумаг;
- **maturity** — дата, когда наступает срок платежа по ценным бумагам. Эта дата определяет момент истечения срока действия ценных бумаг;
- **frequency** — количество выплат процентов за год. Для ежегодных выплат — 1; для полугодовых выплат — 2; для ежеквартальных выплат — 4;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

	A	B	C	D	E	F
1						
2		Дата покупки	25.01.07			
3		Дата платежа	15.11.09			
4		Количество выплат в год	2			
5						
6				71	=COUPDAYBS(C2;C3;C4;1)	
7						

Облигации, выпущенные 15 ноября 1999, с десятилетним сроком; срок платежа — 15 ноября 2009 г. Вы впоследствии покупаете их на вторичном рынке, с датой расчёта от 25 января 2007. Проценты выплачиваются раз в полгода (частота — 2); таким образом проценты подлежат выплате 15 мая и 15 ноября каждый год, в течение срока облигации. Используя $basis = 1$, имеем 71 (фактический) день в процентный период непосредственно перед датой покупки, с 15 ноября 2006 до 25 января 2007 (поэтому = “25.01.2007” – “15.11.2006” также возвращает 71).

COUPDAYSNC

Функция **COUPDAYSNC** возвращает число дней между датой соглашения (датой покупки ценных бумаг) и следующей датой выплаты процентов по ценным бумагам.

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=COUPDAYSNC(settlement; maturity; frequency; basis)

где:

- **settlement** — дата покупки ценных бумаг;
- **maturity** — дата, когда наступает срок платежа по ценным бумагам. Эта дата определяет момент истечения срока действия ценных бумаг;
- **frequency** — количество выплат процентов за год. Для ежегодных выплат — 1; для полугодовых выплат — 2; для ежеквартальных выплат — 4;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

	A	B	C	D	E	F
1						
2		Дата покупки	25.01.07			
3		Дата платежа	15.11.09			
4		Количество выплат в год	2			
5						
6				110	=COUPDAYSNC(C2;C3;C4;4)	
7						

Облигации, выпущенные 15 ноября 1999, с десятилетним сроком; срок платежа — 15 ноября 2009 г. Вы впоследствии покупаете их на вторичном рынке, с датой расчёта от 25 января 2007. Проценты выплачиваются раз в полгода (частота — 2); таким образом проценты подлежат выплате 15 мая и 15 ноября каждый год, в течение срока облигации. Используя $\text{basis} = 4$, имеем 110 дней от даты покупки 25 января 2007 до даты следующей выплаты процентов 15 мая 2007.

COUPNCD

Функция **COUPNCD** возвращает дату выплаты процентов по ценным бумагам следующую после даты соглашения (даты покупки ценных бумаг).

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=COUPNCD(settlement; maturity; frequency; basis)

где:

- **settlement** — дата покупки ценных бумаг;
- **maturity** — дата, когда наступает срок платежа по ценным бумагам. Эта дата определяет момент истечения срока действия ценных бумаг;
- **frequency** — количество выплат процентов за год. Для ежегодных выплат — 1; для полугодовых выплат — 2; для ежеквартальных выплат — 4;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;

- 2 - точное число дней в месяце, год имеет 360 дней;
- 3 - точное число дней в месяце, год имеет 365 дней;
- 4 - европейский метод, 12 месяцев по 30 дней каждый.

	A	B	C	D	E	F
1						
2		Дата покупки	25.01.07			
3		Дата платежа	15.11.09			
4		Количество выплат в год	2			
5						
6			15.05.07	=COUPNCD(C2;C3;C4;3)		
7						

Облигации, выпущенные 15 ноября 1999, с десятилетним сроком; срок платежа — 15 ноября 2009 г. Вы впоследствии покупаете их на вторичном рынке, с датой расчёта от 25 января 2007 г. Проценты выплачиваются раз в полгода (частота — 2); таким образом проценты подлежат выплате 15 мая и 15 ноября каждый год, в течение срока облигации. Поэтому следующая дата выплаты процентов после даты покупки 25 января 2007 — 15 мая 2007.

COUPNUM

Функция **COUPNUM** возвращает количество купонов (платежей процентов) между датой покупки ценных бумаг и датой погашения.

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=COUPNUM(settlement; maturity; frequency; basis)

где:

- **settlement** — дата покупки ценных бумаг;
- **maturity** — дата, когда наступает срок платежа по ценным бумагам. Эта дата определяет момент истечения срока действия ценных бумаг;
- **frequency** — количество выплат процентов за год. Для ежегодных выплат — 1; для полугодовых выплат — 2; для ежеквартальных выплат — 4;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

	A	B	C	D	E	F
1						
2		Дата покупки	25.01.07			
3		Дата платежа	15.11.09			
4		Количество выплат в год	2			
5						
6			6	=COUPNUM(C2;C3;C4;3)		
7						

Облигации, выпущенные 15 ноября 1999, с десятилетним сроком; срок платежа — 15 ноября

2009 г. Вы впоследствии покупаете их на вторичном рынке, с датой расчёта от 25 января 2007 г. Проценты выплачиваются раз в полгода (частота — 2); таким образом проценты подлежат выплате 15 мая и 15 ноября каждый год, в течение срока облигации. Проценты выплачиваются 6 раз: 15 мая 2007, 15 ноября 2007, 15 мая 2008, 15 ноября 2008, 15 мая 2009 и 15 ноября 2009.

COUPPCD

Функция **COUPPCD** возвращает дату купона (выплаты процентов), которая предшествует дате покупки ценных бумаг.

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=COUPPCD(settlement; maturity; frequency; basis)

где:

- **settlement** — дата покупки ценных бумаг;
- **maturity** — дата, когда наступает срок платежа по ценным бумагам. Эта дата определяет момент истечения срока действия ценных бумаг;
- **frequency** — количество выплат процентов за год. Для ежегодных выплат — 1; для полугодовых выплат — 2; для ежеквартальных выплат — 4;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

	A	B	C	D	E	F
1						
2		Дата покупки	25.01.07			
3		Дата платежа	15.11.09			
4		Количество выплат в год	2			
5						
6				15.11.06	=COUPPCD(C2;C3;C4;3)	
7						

Возвращает 15 ноября 2006 в виде числа, представляющего дату; выберите соответствующий формат, чтобы отобразить дату, в том виде, который Вы предпочитаете. Облигации, выпущенные 15 ноября 1999, с десятилетним сроком; срок платежа — 15 ноября 2009 г. Вы впоследствии покупаете их на вторичном рынке, с датой расчёта от 25 января 2007 г. Проценты выплачиваются раз в полгода (частота — 2); таким образом проценты подлежат выплате 15 мая и 15 ноября каждый год, в течение срока облигации. Дата выплаты процентов перед покупкой ценных бумаг 25 января 2007 — 15 ноября 2006.

CUMIPMT

Функция **CUMIPMT** возвращает кумулятивную (нарастающим итогом) сумму процентов, выплачиваемые по кредиту в указанный период.

Синтаксис функции:

=CUMIPMT(rate; numperiods; principal; start; end; type)

где:

- **rate** — процентная ставка за период;
- **numperiods** — общее количество периодов выплат;
- **principal** — начальная заимствованная сумма;
- **start** — первый включаемый период. Периоды нумеруются, начиная с 1.
- **end** — последний включаемый период;
- **type** — когда осуществляются платежи:
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

В кредите с фиксированной ставкой, где можно делать постоянными платежи за каждый отчетный период на погашение кредита в течение срока, часть платежа каждого периода — проценты за непогашенную сумму кредита, а часть — погашение кредита. В течение долгого времени (пока Вы погашаете кредит), сумма процентов становится меньше, а погашение основной суммы кредита становится больше.

IPMT возвращает проценты, уплаченные в указанный период. **PPMT** возвращает сумму погашения кредита, уплачиваемую в этот же период. Вместе в целом они составляют периодическую выплату, задаваемую **PMT**.

CUMIPMT возвращает кумулятивную сумму процентов, уплачиваемую в течение периодов от **start** до **end** включительно — то есть, сумму **IPMT** за это время.

	A	B	C	D	E	F
1						
2		Годовая процентная ставка	5,50%			
3		Срок кредита (лет)	2			
4		Сумма кредита	5000			
5		Начальный период	4			
6		Конечный период	6			
7						
8				-57,80	=CUMIPMT(C2/12;C3*12;C4;C5;C6;0)	
9						

Вы взяли кредит на 2 года в размере 5000 единиц валюты с ежегодной процентной ставкой 5,5%, делая ежемесячные платежи в конце месяца. Проценты, которые Вы платите с 4-ого по 6-ой месяцы включительно, составляют -57,80 единиц валюты. Они устанавливаются отрицательными, потому что Вы их платите.

CUMIPMT форматирует результат как валюту, если ячейка имеет форматирование по умолчанию. Она таким образом показывает реальную сумму валюты. Функция **CUMIPMT_ADD** совместима с Excel; она не применяет форматирование и может таким образом показать дробные значения, например 57.5412415... Сумма, возвращаемая **CUMIPMT** по-прежнему может быть дробной — при отображении она округляется к самому близкому значению. Заметим, что ваш кредитор может выполнять округление различными способом (например всегда вниз).

Согласно разрабатываемой версии стандарта ODF, эта функция должна быть удалена, а **CUMIPMT_ADD** должна быть переименована в **CUMIPMT**.

В отличие от **PMT**, **IPMT**, **PPMT**, эта функция не имеет параметра **finalbalance**.

CUMIPMT_ADD

Функция **CUMIPMT_ADD** возвращает кумулятивную (нарастающим итогом) сумму процентов, выплачиваемые по кредиту в указанный период.

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=CUMIPMT_ADD(rate; numperiods; principal; start; end; type)

где:

- **rate** — процентная ставка за период;
- **numperiods** — общее количество периодов выплат;
- **principal** — первоначальная сумма кредита;
- **start** — первый включаемый период. Периоды нумеруются, начиная с 1.
- **end** — последний включаемый период;
- **type** — когда осуществляются платежи:
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

В кредите с фиксированной ставкой, где можно делать постоянными платежи за каждый отчетный период на погашение кредита в течение срока, часть платежа каждого периода — проценты за непогашенную сумму кредита, а часть — погашение кредита. В течение долгого времени (пока Вы погашаете кредит), сумма процентов становится меньше, а погашение основной суммы кредита становится больше.

IPMT возвращает проценты, уплаченные в указанный период. **PPMT** возвращает сумму погашения кредита, уплачиваемую в этот же период. Вместе в целом они составляют периодическую выплату, задаваемую **PMT**.

CUMIPMT_ADD возвращает кумулятивную сумму процентов, уплачиваемую в течение периодов от **start** до **end** включительно — то есть, сумму **IPMT** за это время.

	A	B	C	D	E	F	
1							
2		Годовая процентная ставка	5,50%				
3		Срок кредита (лет)	2				
4		Сумма кредита	5000				
5		Начальный период	4				
6		Конечный период	6				
7							
8							
9							

-57,80 =CUMIPMT_ADD(C2/12;C3*12;C4;C5;C6;0)

Вы взяли кредит на 2 года в размере 5000 единиц валюты с ежегодной процентной ставкой 5,5%, делая ежемесячные платежи в конце месяца. Проценты, которые Вы платите с 4-ого по 6-ой месяцы включительно, составляют -57,80 единиц валюты. Они устанавливаются отрицательными, потому что Вы их платите.

CUMIPMT_ADD совместима с Excel; она не применяет форматирование валюты и таким образом может показать дробные значения, например 57.5412415... Функция **CUMIPMT** форматирует результат как денежный, если ячейка имеет форматирование по умолчанию, и таким образом показывает вещественное (округленное) значение валюты, например 57.54 \$; значение может все еще быть дробным — она влияет только на отображение.

Согласно разрабатываемой версии стандарта ODF, эта функция должна быть переименована

в CUMIPMT, а существующая CUMIPMT удалена.

CUMPRINC

Функция **CUMPRINC** возвращает кумулятивную (нарастающим итогом) сумму, выплачиваемую в погашение основной суммы кредита в промежутке между двумя периодами.

Синтаксис функции:

=CUMPRINC(rate; numperiods; principal; start; end; type)

где:

- **rate** — процентная ставка за период;
- **numperiods** — общее количество периодов выплат;
- **principal** — первоначальная сумма кредита;
- **start** — первый включаемый период. Периоды нумеруются, начиная с 1.
- **end** — последний включаемый период;
- **type** — когда осуществляются платежи:
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

В кредите с фиксированной ставкой, где можно делать постоянными платежи за каждый отчетный период на погашение кредита в течение срока, часть платежа каждого периода — проценты за непогашенную сумму кредита, а часть — погашение кредита. В течение долгого времени (пока Вы погашаете кредит), сумма процентов становится меньше, а погашение основной суммы кредита становится больше.

IPMT возвращает проценты, уплаченные в указанный период. **PPMT** возвращает сумму погашения кредита, уплачиваемую в этот же период. Вместе в целом они составляют периодическую выплату, задаваемую **PMT**.

CUMPRINC возвращает кумулятивную (нарастающим итогом) сумму, выплачиваемую в погашение основной суммы кредита в течение периодов от **start** до **end** включительно — то есть, сумму **PPMT** за это время.

	A	B	C	D	E	F
1						
2		Годовая процентная ставка	5,50%			
3		Срок кредита (лет)	2			
4		Сумма кредита	5000			
5		Начальный период	4			
6		Конечный период	6			
7						
8				-603,63	=CUMPRINC(C2/12;C3*12;C4;C5;C6;0)	
9						

Вы взяли кредит на 2 года в размере 5000 единиц валюты с ежегодной процентной ставкой 5,5%, делая ежемесячные платежи в конце месяца. Сумма кредита, которую Вы погасили с 4-ого по 6-й месяц включительно, -603.63 единицы валюты. Она устанавливается отрицательной, потому что Вы ее платите.

CUMPRINC форматирует результат как денежное значение, если ячейка имеет форматирование по умолчанию. Она таким образом показывает вещественное денежное значение. Функция **CUMPRINC_ADD** совместима с Excel; она не применяет форматирование и может таким образом показать дробные значения, например 603.6298701...

Сумма, возвращаемая **CUMPRINC** по прежнему может быть дробной — при отображении она округляется к самому близкому значению. Заметим, что ваш кредитор может выполнять округление различными способом (например всегда вниз).

Согласно разрабатываемой версии стандарта ODFE, эта функция должна быть удалена, а **CUMPRINC_ADD** должна быть переименована в **CUMPRINC**.

В отличие от **PMT**, **IMPT**, **PPMT**, эта функция не имеет параметра **finalbalance**.

CUMPRINC_ADD

Функция **CUMPRINC** возвращает кумулятивную (нарастающим итогом) сумму, выплачиваемую в погашение основной суммы кредита в промежутке между двумя периодами.

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=CUMPRINC_ADD(rate; numperiods; principal; start; end; type)

где:

- **rate** — процентная ставка за период;
- **numperiods** — общее количество периодов выплат;
- **principal** — первоначальная сумма кредита;
- **start** — первый включаемый период. Периоды нумеруются, начиная с 1.
- **end** — последний включаемый период;
- **type** — когда осуществляются платежи:
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

В кредите с фиксированной ставкой, где можно делать постоянными платежи за каждый отчетный период на погашение кредита в течение срока, часть платежа каждого периода — проценты за непогашенную сумму кредита, а часть — погашение кредита. В течение долгого времени (пока Вы погашаете кредит), сумма процентов становится меньше, а погашение основной суммы кредита становится больше.

IPMT возвращает проценты, уплаченные в указанный период. **PPMT** возвращает сумму погашения кредита, уплачиваемую в этот же период. Вместе в целом они составляют периодическую выплату, задаваемую **PMT**.

CUMPRINC_ADD возвращает кумулятивную (нарастающим итогом) сумму, выплачиваемую в погашение основной суммы кредита в течение периодов от **start** до **end** включительно — то есть, сумму **PPMT** за это время.

	A	B	C	D	E	F
1						
2		Годовая процентная ставка	5,50%			
3		Срок кредита (лет)	2			
4		Сумма кредита	5000			
5		Начальный период	4			
6		Конечный период	6			
7						
8				-603,63	=CUMPRINC_ADD(C2/12;C3*12;C4;C5;C6;0)	
9						

Вы взяли кредит на 2 года в размере 5000 единиц валюты с ежегодной процентной ставкой

5,5%, делая ежемесячные платежи в конце месяца. Сумма кредита, которую Вы погасили с 4-ого по 6-й месяц включительно, -603.63 единицы валюты. Она устанавливается отрицательной, потому что Вы ее платите.

CUMPRINC_ADD совместима с Excel; она не применяет форматирование валюты и таким образом может показать дробные значения, например 603.6298701... Функция **CUMPRINC** форматирует результат как денежный, если ячейка имеет форматирование по умолчанию, и таким образом показывает вещественное (округленное) значение валюты, например 603.63 \$; значение может все еще быть дробным — она влияет только на отображение.

Согласно разрабатываемой версии стандарта ODFD, эта функция должна быть переименована в **CUMPRINC**, а существующая **CUMPRINC** удалена.

В отличие от **PMT**, **IPMT**, **PPMT**, эта функция не имеет параметра **finalbalance**.

DB

Функция **DB** возвращает сумму амортизации актива за данный год с использованием фиксированной ставки методом уменьшающегося остатка (метод ускоренной амортизации, при котором ежегодная амортизация рассчитывается как процент от текущей балансовой стоимости актива).

Синтаксис функции:

=DB(originalcost; salvagevalue; lifetime; year; months1styear)

где:

- **originalcost** — первоначальная стоимость актива;
- **salvagevalue** — это стоимость актива в конце срока амортизации (иногда называется остаточной стоимостью актива);
- **lifetime** — число лет, за которые обесценивается актив (иногда называется периодом амортизации);
- **year** — год, для которого вычисляется сумма амортизации;
- **months1styear** - количество месяцев в первом году (по умолчанию 12, если опущен).

Для расчета суммы амортизации, DB использует фиксированную ставку на протяжении всего срока службы актива, задаваемую формулой

$$rate = 1 - \left(\frac{salvagevalue}{originalcost} \right)^{\left(\frac{1}{lifetime} \right)}$$

округляемую до 3-х десятичных знаков.

Амортизация в любом году определяется по формуле

$$стоимость_на_начало_года * rate$$

где

$$стоимость_на_начало_года = originalcost - сумма_амортизации_за_предыдущие_периоды$$

Если **months1styear** — меньше чем 12, норма амортизации, используемая в течение первого и последнего годов

$$rate * количество_месяцев_в_году / 12.$$

	A	B	C	D	E
1					
2		Первоначальная стоимость	10000		
3		Ликвидационная стоимость	1000		
4		Срок службы	5		
5		Расчитываемый год	1		
6					
7				3 690,00	=DB(C2;C3;C4;C5)
8					

Возвращается примерно 3690 единиц валюты, являющихся суммой амортизации за первый год для актива, стоимость которого 10000, а остаточная стоимость составляет 1000 через 5 лет.

Этот метод незначительно недооценивает амортизацию, если **months1styear** задается а не равен 12.

DDB

Функция **DDB** возвращает сумму амортизации актива за данный год, используя метод двойного (или другого коэффициента) уменьшения остатка (метод начисления амортизации, при котором ставка амортизации, составляющая удвоенное значение дроби $1/n$ (n — срок службы, выраженный в годах), умножается на остаточную стоимость актива). Сумма амортизации различается для отдельных периодов: для начального периода она максимальна, а для конечного — минимальна.

Синтаксис функции:

=DDB(originalcost; salvagevalue; lifetime; year; factor)

где:

- **originalcost** — первоначальная стоимость актива;
- **salvagevalue** — это стоимость актива в конце срока амортизации (иногда называется остаточной стоимостью актива);
- **lifetime** — число периодов (обычно лет), за которые обесценивается актив (иногда называется периодом амортизации);
- **year** — год, для которого вычисляется сумма амортизации;
- **factor** — коэффициент для задания нормы амортизации (2 если опущен).

Для расчета амортизации, DDB использует фиксированную норму амортизации. Когда **factor** = 2 это метод двойного уменьшения остатка (потому что он удваивает прямолинейную норму, которая обесценивала бы актив к нулю). Норма амортизации определяется по следующей формуле:

$$rate = \frac{factor}{lifetime}$$

Амортизации каждый год рассчитывается как

MINIMUM(book_value_at_start_of_year * rate; book_value_at_start_of_year - salvagevalue)

где:

- **book_value_at_start_of_year** — балансовая стоимость актива в начале года.

Таким образом актив амортизируется по норме, пока балансовая стоимость не достигнет остаточной стоимости.

	A	B	C	D	E	F	G
1							
2		Первоначальная стоимость		12000			
3		Ликвидационная стоимость		3000			
4		Срок службы		5			
5							
6							
7		Год	Сумма амортизации				
8		1	4 800,00 руб.		=DDB(\$D\$2;\$D\$3;\$D\$4;B8)		
9		2	2 880,00 руб.		=DDB(\$D\$2;\$D\$3;\$D\$4;B9)		
10		3	1 320,00 руб.		=DDB(\$D\$2;\$D\$3;\$D\$4;B10)		
11		4	0,00 руб.		=DDB(\$D\$2;\$D\$3;\$D\$4;B11)		
12		5	0,00 руб.		=DDB(\$D\$2;\$D\$3;\$D\$4;B12)		
13							

Отметьте, что этот метод, возможно, не амортизирует актив полностью к его остаточной стоимости. Например, **DDB(1000; 0; 3; 1) + DDB (1000; 0; 3; 2) + DDB (1000; 0; 3; 3) = 962.96**, оставляет остаточную стоимость 37,04 вместо 0. Функция **VDB** — альтернатива, которая преодолевает это.

DISC

Функция **DISC** возвращает учетную ставку по ценным бумагам.

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=DISC(settlementdate; maturitydate; price; redemptionvalue; basis)

где:

- **settlementdate** — дата сделки по ценным бумагам;
- **maturitydate** — дата погашения ценных бумаг;
- **price** — курс ценной бумаги;
- **redemptionvalue** — цена выкупа ценных бумаг;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

DISC вычисляет учетную ставку, используя эту формулу:

$$\text{DISC} = \frac{(\text{redemptionvalue} - \text{price})}{(\text{redemptionvalue} * \text{YEARFRAC}(\text{settlementdate}; \text{maturitydate}; \text{basis}))}$$

	A	B	C	D	E
1					
2		Дата покупки	21.12.07		
3		Дата погашения	15.02.08		
4		Курс	98,9		
5		Цена досрочного выкупа	100		
6					
7			7,2%	=DISC(C2;C3;C4;C5;1)	
8					

Учетная ставка — приблизительно 7,2 %.

Calc как и Excel явно использует **YEARFRAC** в расчете **DISC**. И Excel и Calc вычисляют **YEARFRAC** неправильно при определенных условиях, и эти (небольшие) ошибки распространяются на **DISC**.

DOLLARDE

Функция **DOLLARDE** преобразует дробное представление числа в десятичное число.

Синтаксис функции:

=DOLLARDE(fractionalrep; denominator)

где:

- **fractionalrep** — дробное представление. Иногда цена на ценные бумаги, например, может быть выражена как 2.03, означающее 2 и 3/16 доллара;
- **denominator** — знаменатель, например, 16 в примере выше.

Функция **DOLLARDE** преобразует дробное представление в десятичное число. Несмотря на свое название, она возвращает число, а не денежное значение. Обратная функция — **DOLLARFR**.

	A	B	C	D	E
1					
2		2,1875	=DOLLARDE(2,03; 16)		
3					

Возвращается 2,1875 поскольку число. 2 + 3/16 равняется 2,1875.

DOLLARFR

Функция **DOLLARFR** преобразует десятичное число в дробное представление этого числа.

Синтаксис функции:

=DOLLARFR(decimal; denominator)

где:

- **decimal** — десятичное число;
- **denominator** — знаменатель для дробного представления.

Иногда цена на ценные бумаги, например, может быть выражена как 2.03, означающее 2 и 3/16 доллара. Как десятичное число это — 2,1875.

DOLLARFR преобразует десятичное число в дробное представление. Несмотря на свое название, она возвращает число, а не денежное значение. Обратная функция — **DOLLARDE**.

	A	B	C	D	E
1					
2		2,03=DOLLARFR(2,1875; 16)			
3					

Возвращается 2,03 в виде числа. 2 + 3/16 эквивалентно 2,1875.

DURATION

Функция **DURATION** возвращает количество периодов, необходимых инвестициям для достижения определенной стоимости.

Синтаксис функции:

=DURATION(rate; present_value; future_value)

где:

- **rate** — процентная ставка за период, которая применяется к инвестициям;
- **present_value** — нынешняя стоимость инвестиций;
- **future_value** — желательная стоимость инвестиций в будущем.

DURATION возвращает оценку количества периодов, требуемых для изменения **present_value** в **future_value** при постоянной процентной ставке, начисляемой каждый период.

Решается уравнение:

$$present_value * (1 + rate)^{duration} = future_value$$

дающее результат:

$$DURATION(rate; present_value; future_value) = LOG(future_value / present_value; 1 + rate)$$

Результат точен для целых периодов, и приближителен для неполных периодов.

	A	B	C	D	E
1					
2		Процентная ставка		10,00%	
3		Текущая стоимость		100	
4		Будущая стоимость		121	
5					
6			2=DURATION(C2;C3;C4)		
7					

100 руб. инвестированные под 10 % годовых, стоят 110 руб. в следующем году и 121 руб. через два года.

Функция Calc **DURATION** реализуется как **G_DURATION** в Gnumeric, и не реализована в Excel.

Функция **DURATION**, которая реализована в Gnumeric и Excel (продолжительность Макалея) реализована как **DURATION_ADD** в Calc.

Неточность для дробных периодов возникает потому, что проценты вычисляются линейно в течение периода. Например, 100 руб под 10% стоят 105 руб через полгода, но **DURATION(10%; 100; 105)** не возвращает в точности 0,5. Это может иметь скорее теоретическое, чем практическое значение.

DURATION_ADD

Функция **DURATION_ADD** возвращает продолжительность Макалея для ценных бумаг.

Эта функция только доступна, если установлен *пакет Анализа*.

Синтаксис функции:

=DURATION_ADD(settlementdate; maturitydate; rate; yield; frequency; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **rate** — (ежегодная) процентная ставка по ценным бумагам;
- **yield** — (ежегодная) доходность по ценным бумагам (доход по ценным бумагам, выраженный в процентах к их цене);
- **frequency** — количество выплат процентов в год (1, 2 или 4);
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция возвращает продолжительность Макалея для ценных бумаг, которая является мерой неустойчивости цены и чувствительности к процентным ставкам (зависимость цены для ценных бумаг от изменения процентных ставок). Это — средневзвешенная по времени величина получаемой прибыли по ценным бумагам, где вес в течение каждого раза - текущая стоимость дивидендов от дохода, разделенного непосредственно на текущую стоимость ценных бумаг; результат — таким образом:

$$\sum_i time_until_receipt_i * \frac{present_value_of_income_i}{present_value_of_security}$$

	A	B	C	D	E	F
1						
2		Дата сделки	28.02.08			
3		Дата погашения	31.08.10			
4		Процентная ставка	5,00%			
5		Доходность	6,00%			
6		Количество выплат	2			
7						
8				2,33	=DURATION_ADD(C2;C3;C4;C5;C6;0)	
9						

Функция возвращает приблизительно 2.33 года.

Эта функция реализована как **DURATION** в Gnumeric и Excel.

Функция **DURATION** в Calc полностью отличается.

Эта функция использует в вычислении **YEARFRAC**. Есть известные небольшие погрешности (и в Calc, и в Excel) в **YEARFRAC**.

EFFECT_ADD

Функция **EFFECT_ADD** возвращает эффективную ставку сложных процентов по заданной номинальной процентной ставке.

Синтаксис функции:

=EFFECT_ADD(nom_rate; num)

где:

- **nom_rate** — номинальная годовая процентная ставка.
- **num** — количество выплат процентов по кредиту, в течение периода к которому относится **nom_rate**.

Если инвестиции имеют номинальную ставку, скажем за год, но проценты выплачиваются и начисляются скажем, каждый квартал, то выплачиваемые каждый квартал проценты будут приносить проценты. Это увеличивает эффективное значение. Эта функция возвращает эффективную норму — то есть, норму, которую нужно было бы заплатить в конце (скажем) года, чтобы получить ту же самую отдачу.

Используется формула:

$$\text{EFFECT_ADD} = \left(1 + \frac{\text{nom_rate}}{\text{num}} \right)^{\text{num}} - 1$$

	A	B	C	D	E	F	G
1							
2		Номинальная процентная ставка			6,00%		
3		Количество выплат за год			4		
4							
5					0,06136	=EFFECT_ADD(E2;E3)	
6							

Возвращается приблизительно 0,06136..., что является эффективной ставкой вложений с номинальной ставкой 6 % годовых при квартальных выплатах процентов.

Согласно проекту стандарт ODFD, эта функция будет заменена новой функцией **EFFECT**.

Расчет предполагает, что проценты начисляются в конце в точности равных периодов. В действительности различные кварталы года, например, имеют различное количество дней.

EFFECTIVE

Функция **EFFECTIVE** возвращает эффективную ставку сложных процентов по заданной номинальной процентной ставке.

Синтаксис функции:

=EFFECTIVE(nom_rate; num)

где:

- **nom_rate** — номинальная годовая процентная ставка.
- **num** — количество выплат процентов по кредиту, в течение периода к которому относится **nom_rate**.

Если инвестиции имеют номинальную ставку, скажем за год, но проценты выплачиваются и начисляются скажем, каждый квартал, то выплачиваемые каждый квартал проценты будут приносить проценты. Это увеличивает эффективное значение. Эта функция возвращает эффективную норму — то есть, норму, которую нужно было бы заплатить в конце (скажем) года, чтобы получить ту же самую отдачу.

Используется формула:

$$\text{EFFECTIVE} = \left(1 + \frac{\text{nom rate}}{\text{num}} \right)^{\text{num}} - 1$$

	A	B	C	D	E
1					
2		Номинальная процентная ставка	6,00%		
3		Количество выплат в год	4		
4					
5			6,14%	=EFFECTIVE(C2;C3)	
6					

Возвращается примерно 6,14%, что является эффективной ставкой капиталовложений с номинальной ставкой 6% годовых при квартальных выплатах процентов.

Согласно проекту стандарта ODFD, эта функция будет заменена новой функцией **EFFECT**.

Расчет предполагает, что проценты начисляются в конце в точности равных периодов. В действительности различные кварталы года, например, имеют различное количество дней.

FV

Функция **FV** возвращает будущую стоимость капиталовложений на основе периодических постоянных (равных по величине сумм) платежей и постоянной процентной ставки.

Синтаксис функции:

=FV(rate; numperiods; payment; presentvalue; type)

где:

- **rate** — (фиксированная) процентная ставка за период;
- **numperiods** — общее количество периодов платежей в сроке.
- **payment** — сумма платежа, выполняемого каждый отчетный период. Это значение не может меняться в течение всего периода платежей. Обычно **payment** состоит из основного платежа и платежа по процентам, но не включает других налогов и сборов. Если приводится **presentvalue**, этот аргумент может быть опущен (по умолчанию 0).
- **presentvalue** — единовременно выплачиваемая сумма платежа в начале срока (необязательный — по умолчанию 0). При кредите, это обычно сумма кредита; при ссуде под залог — обычно 0.
- **type** — когда осуществляются платежи:
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

Стоимость денег зависит от времени; например, 100 \$ сегодня стоили бы 110 \$ через год если инвестированы со ставкой 10%.

Функция **FV** может использоваться для:

- Вычисления будущей стоимости единовременных вложений;
- Расчета будущей стоимости периодических платежей, по которым начисляются сложные проценты за определенное количество периодов при заданной процентной ставке.

	A	B	C	D	E	F	G
1							
2		Номинальная процентная ставка	5,00%				
3		Количество выплат за год	3				
4		Сумма платежа	-1000				
5							
6					3152,50	=FV(E2;E3;E4;0;0)	
7							

Возвращается **3 152,50** в денежных единицах. Вы платите по 1000 в конце каждого года в течение 3 лет. Если предположить, что процентная ставка составляет 5% вы ожидаете получить 3152,50 в конце срока.

8		
9	Номинальная процентная ставка	7,00%
10	Количество выплат за год	10
11	Сумма платежа	-1400
12	Сумма кредита	10000
13		
14		-328,49=FV(E9;E10;E11;E12;0)
15		

Возвращается будущая стоимость -328,49 в денежных единицах.

Убедитесь, что вы последовательны в выборе единиц измерения для задания аргументов **rate** и **numperiods**. Если вы делаете ежемесячные выплаты по четырехгодичному кредиту из расчета 12 процентов годовых, то используйте 12%/12 для задания аргумента **rate** и 4*12 для задания аргумента **numperiods**. Если вы выполняете ежегодные платежи по тому же кредиту, то используйте 12% для задания аргумента **rate** и 4 для задания аргумента **numperiods**.

Все аргументы, означающие денежные средства, которые должны быть выплачены (например сберегательные вклады), представляются отрицательными числами; денежные средства, которые должны быть получены (например дивиденды), представляются положительными числами.

Будьте осторожны, понимайте, как эта функция начисляет проценты каждого периода. Многие финансовые калькуляторы позволяют Вам устанавливать отдельный период наращивания — электронные таблицы не позволяют этого. Выберите соответствующую процентную ставку.

FVSCCHEDULE

Функция **FVSCCHEDULE** возвращает будущую стоимость первоначальной основной суммы после применения ряда ставок сложных процентов. Функция используется для вычисления будущей стоимости инвестиции с переменной процентной ставкой.

Синтаксис функции:

=FVSCCHEDULE(principal; interestrates)

где:

- **principal** — первоначальная стоимость инвестиций.
- **interestrates** — диапазон или массив, содержащий график процентных ставок, которые применяются в каждом из отчетных периодов.

FVSCCHEDULE рассчитывает будущую стоимость путем применения каждой процентной ставки в свою очередь, и начисляет сложные проценты. Формула:

$$principal * (1 + rate_1) * (1 + rate_2) * (1 + rate_3) * \dots$$

где $rate_i$ — i-я процентная ставка в **interestrates**.

	A	B	C	D	E	F	G	H
1								
2		Первоначальная сумма			1 000,00			
3								
4					1190,91	=FVSCCHEDULE(E2; {0,05;0,06;0,07})		
5								

Возвращается 1 190.91, будущая стоимость 1000 через 3 года, где, как Вы считаете,

соответствующие процентные ставки будут 5 % в первом году, 6 % во втором году и 7 % в третьем году.

INTRATE

Функция **INTRATE** возвращает эквивалентную ежегодную процентную ставку для инвестиций, купленных по одной цене и проданных по другой.

Синтаксис функции:

=INTRATE(settlementdate; maturitydate; purchasevalue; maturityvalue; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **purchasevalue** - сумма, выплаченная за эти ценные бумаги;
- **maturityvalue** — сумма, полученная за эти ценные бумаги;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Возвращенная эквивалентная процентная ставка — (неначисленная) процентная ставка, которую нужно было бы заплатить за инвестиции **purchasevalue**, чтобы превратить ее в **maturityvalue** при погашении. Эта функция может быть полезна для краткосрочных облигаций с нулевым купоном.

Используется формула:

$$\text{INTRATE} = \frac{(\text{maturityvalue} - \text{purchasevalue})}{\text{purchasevalue}} * \frac{\text{days_in_year}}{\text{days_difference}}$$

где:

- **days_in_year** — число дней в году (зависит от выбранного значения аргумента **basis**);
- **days_difference** — количество дней от даты сделки **settlementdate** до даты погашения **maturitydate**.

Поскольку формула не учитывает никаких начислений на счет, эта функция является наиболее надежной на срок менее одного года.

	A	B	C	D	E	F
1						
2		Дата сделки	02.02.2009			
3		Дата погашения	03.12.2009			
4		Сумма уплаченная	1 000,00			
5		Сумма полученная	1 080,00			
6						
7				9,63%	=INTRATE(C2;C3;C4;C5;0)	
8						

Возвращается примерно 9,63%.

Calc и Excel не договорились об используемом числе дней в году при **basis** = 1. Неясно, которая теоретически является правильной. Calc использует количество дней в году, содержащего **purchasedate**.

IPMT

Функция **IPMT** возвращает сумму процентов, выплаченных за период по кредиту с фиксированной процентной ставкой в течение срока действия кредита.

Синтаксис функции:

=IPMT(rate; period; numperiods; principal; finalbalance; type)

где:

- **rate** — процентная ставка для каждого из периодов времени;
- **period** — период, за который должны быть рассчитаны проценты;
- **numperiods** — общее количество периодов выплаты в срок. Для стандартного 30-летнего кредита с ежемесячными платежами — это 360;
- **principal** — текущая стоимость (или начальная стоимость) кредита. Также известная как основная сумма кредита;
- **finalbalance** - остаток денежных средств, которого вы хотите достичь в конце срока выплат (необязательный — по умолчанию 0);
- **type** — когда осуществляются платежи:
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

В кредите с фиксированной ставкой, где можно делать постоянными платежи за каждый отчетный период на погашение кредита в течение срока, часть платежа каждого периода — проценты за непогашенную сумму кредита, а часть — погашение кредита. В течение долгого времени (пока Вы погашаете кредит), сумма процентов становится меньше, а погашение основной суммы кредита становится больше.

IPMT возвращает проценты, уплаченные в указанный период. **PPMT** возвращает сумму погашения кредита, уплачиваемую в этот же период. Вместе в целом они составляют периодическую выплату, задаваемую **PMT**.

	A	B	C	D	E	F
1						
2		Годовая процентная ставка	5,50%			
3		Период	12			
4		Срок кредита (лет)	2			
5		Сумма кредита	5000			
6						
7				-12,72	=IPMT(C2/12;C3;C4*12;C5;0;0)	
8						

Возвращается -12.72 в валютных единицах. Вы взяли кредит на 2 года в размере 5000 валютных единиц с ежегодной процентной ставкой 5,5%, делая ежемесячные платежи в конце месяца. В 12-ом месяце Вы делаете вашу обычную ежемесячную выплату, в которой 12.72 — проценты по кредиту.

Функция **IPMT** форматирует результат как денежный, если ячейка имеет форматирование по умолчанию. Она таким образом показывает реальную денежную сумму. Сумма, возвращенная **IPMT** может быть дробным — отображение округляет ее к самому близкому реальному денежному значению.

Заметим, что ваш кредитор может выполнять округление другим способом (например всегда вниз).

IRR

Функция **IRR** рассчитывает внутреннюю ставку доходности ряда потоков денежных средств, представленных их численными значениями. Внутренняя ставка доходности — это процентная ставка, принимаемая для инвестиции, состоящей из платежей (отрицательные величины) и доходов (положительные величины), которые осуществляются в последовательные и одинаковые по продолжительности периоды.

Синтаксис функции:

=IRR(payments; guess)

где:

- **payment** — диапазон, содержащий платежи, сделанные или полученные через регулярные промежутки времени;
- **guess** — (необязательный, по умолчанию 10%) — первое предположение о ставке.

IRR использует метод итераций для поиска уровня доходности, который дает нулевую чистую текущую стоимость для потоков наличности. По крайней мере один из потоков наличности должен быть отрицательным и по крайней мере один положительный - для того, чтобы чистая текущая стоимость была равна нулю. Уровень доходности за период, а проценты выплачиваются каждый период.

	A	B	C	D
1				
2		-5000		
3		1000		
4		2000		
5		3000		
6				
7		8,21%	=IRR(B2:B5)	
8				
9				

Эта функция в настоящее время не принимает массивы.

ISPMT

Функция **ISPMT** возвращает проценты, выплаченные за период по кредиту при фиксированной ставке.

Синтаксис функции:

=ISPMT(rate; period; numperiods; principal)

где:

- **rate** — процентная ставка за период;
- **period** — период, за который должны быть рассчитаны проценты;
- **numperiods** — общее количество периодов выплат в срок. Для стандартного 30-летнего кредита с ежемесячными платежами — это 360;
- **principal** — первоначальная сумма кредита.

ISPMT рассматривает фиксированную ставку по кредиту, когда Вы возмещаете одинаковую сумму капитала в начале каждого периода, а также уплачиваете проценты на непогашенный

остаток в конце каждого периода. Процент, который Вы платите, зависит от непогашенного остатка каждого периода, и уменьшается со временем. **ISPMT** возвращает этот процент.

Капитал, который Вы возмещаете в каждом периоде, определяется как **principal/numperiods**.

Это не то же самое, что **IPMT**, где постоянна общая сумма, которую Вы платите в каждый период (проценты плюс капитал).

	A	B	C	D	E	F	G
1							
2		Годовая процентная ставка		5,00%			
3		Срок кредита (лет)		4			
4		Сумма кредита		12000			
5							
6							
7		Период					
8		1	-450	=ISPMT(\$D\$2;B8;\$D\$3;\$D\$4)			
9		2	-300	=ISPMT(\$D\$2;B9;\$D\$3;\$D\$4)			
10		3	-150	=ISPMT(\$D\$2;B10;\$D\$3;\$D\$4)			
11		4	0	=ISPMT(\$D\$2;B11;\$D\$3;\$D\$4)			
12							

Вы берете кредит на 4 года в размере 12000 руб. с ежегодной процентной ставкой 5% и ежегодными выплатами. На следующий день вы возмещаете сумму кредита в размере $12000/4 = 3000$ руб., в результате чего непогашенный остаток по кредиту составляет $12000 - 3000 = 9000$ руб.

В строке 8 возвращается -450. В конце 1-ого периода проценты, которые Вы будете должны уплатить, составят $9000 * 5\% = 450$ руб. Знак минус, потому что Вы должны заплатить проценты по кредиту.

В строке 9 возвращается -300. В начале второго периода, Вы платите проценты в размере 450 руб., как указано выше, а также Вы возвращаете вторую часть суммы кредита в размере 3000 руб., оставляя непогашенный остаток по кредиту $9000 - 3000 = 6000$. В конце 2-го периода проценты, которые Вы будете должны уплатить, составят $6000 * 5\% = 300$ руб. Знак минус, потому что Вы должны заплатить проценты по кредиту.

В строке 10 возвращается -150. В начале третьего периода, Вы платите проценты в размере 300 руб., как указано выше, а также Вы возвращаете третью часть суммы кредита в размере 3000 руб., оставляя непогашенный остаток по кредиту $6000 - 3000 = 3000$. В конце 3-го периода проценты, которые Вы будете должны уплатить, составят $3000 * 5\% = 150$ руб. Знак минус, потому что Вы должны заплатить проценты по кредиту.

В строке 11 возвращается 0. В начале четвертого периода, Вы платите проценты в размере 150 руб, как указано выше, а также Вы возвращаете четвертую часть суммы кредита в размере 3000 руб, оставляя непогашенный остаток по кредиту $3000 - 3000 = 0$. Таким образом не требуется уплачивать проценты в конце четвертого периода.

ISPMT реализована в Excel, по-видимому, потому, что эта функция была доступна в Lotus 1-2-3. Совсем не понятно, работает ли таким образом какой-нибудь реальный кредит.

MDURATION

Функция **MDURATION** возвращает модифицированную продолжительность Макалея для ценных бумаг.

Эта функция только доступна, если установлен пакет *Анализа*.

Синтаксис функции:

=MDURATION(settlementdate; maturitydate; rate; yield; frequency; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **rate** — (ежегодная) процентная ставка по ценным бумагам;
- **yield** — (ежегодная) доходность по ценным бумагам (доход по ценным бумагам, выраженный в процентах к их цене);
- **frequency** — количество выплат процентов в год (1, 2 или 4);
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция возвращает *модифицированную* продолжительность Макалея для ценных бумаг, которая является мерой неустойчивости цены и чувствительности к процентным ставкам (зависимость цены для ценных бумаг от изменения процентных ставок). Она возвращает:

$$\text{MDURATION} = \frac{\text{Macaulay duration}}{\left(1 + \frac{\text{yield}}{\text{frequency}}\right)}$$

Продолжительность Макалея возвращается функцией **DURATION_ADD**.

	A	B	C	D	E	F
1						
2		Дата сделки	28.02.2008			
3		Дата погашения	31.08.2010			
4		Процентная ставка	5,00%			
5		Доходность	6,00%			
6		Количество выплат	2			
7						
8			2,26	=MDURATION(C2;C3;C4;C5;C6;0)		
9						

Функция возвращает приблизительно 2,26 года.

Эта функция использует в вычислении **YEARFRAC**. Есть известные небольшие погрешности (и в Calc, и в Excel) в **YEARFRAC**.

MIRR

Функция **MIRR** возвращает модифицированную внутреннюю норму доходности ряда потоков денежных средств. **MIRR** учитывает как затраты на привлечение инвестиции, так и процент, получаемый от реинвестирования денежных средств.

Синтаксис функции:

=MIRR(payments; financerate; reinvestrate)

где:

- **payment** — диапазон, содержащий платежи, сделанные или полученные через равные промежутки времени;

- **financerate** — процентная ставка, выплачиваемая за деньги, используемые в денежных потоках;
- **reinvestrate** — процентная ставка, получаемая на денежные потоки при их реинвестировании.

По крайней мере один из платежей должен быть отрицательными и по крайней мере один — положительный. Платежи, как предполагают, возникают в начале каждого периода; порядок, в котором задаются платежи, важен. Убедитесь, что значения выплат и поступлений введены в правильной последовательности и с правильными знаками (положительные значения для получаемых денег и отрицательные значения для выплачиваемых).

MIRR находит модифицированную внутреннюю норму доходности по формуле:

$$(1 + MIRR)^{n-1} = \frac{\text{финальное_значение_поступлений}}{\text{начальное_значение_выплат}}$$

	A	B	C	D	E	F
1						
2		Процентная ставка		5,00%		
3		Ставка реинвестирования		8,00%		
4						
5						
6		-5000		8,16%	=MIRR(B6:B9;D2;D3)	
7		1000				
8		2000				
9		3000				
10						

Эта функция в настоящее время не принимает массивы.

NOMINAL

Функция **NOMINAL** возвращает номинальную процентную ставку с учетом начисления эффективной процентной ставки.

Синтаксис функции:

=NOMINAL(eff_rate; num)

где:

- **eff_rate** — эффективная процентная ставка;
- **num** — количество раз, которое проценты начисляются в течение периода, к которому относится номинальная ставка.

Если инвестиции имеют номинальную ставку, скажем за год, но проценты выплачиваются и добавляются к инвестициям, скажем, каждый квартал, то проценты, выплаченные каждый квартал, самостоятельно начинают зарабатывать проценты. Это увеличивает эффективную стоимость. Эффективная ставка — ставка, которая должна быть выплачена в конце (скажем) года, чтобы дать тот же самый результат.

Учитывая эффективную ставку, эта функция возвращает соответствующую номинальную ставку.

Используется формула:

$$\text{eff_rate} = \left(1 + \frac{\text{nom_rate}}{\text{num}}\right)^{\text{num}} - 1$$

поэтому

$$nom_rate = num((1 + eff_rate)^{(1/num)} - 1)$$

	A	B	C	D	E
1					
2		Эффективная ставка	6,00%		
3		Число периодов	4		
4					
5			5,87%	=NOMINAL(C2;C3)	
6					

Возвращается приблизительно 5,87 %, что является номинальной ставкой для инвестиций с эффективной ставкой 6 % в год, начисляемой ежеквартально.

Расчет предполагает, что проценты добавляются к инвестициям в конце в точности равных периодов. В действительности различные кварталы года, например, имеют различное количество дней.

NOMINAL_ADD

Функция **NOMINAL_ADD** возвращает номинальную процентную ставку с учетом начисления эффективной процентной ставки.

Синтаксис функции:

=NOMINAL_ADD(eff_rate; num)

где:

- **eff_rate** — эффективная процентная ставка;
- **num** — количество раз, которое проценты начисляются в течение периода, к которому относится номинальная ставка.

Если инвестиции имеют номинальную ставку, скажем за год, но проценты выплачиваются и добавляются к инвестициям, скажем, каждый квартал, то проценты, выплаченные каждый квартал, самостоятельно начинают зарабатывать проценты. Это увеличивает эффективную стоимость. Эффективная ставка — ставка, которая должна быть выплачена в конце (скажем) года, чтобы дать тот же самый результат.

Учитывая эффективную ставку, эта функция возвращает соответствующую номинальную ставку.

Используется формула:

$$eff_rate = \left(1 + \frac{nom_rate}{num}\right)^{num} - 1$$

поэтому

$$nom_rate = num((1 + eff_rate)^{(1/num)} - 1)$$

	A	B	C	D	E
1					
2		Эффективная ставка	6,00%		
3		Число периодов	4		
4					
5			0,0587	=NOMINAL_ADD(C2;C3)	
6					

Возвращается приблизительно 0.0587, что является номинальной ставкой для инвестиций с эффективной ставкой 6 % в год, начисляемой ежеквартально.

Расчет предполагает, что проценты добавляются к инвестициям в конце в точности равных периодов. В действительности различные кварталы года, например, имеют различное

количество дней.

Функции **NOMINAL** и **NOMINAL_ADD** вычисляют одинаковый результат, и имеют одинаковое количество параметров. В отличие от функции **NOMINAL**, которая отображает возвращаемое значение в процентах, функция **NOMINAL_ADD** отображает возвращаемое значение как число. Согласно разрабатываемой версии стандарта ODFE, обе эти функции должны быть заменены на функцию **NOMINAL**.

NPER

Функция **NPER** возвращает количество периодов оплаты для для инвестиции с постоянными выплатами и постоянной процентной ставкой.

Синтаксис функции:

=NPER(rate; payment; presentvalue; futurevalue; type)

где:

- **rate** — (фиксированная) процентная ставка за период;
- **payment** — сумма выплаты, выполняемой каждый отчетный период. Это значение не может меняться в течение всего периода выплат. Обычно платеж состоит из основного платежа и платежа по процентам и не включает налогов и сборов;
- **presentvalue** — единовременная сумма платежа в начале срока;
- **futurevalue** — остаток денежных средств, выплачиваемый в конце срока (необязательный — по умолчанию 0).
- **type** — когда осуществляются платежи (необязательный — по умолчанию 0):
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

NPER возвращает количество периодов выплаты подразумевающих единовременно выплачиваемую сумму (**presentvalue**) в начале срока, оплату, выполняемую каждый период в течение **numperiods** периодов, по фиксированной процентной ставке, составляющую каждый период, а также единовременно выплачиваемую сумму (**futurevalue**) в конце срока.

	A	B	C	D	E
1					
2		Процентная ставка	5,00%		
3		Сумма периодических выплат	-100		
4		Сумма платежа в начале срока	0		
5		Сумма платежа в конце срока	1000		
6					
7		Количество периодов	8,31	=NPER(C2;C3;C4;C5;0)	
8					

Возвращенное число периодов может быть дробным. Это результат решения уравнения, и само по себе не является значимым. Реальное количество периодов будет целым.

Будьте осторожны, понимайте, как эта функция начисляет проценты каждого периода. Многие финансовые калькуляторы позволяют Вам устанавливать отдельный период наращивания — электронные таблицы не позволяют этого. Выберите соответствующую процентную ставку.

NPV

Функция **NPV** возвращает чистую приведенную стоимость инвестиций с регулярными

денежными выплатами.

Синтаксис функции:

=NPV(discountrate; payment1; payment2; ... payment30)

где:

- **discountrate** — ставка дисконтирования (выраженная в виде доли от 1), которая, по вашему мнению, относится к одному периоду. Она, как предполагается, является постоянной в течение всех периодов;
- **payment1** до **payment30** — до 30 чисел или диапазонов, содержащих цифры, представляющие платежи, произведенные сделанные в конце каждого ряда периодов фиксированной продолжительности. Платежи могут быть как положительными, так и отрицательными, для поступлений и выплат.

NPV вычисляет чистую текущую стоимость с использованием следующей формулы:

$$\sum_{i=1}^n \frac{payment_i}{(1 + discountrate)^i}$$

	A	B	C	D	E	F
1						
2		Ставка дисконтирования		8,75%		
3						
4				1 000,00 руб.		
5				2 000,00 руб.		
6				3 000,00 руб.		
7						
8				4 943,21 руб.	=NPV(D2;D4:D6)	
9						

Когда учетная ставка 8.75%, является конкурентоспособным возвращением через один год, и требуется заплатить 1000 в конце 1-го года, 2000 в конце 2-го года и 3000 в конце 3-го года, возвращается 4943.21 как денежная величина.

Выбранная учетная ставка, как предполагают, составляет каждый период. Если период — скажем, один месяц, а Вы желаете вычислять используя конкурентоспособные возвраты, известные через один год, Вы можете выбрать ставку дисконтирования равную одной двенадцатой конкурентоспособного возвращения — но знайте, что это не является абсолютно точным. Простой формулы не существует.

ODDFPRICE

Функция **ODDFPRICE** возвращает стоимость ценных бумаг за 100 валютных единиц номинальной стоимости, где время до первой выплаты процентов не целый период.

=ODDFPRICE(settlementdate; maturitydate; issuedate; firstinterestdate; rate; yield; redemptionvalue; frequency; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **issuedate** — дата выпуска (эмиссии) ценных бумаг;
- **firstinterestdate** — дата, на которую приходится первая выплата процентов по ценным бумагам;
- **rate** — процентная ставка по ценным бумагам;

- **yield** — (ежегодная) доходность по ценным бумагам (доход по ценным бумагам, выраженный в процентах к их цене);
- **redemptionvalue** — цена досрочного погашения ценных бумаг за 100 валютных единиц номинальной стоимости.
- **frequency** — количество выплат процентов в год (1, 2 или 4);
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция в настоящее время возвращает ошибочные результаты.

ODDFYIELD

Функция **ODDFYIELD** возвращает доходность ценных бумаг, когда время до первой выплаты процентов не целый период.

Синтаксис функции:

=ODDFYIELD(settlementdate; maturitydate; issuedate; firstinterestdate; rate; price; redemptionvalue; frequency; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **issuedate** — дата выпуска (эмиссии) ценных бумаг;
- **firstinterestdate** — дата, на которую приходится первая выплата процентов по ценным бумагам;
- **rate** — процентная ставка по ценным бумагам;
- **price** — курс ценных бумаг;
- **redemptionvalue** — цена досрочного погашения ценных бумаг за 100 валютных единиц номинальной стоимости.
- **frequency** — количество выплат процентов в год (1, 2 или 4);
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция в настоящее время возвращает ошибочные результаты.

ODDLPRICE

Функция **ODDLPRICE** возвращает стоимость ценных бумаг за 100 валютных единиц номинальной стоимости, если последняя выплата процентов не за целый период.

Синтаксис функции:

=ODDLPRICE(settlementdate; maturitydate; lastinterestdate; rate; yield; redemptionvalue; frequency; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **lastinterestdate** — дата, на которую приходится последняя выплата процентов по ценным бумагам;
- **rate** — процентная ставка по ценным бумагам;
- **yield** — (ежегодная) доходность по ценным бумагам (доход по ценным бумагам, выраженный в процентах к их цене);
- **redemptionvalue** — цена досрочного погашения ценных бумаг за 100 валютных единиц номинальной стоимости.
- **frequency** — количество выплат процентов в год (1, 2 или 4);
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция в настоящее время возвращает несколько иные результаты нежели Excel.

	A	B	C	D	E	F
1						
2		Дата покупки	01.06.90			
3		Дата погашения	31.12.95			
4		Дата последней выплаты	01.01.90			
5		Процентная ставка	2,00%			
6		Доходность	1,50%			
7		Цена досрочного погашения	100			
8		Количество выплат	4			
9						
10				102,512	=ODDLPRICE(C2;C3;C4;C5;C6;C7;C8;1)	
11						

Calc возвращает 102,512087533821, что очень немного неточно.

ODDLYIELD

Функция **ODDLYIELD** возвращает доходность по ценным бумагам, когда последняя выплата процентов не за целый период.

Синтаксис функции:

=ODDLYIELD(settlementdate; maturitydate; lastinterestdate; rate; price; redemptionvalue; frequency; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **lastinterestdate** — дата, на которую приходится последняя выплата процентов по ценным бумагам;
- **rate** — процентная ставка по ценным бумагам;
- **price** — курс ценных бумаг;
- **redemptionvalue** — цена досрочного погашения ценных бумаг за 100 валютных единиц номинальной стоимости.
- **frequency** — количество выплат процентов в год (1, 2 или 4);
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция в настоящее время возвращает несколько иные результаты нежели Excel.

	A	B	C	D	E	F
1						
2		Дата покупки	01.06.90			
3		Дата погашения	31.12.95			
4		Дата последней выплаты	01.01.90			
5		Процентная ставка	2,00%			
6		Курс	103,00			
7		Цена досрочного погашения	100,00			
8		Количество выплат	4			
9						
10				0,014	=ODDLYIELD(C2;C3;C4;C5;C6;C7;C8;1)	
11						

Calc возвращает 0,014087886014, что очень немного неточно.

PMT

Функция **PMT** возвращает сумму платежа за период на основе постоянства сумм платежей и постоянства процентной ставки.

Синтаксис функции:

=PMT(rate; numperiods; principal; finalbalance; type)

где:

- **rate** — (фиксированная) процентная ставка за период;
- **numperiods** — общее количество периодов выплат в срок. Для стандартного 30-летнего кредита с ежемесячными платежами — 360;

- **principal** — первоначальная сумма кредита, называемая также основной суммой;
- **finalbalance** - остаток денежных средств, которого вы хотите достичь в конце срока выплат (необязательный — по умолчанию 0);
- **type** — когда осуществляются платежи (необязательный — по умолчанию 0):
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

	A	B	C	D	E	F
1						
2		Годовая процентная ставка	5,50%			
3		Срок кредита (лет)	2			
4		Сумма кредита	5 000,00 руб.			
5						
6				-220,48 руб.	=PMT(C2/12;C3*12;C4;0;0)	
7						

Вы берете в кредит сроком на 2 года в сумме 5 000 руб. с ежегодной процентной ставкой 5,5%, делая ежемесячные платежи в конце месяца. Вы платите 220.48 руб. ежемесячно; сумма отрицательная, потому что Вы ее платите.

	A	B	C	D	E	F
1						
2		Годовая процентная ставка	5,00%			
3		Срок вклада (лет)	2			
4		Сумма накопления	1 000,00 руб.			
5						
6				-39,54 руб.	=PMT(C2/12;C3*12;0;C4;1)	
7						

Вы хотите накопить 1 000 руб. в течение 2-х лет, делая ежемесячные платежи, начиная с сегодняшнего дня. Вы предполагаете, что ставка останется постоянной — 5%. Проценты начисляются ежемесячно. Если вы вкладываете 39,54 руб в месяц, сумма на конец 2-х лет будет равна 1000 руб.

	A	B	C	D	E	F
1						
2		Годовая процентная ставка	5,50%			
3		Срок вклада (лет)	2			
4		Сумма кредита	5 000,00 руб.			
5		Сумма накопления	1 000,00 руб.			
6						
7				-259,99 руб.	=PMT(C2/12;C3*12;C4;C5;0)	
8						

Вы берете кредит на сроком 2 года в сумме 5000 руб. с ежегодной процентной ставкой 5,5%, делая ежемесячные платежи в конце месяца. Вы хотите получить единовременную выплату в 1000 руб в конце срока кредита. Проценты начисляются ежемесячно.

Это гипотетический пример, который делает маловероятное предположение о том, что проценты, начисляемые на Ваш кредит имеют то же значение, которое вы получаете на вклад (при накоплении вашей единовременно выплачиваемой суммы 1000 руб.). Требуется объяснить знак каждого параметра: 5000 — положительное, потому что Вам выплачивают сумму кредита; 1000 — положительное, потому что Вам выплачивается единовременная сумма; 259,99 - отрицательное, потому что Вы платите ежемесячную сумму.

Выплаты, возвращаемые функцией **PMT**, включают основные платежи и платежи по процентам, но не включают налогов, резервных платежей или комиссий, иногда связываемых с кредитом.

Для кредита с ежемесячными выплатами, чтобы получить месячную процентную ставку, разделите годовую ставку на 12. Чтобы узнать количество выплат, умножьте количество лет кредита на 12.

Функция **PMT** форматирует результат как денежный, если ячейка имеет форматирование по умолчанию. Она таким образом показывает реальную денежную сумму. Сумма, возвращенная **PMT** может быть дробным — отображение округляет ее к самому близкому реальному денежному значению. Заметим, что ваш кредитор может выполнять округление другим способом (например всегда вниз).

С небольшими значениями **rate**, Calc (совместно с Excel) может приводить к ошибке. Это не имеет последствий в расчетах реального мира.

PPMT

Функция **PPMT** возвращает величину платежа в погашение основной суммы кредита за период на основе постоянства периодических платежей и постоянства процентной ставки.

Синтаксис функции:

=PPMT(rate; period; numperiods; principal; finalbalance; type)

где:

- **rate** — (фиксированная) процентная ставка за период;
- **period** — период, за который должна быть рассчитана величину платежа в погашение основной суммы кредита;
- **numperiods** — общее количество периодов выплат в срок. Для стандартного 30-летнего кредита с ежемесячными платежами — 360;
- **principal** — первоначальная сумма кредита, называемая также основной суммой;
- **finalbalance** - остаток денежных средств, которого вы хотите достичь в конце срока выплат (необязательный — по умолчанию 0);
- **type** — когда осуществляются платежи (необязательный — по умолчанию 0):
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

В кредите с фиксированной ставкой, где можно делать постоянными платежи за каждый отчетный период на погашение кредита в течение срока, часть платежа каждого периода — проценты за непогашенную сумму кредита, а часть — погашение кредита. В течение долгого времени (пока Вы погашаете кредит), сумма процентов становится меньше, а погашение основной суммы кредита становится больше.

IPMT возвращает проценты, уплаченные в указанный период. **PPMT** возвращает сумму погашения кредита, уплачиваемую в этот же период. Вместе в целом они составляют периодическую выплату, задаваемую **PMT**.

	A	B	C	D	E	F
1						
2		Годовая процентная ставка	5,50%			
3		Период	12			
4		Срок кредита (лет)	2			
5		Сумма кредита	5 000,00 руб.			
6						
7				-207,75 руб.	=PPMT(C2/12;C3;C4*12;C5;0;0)	
8						

Вы берете в кредит сроком на 2 года в сумме 5 000 руб. с ежегодной процентной ставкой

5,5%, делая ежемесячные платежи в конце месяца. В 12-ом месяце Вы сделали ваш обычный ежемесячный платеж, в котором 207,75 руб — погашение основной суммы кредита.

Функция **PPMT** форматирует результат как денежный, если ячейка имеет форматирование по умолчанию. Она таким образом показывает реальную денежную сумму. Сумма, возвращенная **PPMT** может быть дробным — отображение округляет ее к самому близкому реальному денежному значению. Заметим, что ваш кредитор может выполнять округление другим способом (например всегда вниз).

PRICE

Функция **PRICE** вычисляет котировальную цену для выплаты процентов по ценным бумагам, за 100 валютных единиц номинальной стоимости.

Синтаксис функции:

=PRICE(settlementdate; maturitydate; rate; yield; redemptionvalue; frequency; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **rate** — процентная ставка по ценным бумагам;
- **yield** — необходимый ежегодный уровень доходности (доход по ценным бумагам, выраженный в процентах к их цене);
- **redemptionvalue** — цена досрочного погашения ценных бумаг за 100 валютных единиц номинальной стоимости;
- **frequency** — количество выплат процентов в год (1, 2 или 4);
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция вычисляет котировальную цену для ценных бумаг (“чистая” цена). Цена, фактически заплаченная (“грязная” цена) — больше, потому что она включает начисленные проценты.

Если

- A — число дней от **settlementdate** до следующей до даты следующего купона,
- B — количество дней в периоде купона, на который приходится **settlementdate**,
- C — количество оплачиваемых купонов между **settlementdate** и **maturitydate**,
- D — количество дней от начала периода купона до **settlementdate**,

то **PRICE** вычисляется как:

$$PRICE = \frac{redemptionvalue}{\left(1 + \frac{yield}{frequency}\right)^{C-1+\frac{A}{B}}} + \sum_{k=1}^C \frac{\frac{100 * rate}{frequency}}{\left(1 + \frac{yield}{frequency}\right)^{k-1+\frac{A}{B}}} - 100 * \frac{rate}{frequency} * \frac{D}{B}$$

	A	B	C	D	E	F
1						
2		Дата покупки	15.02.08			
3		Дата погашения	15.11.10			
4		Процентная ставка	5,00%			
5		Доходность	7,00%			
6		Цена досрочного погашения	100,00			
7		Количество выплат	2			
8						
9			95,064	=PRICE(C2;C3;C4;C5;C6;C7;0)		
10						

Возвращается приблизительно 95,064.

Эта функция вычисляет текущая стоимости, начисляя необходимую ставку доходности каждый период купона.

Цена рассчитывается по состоянию на дату сделки (когда деньги переходят к другому владельцу). Контракт на покупку облигаций (дата заключения сделки) может предшествовать этому (например, на 3 дня).

PRICEDISC

Функция **PRICEDISC** вычисляет цену для дисконтных облигаций без выплаты процентов.

Синтаксис функции:

=PRICEDISC(settlementdate; maturitydate; discountrate; redemptionvalue; basis)

где:

- **settlementdate** — дата сделки (покупки) по облигациям;
- **maturitydate** — дата погашения (выкупа) облигаций;
- **discountrate** — (ежегодная) ставка дисконта облигаций;
- **redemptionvalue** — цена досрочного погашения облигаций за 100 валютных единиц номинальной стоимости;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция вычисляет цену (или стоимость) за 100 валютных единиц номинальной стоимости, для облигаций, по которым не выплачивается никаких процентов (“чистый дисконтный инструмент” или “дисконтная облигация с нулевым купоном”).

PRICEDISC возвращает:

$$redemptionvalue - (redemptionvalue * discountrate * \frac{days_to_maturity}{days_in_year})$$

где:

- **days_to_maturity** — количество дней от даты сделки до даты погашения;

- *days_in_year* - количество дней в году, зависит от значения **basis**.

	A	B	C	D	E	F
1						
2		Дата покупки	15.02.08			
3		Дата погашения	30.11.08			
4		Ставка дисконта	3,00%			
5		Цена досрочного погашения	100,00			
6						
7			97,631	=PRICEDISC(C2;C3;C4;C5;1)		
8						

Вы покупаете и оплачиваете облигации 15 февраля 2008; облигации подлежат оплате 30 ноября 2008, приносящие номинальную стоимость 1000 руб. Выбранная учетная ставка 3 % в год, стоимость облигаций — 976,31 руб., при использовании системы дат с **basis** = 1.

Есть (редкие) случаи, когда результаты в Calc и Excel отличаются друг от друга. Данная формула включает “*days_in_year*”, который не достаточно точно определяется при **basis** = 1, когда диапазон лет может включать високосные годы.

Использованная формула не учитывает начисляемые сложные проценты. Если вы используете эту функцию для расчета стоимости облигаций, и время до погашения более одного года, необходимо выбрать дисконтную ставку с надлежащей осторожностью.

Цена рассчитывается по состоянию на дату сделки (когда деньги переходят к другому владельцу). Контракт на покупку облигаций (дата заключения сделки) может предшествовать этому (например, на 3 дня).

PRICEMAT

Функция **PRICEMAT** вычисляет цену (в расчете на 100 денежных единиц номинальной стоимости) облигаций, по которым выплачиваются проценты при погашении.

Синтаксис функции:

=PRICEMAT(settlementdate; maturitydate; issuedate; rate; yield; basis)

где:

- **settlementdate** — дата сделки (покупки) по облигациям;
- **maturitydate** — дата погашения (выкупа) облигаций;
- **issuedate** — дата выпуска (эмиссии) облигаций;
- **rate** — (ежегодная) процентная ставка по облигациям;
- **yield** — (ежегодная) доходность облигаций;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция вычисляет цену (или ценность), в расчете на 100 денежных единиц номинальной стоимости, для облигаций, по которым проценты выплачиваются только однажды, при погашении.

	A	B	C	D	E	F
1						
2		Дата покупки	15.02.2007			
3		Дата погашения	06.04.2007			
4		Дата выпуска	06.01.2007			
5		Процентная ставка	5,00%			
6		Доходность	6,00%			
7						
8				99,85	=PRICEMAT(C2;C3;C4;C5;C6;0)	
9						

Вы покупаете и оплачиваете облигации 15 февраля 2007; (90 дневные) облигации были выпущены 6 января 2007 и погашаются 6 апреля 2007, принося свою номинальную стоимость 1000 руб. и выплачивая проценты $1000 \text{ руб} * 5 \% * 90/360$. Выбирая доходность в 6% годовых, стоимость облигации на 15 февраля 2007 года составляет 998,50 руб., при использовании системы дат с **basis** = 0.

Бывают (редкие) ситуации, когда результаты в Calc и Excel отличаются.

Используемая формула не учитывает начисляемых процентов. Если вы используете эту функцию для расчета стоимости облигаций, и время от выпуска до погашения более одного года, необходимо выбирать требуемую доходность с определенной осторожностью.

Цена рассчитывается по состоянию на дату сделки (когда деньги переходят к другому владельцу). Контракт на покупку облигаций (дата заключения сделки) может предшествовать этому (например, на 3 дня).

PV

Функция **PV** возвращает текущую стоимость потока будущих платежей с заключительной единовременно выплачиваемой суммой.

Синтаксис функции:

=PV(rate; numperiods; payment; futurevalue; type)

где:

- **rate** — (фиксированная) процентная ставка за период;
- **numperiods** — общее количество периодов выплат в срок;
- **payment** — сумма платежа, выполняемого каждый отчетный период. Это значение не может меняться в течение всего периода платежей. Обычно **payment** состоит из основного платежа и платежа по процентам, но не включает других налогов и сборов. Если приводится **futurevalue**, этот аргумент может быть опущен (по умолчанию 0);
- **futurevalue** - остаток денежных средств, которого вы хотите достичь в конце срока выплат (необязательный — по умолчанию 0). Для кредита — обычно 0; для облигаций — цена погашения;
- **type** — когда осуществляются платежи (необязательный — по умолчанию 0):
 - 0 — в конце каждого периода.
 - 1 — в начале каждого периода (включая выплату в начале срока).

Стоимость денег зависит от времени; например, 100 \$ сегодня стоили бы 110 \$ через год, если инвестиции выполняются с процентной ставкой — 10%.

PV возвращает сегодняшнюю стоимость, оплаты, выполняемой в каждый период в течение **numperiods** периодов, с дополнительной единовременной выплатой (**futurevalue**) в конце срока, с фиксированной процентной ставкой, выплачиваемых каждый период.

Примеры далее поясняют, каким образом функция может быть использована.

	A	B	C	D	E
1					
2		Годовая процентная ставка	5,00%		
3		Количество периодов	15		
4		Сумма ежегодной выплаты	1 000,00 руб.		
5					
6				-10 379,66 руб.	=PV(C2;C3;C4;0;0)
7					

Вы имеете возможность купить ренту, которая выплачивала бы Вам 1000 в конце каждого года в течение 15 лет. Вы принимаете постоянную годовую процентную ставку 5 %. Исходя из этого, рента сегодня стоит 10 379.66; если она стоит больше, Вы не захотите ее купить. Результат отрицателен, потому что Вы должны заплатить за ренту.

	A	B	C	D	E
1					
2		Годовая процентная ставка	5,00%		
3		Срок кредита (лет)	3		
4		Сумма ежемесячных платежей	-100,00 руб.		
5					
6				3 336,57 руб.	=PV(C2/12;C3*12;C4;0;0)
7					

Вы хотите взять кредит в размере 3 500 руб., который означает, что вы возвращаете сумму 100 руб. в конце каждого месяца в течение 3 лет. Вы принимаете постоянную процентную ставку 5 %. На этом основании, кредит стоит только 3 336.57 руб., поэтому это плохая сделка. Ежемесячная оплата отрицательная, потому что Вы ее платите, а результат положительный, потому что кредит платится Вам.

	A	B	C	D	E
1					
2		Годовая процентная ставка	7,00%		
3		Срок кредита (лет)	3		
4		Сумма ежемесячных выплат	100,00 руб.		
5		Сумма выплаты по окончании срока	1 000,00 руб.		
6					
7				-1 346,36 руб.	=PV(C2/2;C3*2;C4;C5;0)
8					

Выпускаются 3-х летние облигации, с выплатами 100 руб. по полугодовым купонам и 1000 руб. по завершении срока. Вам нужна годовая ставка доходности от 7% (выплачивается раз в полгода) принимая во внимание любые риски, следовательно, Вы оцениваете выпуск облигаций в 1346,36 руб. Результат показывается отрицательным, так как Вы должны заплатить за облигации.

Убедитесь, что вы последовательны в выборе единиц измерения для задания аргументов **rate** и **numperiods**. Если вы делаете ежемесячные выплаты по трехгодичному займу из расчета 5 процентов годовых, то используйте 5%/12 для задания аргумента **rate** и 3*12 для задания аргумента **numperiods**. Если Вы делаете ежегодные платежи по тому же займу, то используйте 5% для задания аргумента **rate** и 3 для задания аргумента **numperiods**.

В функциях, связанных с аннуитетами, выплачиваемые денежные средства, такие как депозит на сбережения, представляются отрицательным числом; полученные денежные средства, такие как чеки на дивиденды, представляются положительным числом.

Будьте осторожны, понимайте, как эта функция начисляет проценты каждого периода. Многие финансовые калькуляторы позволяют Вам устанавливать отдельный период наращения — электронные таблицы не позволяют этого. Выберите соответствующую

процентную ставку.

RATE

Функция **RATE** вычисляет процентную ставку по аннуитету за один период.

Синтаксис функции:

=RATE(numperiods; payment; presentvalue; futurevalue; type; guess)

где:

- **numperiods** — общее количество периодов выплат в срок;
- **payment** — сумма платежа, выполняемого каждый отчетный период, величина которого остается постоянной в течение всего срока аннуитета. Если приводится **futurevalue**, этот аргумент может быть опущен (по умолчанию 0);
- **presentvalue** — выплата единовременной суммы в начале срока;
- **futurevalue** — остаток денежных средств, которого вы хотите достичь в конце срока выплат (необязательный — по умолчанию 0);
- **type** — когда осуществляются платежи (необязательный — по умолчанию 0):
 - 0 — в конце каждого периода;
 - 1 — в начале каждого периода (включая выплату в начале срока).
- **guess** — необязательное первое предположение о ставке (по умолчанию — 10%).

RATE вычисляет требуемую фиксированную процентную ставку, так, чтобы **presentvalue** (приведенная к текущему моменту стоимость) сопровождаемая выплатами каждый период в течение **numperiods** периодов привела к **futurevalue**, стоимость в конце срока. Проценты, как предполагают, начисляются каждый период. При расчете используется итерация для решения уравнения, с **guess** в качестве отправного значения для итерации.

Если эта функция возвращает ошибку **Err:523**, итерации не удалось сойтись — попробуйте разные **guess** или перепроверьте параметры.

	A	B	C	D	E
1					
2		Количество периодов		3	
3		Ежемесячный платеж	1 000,00 руб.		
4		Сумма в конце срока	3 152,00 руб.		
5					
6				4,98%	=RATE(C2;-C3;0;C4)
7					

Вы платите 1 000 руб. в конце каждого года в течение 3 лет, и получаете 3 152 руб. в конце срока. Примененная процентная ставка составляет 4.98 %.

Убедитесь, что вы последовательны в выборе единиц измерения для задания аргументов **guess** и **numperiods**. Если делаются ежемесячные выплаты по трехгодичному займу под 12 процентов годовых, используйте 10%/12 для задания аргумента **guess** и 3*12 для задания аргумента **numperiods**. Если делаются ежегодные платежи по тому же займу, то используйте 10% для задания аргумента **guess** и 3 для задания аргумента **numperiods**.

RECEIVED

Функция **RECEIVED** вычисляет сумму, получаемую при погашении для облигации с нулевым купоном.

Синтаксис функции:

=RECEIVED(settlementdate; maturitydate; purchasevalue; discountrate; basis)

где:

- **settlementdate** — дата сделки (покупки) по облигациям;
- **maturitydate** — дата погашения (выкупа) облигаций;
- **purchasevalue** — стоимость облигации при покупке;
- **discountrate** — учетная ставка по облигациям;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

RECEIVED вычисляет сумму, возвращаемую при погашении для дисконтной облигации, по которой проценты не выплачиваются. Она возвращает:

$$\frac{\text{purchasevalue}}{(1 - \text{days_difference} / \text{days_in_year})}$$

где:

- *days_difference* — число дней между **settlementdate** и **maturitydate**,
- *days_in_year* — число дней в году,

оба вычисляются согласно календарной системе, задаваемой параметром **basis**.

Поскольку формула не учитывает начисления сложных процентов, эта функция является наиболее надежной на срок менее одного года.

	A	B	C	D	E	F
1						
2		Дата покупки	01.06.2009			
3		Дата погашения	31.12.2009			
4		Стоимость при покупке	10 000,00			
5		Учетная ставка	5,00%			
6						
7				10 300,5503	=RECEIVED(C2;C3;C4;C5;1)	
8						

Calc и Excel не договорились о количестве дней в году, при **basis** = 1. Не ясно, которое является теоретически правильным. Calc использует на количество дней в году, содержащем **settlementdate**.

Excel, судя по всему, имеет ошибку для **basis** = 4.

RRI

Функция **RRI** возвращает эквивалентную процентную ставку, когда инвестиции увеличиваются в стоимости.

Синтаксис функции:

=RRI(numperiods; presentvalue; futurevalue)

где:

- **numperiods** — общее количество периодов выплат в сроке;
- **presentvalue** — стоимость в начале срока;
- **futurevalue** — стоимость в конце срока.

RRI вычисляет требуемую фиксированную процентную ставку, чтобы **presentvalue** инвестированные на **numperiods** (с начислением процентов каждый период) стоили **futurevalue** в конце срока. **RRI** возвращает фиксированную процентную ставку, которая применяется к каждому периоду.

Основное уравнение:

$$presentvalue * (1 + RRI)^{numperiods} = futurevalue.$$

	A	B	C	D	E
1					
2		Количество периодов		8	
3		Сумма в начале срока	1 000 000,00 руб.		
4		Сумма в конце срока	1 600 000,00 руб.		
5					
6			6,05%	=RRI(C2;C3;C4)	
7					

Вы инвестируете 1 000 000 руб, и получаете 1 600 000 руб. через 8 лет. Если бы Вы вместо этого вложили эту сумму в счет с фиксированной процентной ставкой, с ежегодным начислением процентов, то счету требовалось бы выплачивать 6.05 %, чтобы получить ту же самую сумму.

Количество периодов в сроке **numperiods** может быть введено как дробное число, но в этом случае результат может быть немного неточным, поскольку дробный период будет вычисляться, как если бы проценты были начислены частично.

SLN

Функция **SLN** возвращает сумму амортизации актива за один период, используя линейный метод амортизации.

Синтаксис функции:

=SLN(originalcost; salvagevalue; lifetime)

где:

- **originalcost** — первоначальная стоимость актива;
- **salvagevalue** — это стоимость актива в конце срока амортизации (иногда называется остаточной стоимостью актива);
- **lifetime** — число периодов (обычно лет), за которые обесценивается актив (иногда называется периодом амортизации).

Линейный метод амортизации начисляет одну и ту же сумму амортизации актива в каждом периоде; это значение возвращается функцией **SLN**. Формула:

$$сумма\ амортизации = \frac{(originalcost - salvagevalue)}{lifetime}$$

	A	B	C	D	E
1					
2		Первоначальная стоимость	14000		
3		Ликвидационная стоимость	4000		
4		Срок службы	5		
5					
6				2 000,00	=SLN(C2;C3;C4)
7					

Для более точного определения отчислений в определенном периоде эксплуатации при задании даты конца периода (когда покупка актива происходит в середине бухгалтерского периода) можно использовать функцию **AMORLINC**, использующую тот же метод начисления.

SYD

Функция **SYD** возвращает сумму амортизации актива за данный год, используя метод суммы годовых цифр (метод ускоренной амортизации, при котором норма амортизации подсчитывается путем деления числа периодов, оставшихся до истечения срока службы актива, на сумму периодов службы этого актива). Это позволяет списать на амортизацию уже в первые годы использования актива большую стоимость.

Сумма амортизации различается для отдельных лет: для первого года она максимальна, а для последнего – минимальна.

Синтаксис функции:

=SYD(originalcost; salvagevalue; lifetime; year)

где:

- **originalcost** — первоначальная стоимость актива;
- **salvagevalue** — это стоимость актива в конце срока амортизации (иногда называется остаточной стоимостью актива);
- **lifetime** — число лет, за которые обесценивается актив (иногда называется периодом амортизации);
- **year** — год, для которого вычисляется сумма амортизации.

Для расчета амортизации, **SYD** складывает номера лет в течение срока службы актива; номера лет, взятые в обратном порядке и разделенные на их сумму дают пропорцию полной потери стоимости для определения амортизации в том году.

Например, при 4-х годах амортизации, где актив теряет 5000 в стоимости: сумма номеров лет 1 + 2 + 3 + 4 = 10, и амортизация каждый год (по порядку) - 4/10, 3/10, 2/10 и далее 1/10 от 5000.

Формула для годовой амортизации

$$\text{сумма амортизации} = \frac{(\text{originalcost} - \text{salvagevalue}) * (\text{lifetime} + 1 - \text{year}) * 2}{((\text{lifetime} + 1) * \text{lifetime})}$$

	A	B	C	D	E
1					
2		Первоначальная стоимость	10000		
3		Ликвидационная стоимость	2000		
4		Срок службы	4		
5		Расчетный год	1		
6					
7				3 200,00	=SYD(C2;C3;C4;C5)
8					

Возвращается 3200 в денежных единицах, что составляет амортизацию в первый год для актива, стоимость которого 10000, а остаточная стоимость 2000 через 4 года. Полная потеря стоимости 8000; 3200 составляет 4/10 от этого.

TBILLEQ

Функция **TBILLEQ** возвращает эквивалентную доходность по облигациям (BEY) для векселя Казначейства США.

Синтаксис функции:

=TBILLEQ(settlementdate; maturitydate; discountrate)

где:

- **settlementdate** — дата сделки (покупки) по казначейскому векселю;
- **maturitydate** — дата погашения (выкупа) казначейского векселя;
- **discountrate** — учетная ставка по казначейскому векселю.

Казначейский вексель — краткосрочная (со сроком обращения до одного года) правительственная ценная бумага, проданная со скидкой к ее номиналу (номинальной стоимости). По ней не начисляются проценты, и она выкупается по номинальной стоимости.

Эта функция рассчитывает доходность, которая требуется облигации, чтобы обеспечить рост, эквивалентный казначейскому векселю. При рассмотрении облигации принимается 365 дней в году, а проценты выплачиваются только в конце срока (то есть проценты не начисляются).

Казначейский вексель предполагает 360 дней в году.

Формула для **TBILLEQ**:

$$\frac{365 * \text{discountrate}}{(360 - \text{discountrate} * \text{number_of_days_in_the_term})}$$

где *number_of_days_in_the_term* — фактическое число дней между **settlementdate** и **maturitydate**.

Ошибочный результат, если заданный срок — не меньше чем один год.

	A	B	C	D	E
1					
2		Дата покупки	14.07.2008		
3		Дата погашения	14.01.2009		
4		Учетная ставка	4,00%		
5					
6				0,0414	=TBILLEQ(C2;C3;C4)
7					

Возвращается приблизительно 0,0414 или 4,14 %.

Эта функция может помочь с векселями казначейства США если используется аккуратно. Основная формула не может применяться к казначейским векселям, выданным другими органами власти.

TBILLPRICE

Функция **TBILLPRICE** возвращает эмиссионную цену (цена, по которой размещаются ценные бумаги нового выпуска) для векселя Казначейства США, за \$ 100 номинальной стоимости, учитывая учетную ставку.

Синтаксис функции:

=TBILLPRICE(settlementdate; maturitydate; discountrate)

где:

- **settlementdate** — дата сделки (покупки) по казначейскому векселю;
- **maturitydate** — дата погашения (выкупа) казначейского векселя;
- **discountrate** — учетная ставка по казначейскому векселю.

Казначейский вексель — краткосрочная (со сроком обращения до одного года) правительственная ценная бумага, проданная со скидкой к ее номиналу (номинальной стоимости). По ней не начисляются проценты, и она выкупается по номинальной стоимости.

Казначейский вексель предполагает 360 дней в году.

Формула для **TBILLPRICE**:

$$100 * (1 - (\text{discountrate} * \text{number_of_days_in_the_term} / 360))$$

где *number_of_days_in_the_term* — фактическое число дней между **settlementdate** и **maturitydate**.

Ошибочный результат, если заданный срок — не меньше чем один год.

	A	B	C	D	E
1					
2		Дата покупки	14.07.2008		
3		Дата погашения	14.01.2009		
4		Учетная ставка	4,00%		
5					
6				97,99	=TBILLPRICE(C2;C3;C4)
7					

Возвращается приблизительно 97,99.

Эта функция может помочь с векселями казначейства США если используется аккуратно. Основная формула не может применяться к казначейским векселям, выданным другими органами власти.

TBILLYIELD

Возвращает доходность для векселя Казначейства США.

Синтаксис функции:

=TBILLYIELD(settlementdate; maturitydate; issueprice)

где:

- **settlementdate** — дата сделки (покупки) по казначейскому векселю;
- **maturitydate** — дата погашения (выкупа) казначейского векселя;
- **issueprice** — эмиссионная цена Казначейского векселя за \$100 номинальной стоимости.

Казначейский вексель — краткосрочная (со сроком обращения до одного года) правительственная ценная бумага, проданная со скидкой к ее номиналу (номинальной стоимости). По ней не начисляются проценты, и она выкупается по номинальной стоимости.

Казначейский вексель предполагает 360 дней в году.

Доходность рассчитывается как:

$$\frac{(100 - \text{issueprice})}{\text{issueprice}} * \frac{360}{\text{number_of_days_in_the_term}}$$

где *number_of_days_in_the_term* — фактическое число дней между **settlementdate** и **maturitydate**.

Ошибочный результат, если заданный срок — не меньше чем один год.

	A	B	C	D	E
1					
2		Дата покупки	14.07.2008		
3		Дата погашения	14.01.2009		
4		Эмиссионная цена	96,00		
5					
6			0,0829	=TBILLYIELD(C2;C3;C4)	
7					

Возвращается приблизительно 0,0829 или 8,29 %.

Эта функция может помочь с векселями казначейства США если используется аккуратно. Основная формула не может применяться к казначейским векселям, выданным другими органами власти.

VDB

Возвращает сумму амортизации актива за данный год, используя метод переменного уменьшающегося остатка.

Синтаксис функции:

=VDB(originalcost; salvagevalue; lifetime; periodstart; periodend; factor; nostraightline)

где:

- **originalcost** — первоначальная стоимость актива;
- **salvagevalue** — это стоимость актива в конце срока амортизации (иногда называется остаточной стоимостью актива);
- **lifetime** — число лет, за которые обесценивается актив (иногда называется периодом амортизации);
- **periodstart, periodend** — номера начального и конечного годов, определяющих период, за который рассчитывается амортизация. Они могут быть дробными, от 0 до **lifetime**.
- **factor** — коэффициент для задания нормы амортизации (2 если опущен);
- **nostraightline** — логическое значение, определяющее, следует ли использовать линейную амортизацию в том случае, когда амортизация превышает величину, рассчитанную методом снижающегося остатка. Если **TRUE** то переключения на метод начисления линейной амортизации не происходит, даже если амортизация больше величины, вычисленной методом снижающегося остатка. Если **FALSE** или опущен, то происходит переключение на метод начисления линейной амортизации, если амортизация больше величины, рассчитанной методом снижающегося остатка.

Для расчета амортизации, **VDB** использует фиксированную норму амортизации. Когда **factor** = 2 — это метод двойного уменьшения остатка (потому что он удваивает прямолинейную норму, которая обесценивала бы актив к нулю). Норма амортизации определяется по следующей формуле:

$$rate = \frac{factor}{lifetime}$$

Амортизации каждый год рассчитывается в конце года как:

$$book_value_at_start_of_year * rate$$

Однако если прямолинейная амортизации до **salvagevalue** будет больше, чем рассчитанная по вышеприведенной формуле, используется она. Прямолинейная амортизации до **salvagevalue** за год, определяется по формуле:

$$(book_value_at_start_of_year - salvagevalue) / years_of_life_left$$

Для предотвращения снижения балансовой стоимости ниже **salvagevalue**, амортизация в любом году ограничивается значением:

$$book_value_at_start_of_year - salvagevalue$$

Это определяет балансовую стоимость на конец каждого года. VDB возвращает сумму амортизации за период с **periodstart** до **periodend**. 0 соответствует началу срока службы актива, а **lifetime** окончанию. Таким образом, период от 0 до 2 возвращает сумму амортизации за первые два года эксплуатации актива. Амортизации рассчитывается как балансовая стоимость на **periodstart** минус балансовая стоимость на **periodend**.

Дробни допускаются: период 1,25 до 3,5 возвращает сумму амортизации с конца первого квартала второго года до конца второго квартала четвертого года. Балансовая стоимость за дробный период находится линейно интерполирования на конец года балансовая стоимость.

	A	B	C	D	E	F	G
1							
2		Первоначальная стоимость		100 000,00 руб.			
3		Ликвидационная стоимость		5 000,00 руб.			
4		Срок службы		10			
5							
6							
7		Начальный период	Конечный период	Сумма амортизации			
8		0	1	20 000,00 руб.	=VDB(\$D\$2;\$D\$3;\$D\$4;B8;C8)		
9		0	2	36 000,00 руб.	=VDB(\$D\$2;\$D\$3;\$D\$4;B9;C9)		
10		1,5	2,5	14 400,00 руб.	=VDB(\$D\$2;\$D\$3;\$D\$4;B10;C10)		
11		7	8	5 303,60 руб.	=VDB(\$D\$2;\$D\$3;\$D\$4;B11;C11)		
12		9	10	5 303,60 руб.	=VDB(\$D\$2;\$D\$3;\$D\$4;B12;C12)		
13		0	10	89 262,58 руб.	=VDB(\$D\$2;\$D\$3;\$D\$4;B13;C13;2;1)		
14							

В строке 8 возвращается 20 000 руб., амортизация за первый год — то есть $100000 * 2 / 10$ (амортизация методом двойного уменьшения остатка).

В строке 9 возвращается 36000 руб., амортизация в первые два года — то есть 20 000 в первый год плюс $80000 * 2 / 10$ (амортизация методом двойного уменьшения остатка).

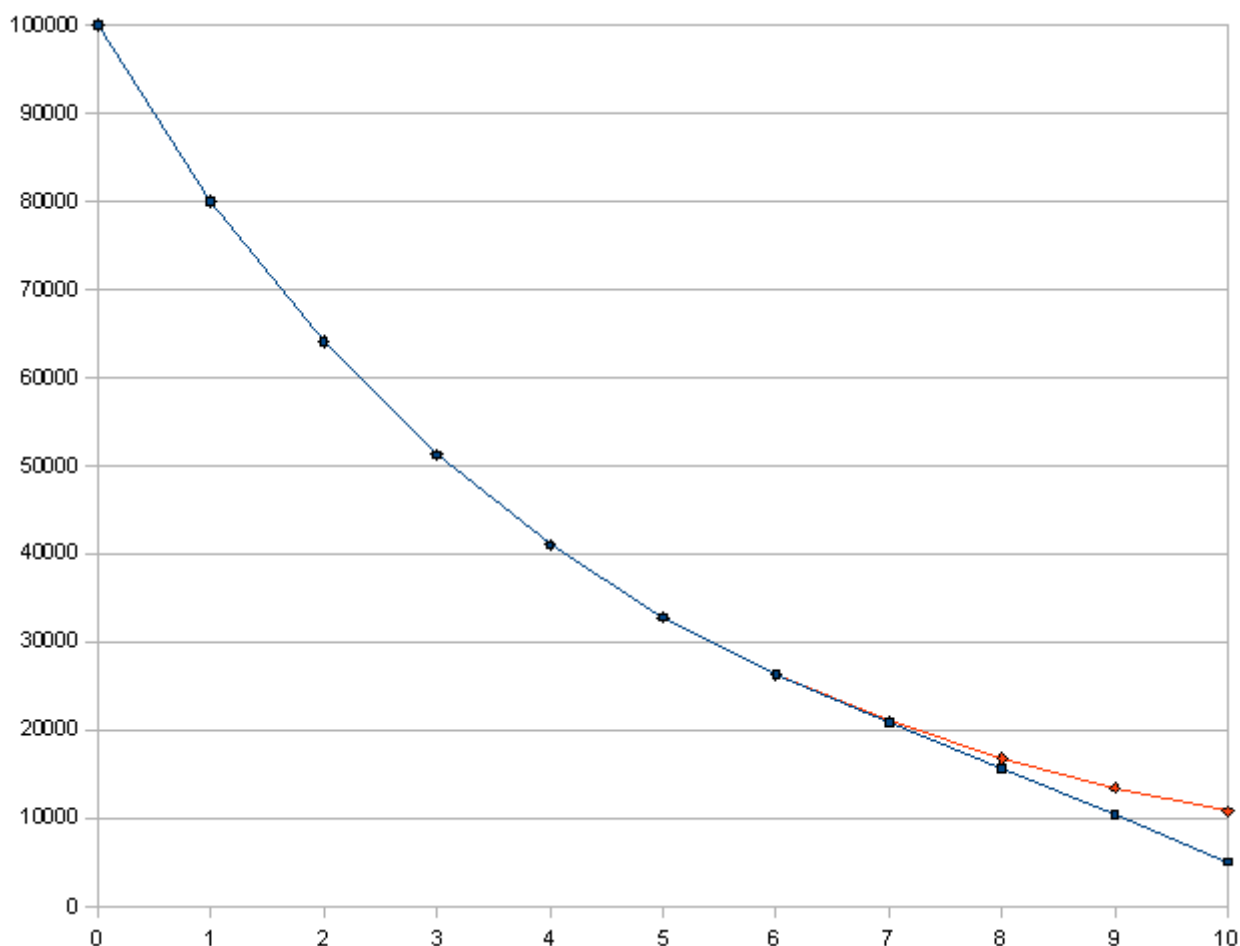
В строке 10 возвращается 14400 руб., амортизацию с середины второго года до середины третьего года. Балансовая стоимость для периода = 1, 80000, а для периода = 2, 64000. При помощи линейной интерполяции балансовая стоимость для периода = 1,5 имеем $64000 + (80000 - 64000) * 0,5 = 72000$; аналогично балансовая стоимость для периода = 2,5 составляет 57600. Возвращаемая амортизация — разница между этими балансовыми стоимостями, то есть $72000 - 57600 = 14400$.

В строке 11 возвращается 5303,60 руб., сумма амортизации за 8-й год. Балансовая стоимость на начало 8-го года — 20910,80 ($100000 - VDB(100000; 5000; 10; 0; 7)$). Линейная амортизация (синяя линия) составляет $(20910,80 - 5000) / 3 = 5303,60$, что больше, чем амортизация методом двойного уменьшения остатка (красная линия, она расположена выше синей линии, поскольку амортизация меньше).

В строке 12 возвращается 5303,60 руб., как указано выше; линейная амортизация является постоянной.

В строке 13 возвращается 89262,58 руб. С запрещенной линейной амортизацией суммарная амортизация в течение всего срока ниже ожидаемых 95000 руб., необходимых для

уменьшения балансовой стоимости до 5000 руб. (красная линия).



Округление результата до 2 знаков после запятой соответствует поведению Excel, что неудобно для тех, чья валюта обычно не выражается 2 десятичными знаками.

VDB(100000; 20000; 10; 7,25; 7,75) возвращает ошибочный результат как в Excel так и в Calc (разные в каждом). Это, как представляется, затрагивает лишь дробные периоды в году, когда амортизация ограничена.

XIRR

Функция **XIRR** рассчитывает внутренний уровень доходности ряда нерегулярных денежных потоков.

Синтаксис функции:

=XIRR(payments; dates; guess)

где:

- **payments** — диапазон, содержащий платежи, сделанные или полученные, через нерегулярные интервалы. Первый платеж — отрицательное число, которое представляет инвестиции. Более поздние платежи дисконтируются на основе годовой учетной ставки и даты платежа. По крайней мере один из платежей должен быть положительным и один — отрицательным;
- **dates** — диапазон, содержащий даты, в которые платежи были сделаны или получены. Первая дата соответствует началу платежей;
- **guess** — (необязательное, по умолчанию — 10%) первое предположение о ставке.

Диапазоны **payments** и **dates** должны быть одинакового размера.

XIRR использует итеративный метод для вычисления нормы прибыли, которая дает нулевую чистую приведенную стоимость денежных потоков. По крайней мере один из потоков наличности должен быть отрицательным и по крайней мере один положительный — для того, чтобы чистая приведенная стоимость была равна нулю. Показатель прибыли за год, и проценты, как предполагается начисляются ежегодно, год считается продолжительностью 365 дней. Если быть более точным, уравнение решения имеет вид:

$$\sum_i \frac{payment_i}{(1 + XIRR)^{(date_i - date_0)/365}} = 0$$

Порядок, в котором указаны платежи/даты, не важен, за исключением того, что первый платеж должен иметь самую раннюю дату.

	A	B	C	D	E
1					
2		Платежи		Даты	
3		-2750	05.02.2008		
4		1000	05.07.2008		
5		2000	05.01.2009		
6					
7			12,4%	=XIRR(B3:B5;C3:C5)	
8					

Диапазон B3:B5 содержит суммы -2750, 1000, 2000, а диапазон C3:C5 — даты “05.02.2008”, “05.07.2008” и “05.01.2009”. Функция XIRR возвращает приблизительно 0,124 или 12,4 %. Введенные в диапазоне C3:C5 даты, отображаются как установлено в региональных настройках.

Эта функция в настоящее время не принимает массивы.

XNPV

Функция **XNPV** возвращает чистую приведенная стоимость инвестиций с нерегулярными наличными расчетами.

Синтаксис функции:

=XNPV(rate; payments; dates)

где:

- **rate** — ежегодная процентная ставка. Значение должно быть больше -1;
- **payments** — диапазон или множество, содержащий платежи, сделанные или полученные, через нерегулярные интервалы. Первый платеж — отрицательное число, которое представляет инвестиции. Более поздние платежи дисконтируются на основе годовой учетной ставки и даты платежа. По крайней мере один из платежей должен быть положительным и один — отрицательным;
- **dates** — диапазон или множество, содержащий даты, в которые платежи были сделаны или получены. Первая дата соответствует началу платежей.

Диапазоны **payments** и **dates** должны быть одинакового размера.

XNPV вычисляет чистую приведенную стоимость на заданную первую дату ($date_0$), используя формулу:

$$\sum_i \frac{pay_i}{(1 + rate)^{(date_i - date_0)/365}}$$

где pay_i — i -я оплата, а $date_i$ — i -я дата; i начинается с 0.

	A	B	C	D	E	F
1						
2		Ежегодная процентная ставка		5,00%		
3						
4		Платежи	Даты			
5		-1000	01.01.2008			
6		500	01.01.2009			
7		550	01.07.2009			
8						
9				-12,65	=XNPV(D2;B5:B7;C5:C7)	
10						

Если Вы приобрели инвестиции за 1000 руб. 2008-01-01, по которым выплатили 500 руб. 2009-01-01 и 550 руб. 2009-07-01 Вы сделали небольшую потерю, по сравнению с вложением с 5%-ой процентной ставкой.

	A	B	C	D	E	F
1						
2		Ежегодная процентная ставка		5,00%		
3						
4		Платежи	Даты			
5		0	01.01.2008			
6		500	01.01.2009			
7		550	01.07.2009			
8						
9				987,35	=XNPV(D2;B5:B7;C5:C7)	
10						

Возвращается приблизительно 987,35 руб. — чистая приведенная стоимость на 2008-01-01 по инвестициям, по которым заплатили 500 руб. на 2009-01-01 и 550 руб. на 2009-07-01, используя 5%-ую процентную ставку.

Расчет основан на том, что в году 365 дней, даже в високосный год, используя фактическую разницу дней. Есть и другие способы для расчета чистой приведенной стоимости.

YIELD

Функция **YIELD** вычисляет доходность для ценных бумаг, по которым производятся периодические выплаты процентов.

Синтаксис функции:

=YIELD(settlementdate; maturitydate; rate; price; redemptionvalue; frequency; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **rate** — (ежегодная) процентная ставка по ценным бумагам;
- **price** — курс ценных бумаг за 100 валютных единиц номинальной стоимости;
- **redemptionvalue** — цена погашения ценных бумаг за 100 валютных единиц номинальной стоимости;
- **frequency** — количество выплат процентов в год (1, 2 или 4);
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;

- 1 - точное число дней в месяцах, точное число дней в году;
- 2 - точное число дней в месяце, год имеет 360 дней;
- 3 - точное число дней в месяце, год имеет 365 дней;
- 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция вычисляет доходность для ценных бумаг. Она использует функцию **PRICE**, и повторно выполняет расчеты (до 100 раз), чтобы найти доходность, которую даст **price**.

	A	B	C	D	E	F
1						
2		Дата покупки	15.02.2008			
3		Дата погашения	15.11.2010			
4		Процентная ставка	5,00%			
5		Курс	95,00			
6		Цена погашения	100,00			
7		Кол-во выплат в год	2			
8						
9				0,070	=YIELD (C2;C3;C4;C5;C6;C7;0)	
10						

Возвращается приблизительно 0,070 или 7 %.

Эта функция не точна, поскольку она полагается на итерационное вычисление. Таким образом, она может дать немного отличные от других электронных таблиц результаты. В практических целях какие-либо различия вряд ли будут существенны.

YIELDDISC

Функция **YIELDDISC** вычисляет доходность для дисконтных облигаций по которым выплата процентов не производится.

Синтаксис функции:

=YIELDDISC(settlementdate; maturitydate; price; redemptionvalue; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **price** — курс ценных бумаг за 100 валютных единиц номинальной стоимости;
- **redemptionvalue** — цена погашения ценных бумаг за 100 валютных единиц номинальной стоимости;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

YIELDDISC вычисляет ежегодную доходность по облигациям, по которым не выплачиваются проценты (“чистый дисконтный инструмент” или “дисконтная облигация с нулевым купоном”). Она возвращает:

$$\frac{\left(1 - \frac{\text{price}}{\text{redemptionvalue}}\right)}{\text{YEARFRAC}(\text{settlementdate}; \text{maturitydate}; \text{basis})}$$

Сопутствующая функция **PRICEDISC** имеет параметр “учетная ставка”. Отметьте, что это не то же самое, что “доходность”, хотя эти два понятия связаны, и в целом близки.

	A	B	C	D	E	F
1						
2		Дата покупки		15.02.2008		
3		Дата погашения		30.11.2008		
4		Курс		97,63		
5		Цена погашения		100,00		
6						
7				0,0307	=YIELDDISC(C2;C3;C4;C5;1)	
8						

Возвращается приблизительно 0,0307 или 3,07 %. Вы покупаете и оплачиваете облигацию за 976.30 руб. 15 февраля 2008; облигация погашается 30 ноября 2008, принося ее номинальную стоимость 1000 руб. Доходность — приблизительно 3 %, при использовании календарной системы с **basis** = 1.

Бывают (редкие) ситуации, когда результаты в Calc и Excel отличаются.

Используемая формула не учитывает начисляемых процентов. Она может быть наиболее подходящей, когда используется с периодами до одного года.

YIELDMAT

Функция **YIELDMAT** вычисляет доходность по облигациям, по которым выплата процентов производится при наступлении срока погашения.

Синтаксис функции:

=YIELDMAT(settlementdate; maturitydate; issuedate; rate; price; basis)

где:

- **settlementdate** — дата сделки (покупки) по ценным бумагам;
- **maturitydate** — дата погашения (выкупа) ценных бумаг;
- **issuedate** — дата выпуска (эмиссии) ценных бумаг;
- **rate** — (ежегодная) процентная ставка по ценным бумагам;
- **price** — курс ценных бумаг за 100 валютных единиц номинальной стоимости;
- **basis** — определяет используемую календарную систему. По умолчанию 0 если опущено:
 - 0 - американский метод (NASD), 12 месяцев по 30 дней каждый;
 - 1 - точное число дней в месяцах, точное число дней в году;
 - 2 - точное число дней в месяце, год имеет 360 дней;
 - 3 - точное число дней в месяце, год имеет 365 дней;
 - 4 - европейский метод, 12 месяцев по 30 дней каждый.

Эта функция вычисляет доходность по облигациям, по которым выплата процентов производится только однажды, при наступлении срока погашения.

YIELDMAT возвращает доходность, задаваемую следующим уравнением:

$$1 + yield * YEARFRAC(settlementdate; maturitydate; basis) = \frac{maturity_value}{settlement_value}$$

	A	B	C	D	E	F
1						
2		Дата покупки	15.02.2007			
3		Дата погашения	06.04.2007			
4		Дата выпуска	06.01.2007			
5		Процентная ставка	5,00%			
6		Курс	99,00			
7						
8			0,1211	=YIELDMAT(C2;C3;C4;C5;C6;0)		
9						

Возвращается приблизительно 0,1211. Вы покупаете и оплачиваете облигацию 15 февраля 2007; (90 дневная) облигация была выпущена 6 января 2007 и погашение наступает 6 апреля 2007, принося ее номинальную стоимость 1000 руб. и выплачивая проценты 1000 руб. * 5 % * 90/360. Цена — 99, дает стоимость 990 руб.; доходность при погашении — приблизительно 12,11 %.

Бывают (редкие) ситуации, когда результаты в Calc и Excel отличаются.

Используемая формула не учитывает начисляемых процентов. Для периодов в течение одного года стоит относиться с определенной осторожностью при интерпретации доходности.

Цена рассчитывается по состоянию на дату сделки (когда деньги переходят к другому владельцу). Контракт на покупку облигаций (дата заключения сделки) может предшествовать этому (например, на 3 дня).

Примеры использования финансовых функций

Расчет по кредиту

В некоторый момент Вы захотите использовать OOo Calc для выполнения своего рода финансового анализа. В качестве введения в тему использования финансовых функций в OOo Calc, мы рассмотрим пример закладной на дом.

Функция **РМТ** рассчитывает регулярные платежи, которые необходимо будет заплатить по указанному кредиту с указанной процентной ставкой и указанном числе периодов платежей.

В примере ниже, мы имеем 30-летний кредит, на сумму 3 000 000 руб, который выплачивается ежемесячно с годовой процентной ставкой 8 %.

	A	B	C	D	E	F
1		30-и летний кредит				
2						
3		Годовая процентная ставка	8,00%			
4		Количество периодов	360			
5		Сумма кредита	3 000 000,00			
6		Сумма платежа	22 012,94 руб.	=PMT(C3/12;C4;C5)		
7						

Ежемесячная оплата по этому кредиту, оказывается, 22 096,65 руб. Что мы также хотели бы знать — как наша основная сумма кредита уменьшается в течение продолжительного времени. Каждый ежемесячный платеж делится на проценты по кредиту и сумму погашения кредита. Сначала в графике погашения, большая доля платежа приходится на проценты по кредиту.

Формула для вычисления основного баланса в конце каждого периода (D11:D370) следующая.

=FV(rate; nper; pmt; pv)

9			
10		01.01.2000	3 000 000,00 руб.
11		01.02.2000	2 997 987,06 руб. =FV(\$C\$3/12;1;\$C\$6;-C10)
12		01.03.2000	2 995 960,71 руб. =FV(\$C\$3/12;1;\$C\$6;-C11)
13		01.04.2000	2 993 920,84 руб. =FV(\$C\$3/12;1;\$C\$6;-C12)
14		01.05.2000	2 991 867,38 руб. =FV(\$C\$3/12;1;\$C\$6;-C13)
15		01.06.2000	2 989 800,22 руб. =FV(\$C\$3/12;1;\$C\$6;-C14)
16		01.07.2000	2 987 719,28 руб. =FV(\$C\$3/12;1;\$C\$6;-C15)
17		01.08.2000	2 985 624,48 руб. =FV(\$C\$3/12;1;\$C\$6;-C16)
18		01.09.2000	2 983 515,70 руб. =FV(\$C\$3/12;1;\$C\$6;-C17)
19		01.10.2000	2 981 392,87 руб. =FV(\$C\$3/12;1;\$C\$6;-C18)
20		01.11.2000	2 979 255,88 руб. =FV(\$C\$3/12;1;\$C\$6;-C19)

В данном случае мы вычисляем будущую стоимость кредита через один период времени, на основе текущей стоимости кредита и процентной ставки.

360		01.03.2029	212 268,62 руб. =FV(\$C\$3/12;1;\$C\$6;-C359)
361		01.04.2029	191 670,81 руб. =FV(\$C\$3/12;1;\$C\$6;-C360)
362		01.05.2029	170 935,68 руб. =FV(\$C\$3/12;1;\$C\$6;-C361)
363		01.06.2029	150 062,31 руб. =FV(\$C\$3/12;1;\$C\$6;-C362)
364		01.07.2029	129 049,79 руб. =FV(\$C\$3/12;1;\$C\$6;-C363)
365		01.08.2029	107 897,18 руб. =FV(\$C\$3/12;1;\$C\$6;-C364)
366		01.09.2029	86 603,56 руб. =FV(\$C\$3/12;1;\$C\$6;-C365)
367		01.10.2029	65 167,98 руб. =FV(\$C\$3/12;1;\$C\$6;-C366)
368		01.11.2029	43 589,50 руб. =FV(\$C\$3/12;1;\$C\$6;-C367)
369		01.12.2029	21 867,16 руб. =FV(\$C\$3/12;1;\$C\$6;-C368)
370		01.01.2030	0,00 руб. =FV(\$C\$3/12;1;\$C\$6;-C369)
371			

Расчет суммы погашения и процентов по кредиту

Здесь мы покажем, как определить сумму погашения кредита и проценты выплаты по кредиту, используя встроенные в ООо Calc функции **IPMT** и **PPMT**. Для стандартного или ипотечного кредита с фиксированной процентной ставкой в течение срока действия кредита, один регулярный платеж может быть рассчитан при помощи функции **PMT**. Это обсуждалось в предыдущем совете.

Однако, поскольку сумма непогашенного остатка уменьшается с течением времени, сумма погашения и проценты по кредиту изменяются.

ООо Calc предоставляет две функции, **IPMT** и **PPMT**, которые вычисляют сумму погашения кредита и выплаты процентов по кредиту для любой выплаты.

В примере ниже, параметры кредита определены в **C3:C5**, а оплата по кредиту в **C6** рассчитывается как

=-PMT(C3;C4;C5)

сумма погашения в **C9** рассчитывается как

=-PPMT(\$C\$3;B9;\$C\$4;\$C\$5)

а проценты в **D9** рассчитывается как

=-IPMT(\$C\$3;B9;\$C\$4;\$C\$5)

В течение последующих периодов, единственный аргумент этих функций, который изменяется — номер периода.

В случае, если вам нужны дополнительные убедительные аргументы, компоненты **IPMT** и

PPMT суммируются в столбце E, и сумма является постоянной на протяжении всего срока кредита - как и ожидалось.

	A	B	C	D	E	F
1		IPMT и PPMT				
2						
3		Процентная ставка	0,67%			
4		Кол-во периодов	360			
5		Сумма кредита	2000000			
6		Платеж	14 731,10 р.			
7						
8		Период	PPMT	IPMT	IPMT + PPMT	
9		1	1 331,10 р.	13 400,00 р.	14 731,10 р.	
10		2	1 340,02 р.	13 391,08 р.	14 731,10 р.	
11		3	1 349,00 р.	13 382,10 р.	14 731,10 р.	
12		4	1 358,03 р.	13 373,07 р.	14 731,10 р.	
13		5	1 367,13 р.	13 363,97 р.	14 731,10 р.	
14		6	1 376,29 р.	13 354,81 р.	14 731,10 р.	
15		7	1 385,51 р.	13 345,59 р.	14 731,10 р.	
16		
17		354	14 058,36 р.	672,74 р.	14 731,10 р.	
18		355	14 152,55 р.	578,55 р.	14 731,10 р.	
19		356	14 247,37 р.	483,73 р.	14 731,10 р.	
20		357	14 342,83 р.	388,27 р.	14 731,10 р.	
21		358	14 438,93 р.	292,17 р.	14 731,10 р.	
22		359	14 535,67 р.	195,43 р.	14 731,10 р.	
23		360	14 633,06 р.	98,04 р.	14 731,10 р.	
24						

Функции CUMIPMT и CUMPRINC

Полезно знать для выданного кредита — итоговый процент, подлежащий выплате по ходу погашения кредита — или итоговый процент, выплаченный к определенному моменту. В ООО Calc существует созданная специально для этой цели функция — CUMIPMT. Парная функция — CUMPRINC вычисляет основную сумму, погашенную к определенному моменту.

Электронная таблица ниже иллюстрирует использование функция CUMIPMT и CUMPRINC.

	A	B	C	D	E	F	G
1		CUMIPMT и CUMPRINC					
2							
3		Сумма кредита	\$200 000,00				
4		Срок кредита	30				
5		Кол-во платежей в год	12				
6		Годовая процентная ставка	8,00%				
7							
8		Кол-во периодов оплаты	360	=D\$4 * D\$5			
9		Процент за период	0,667%	=D\$6 / D\$5			
10							
11		Платеж по кредиту	\$1 467,53	=PMT(D\$9;D\$8;D\$3;0)			
12							
13		Итоговая сумма процентов	\$328 310,49	=CUMIPMT(D\$9;D\$8;D\$3;1;D\$8;0)			
14		Итоговая сумма погашения	\$200 000,00	=CUMPRINC(D\$9;D\$8;D\$3;1;D\$8;0)			
15		Итоговая сумма платежей	\$528 310,49	=D\$13 + D\$14			
16							

Диапазон **D3:D6** — входные параметры. Измените любое из этих значений, чтобы исследовать их влияние на результаты.

В этом примере, **CUMIPMT** и **CUMPRINC** вычисляются для последнего платежа по кредиту — и как и ожидалось, **CUMPRINC** равняется сумме кредита. Кроме того, полные платежи по ходу погашения кредита — просто сумма **CUMIPMT** и **CUMPRINC**.

Следующий пример показывает, как **CUMIPMT** и **CUMPRINC** могут быть вычислены в любой момент в течение срока кредита.

	A	B	C	D	E	F	G
1							
2		Сумма кредита		\$200 000,00			
3		Срок кредита		30			
4		Кол-во платежей в год		12			
5		Годовая процентная ставка		8,00%			
6							
7		Кол-во периодов оплаты		360	=D\$3 * D\$4		
8		Процент за период		0,667%	=D\$5 / D\$4		
9							
10		Платеж по кредиту		\$1 467,53	=-PMT(D\$8;D\$7;D\$2;0)		
11							
12		Период	IPMT	PPMT	CUMIPMT	CUMPRINC	
13		1	\$1 333,33	\$134,20	\$1 333,33	\$134,20	
14		2	\$1 332,44	\$135,09	\$2 665,77	\$269,29	
15		3	\$1 331,54	\$135,99	\$3 997,31	\$405,28	
16		4	\$1 330,63	\$136,90	\$5 327,94	\$542,17	
17		5	\$1 329,72	\$137,81	\$6 657,66	\$679,99	
18		10	\$1 325,06	\$142,47	\$13 292,35	\$1 382,94	
19		20	\$1 315,28	\$152,25	\$26 489,69	\$2 860,89	
20		...					
21		100	\$1 208,46	\$259,07	\$127 762,27	\$18 990,65	
22		...					
23		200	\$964,04	\$503,49	\$237 608,20	\$55 897,63	
24		...					
25		355	\$57,36	\$1 410,17	\$328 165,99	\$192 806,85	
26		356	\$47,95	\$1 419,57	\$328 213,95	\$194 226,43	
27		357	\$38,49	\$1 429,04	\$328 252,44	\$195 655,47	
28		358	\$28,96	\$1 438,57	\$328 281,40	\$197 094,03	
29		359	\$19,37	\$1 448,16	\$328 300,77	\$198 542,19	
30		360	\$9,72	\$1 457,81	\$328 310,49	\$200 000,00	

Формулы в **C13:F13**:

=-IPMT(D\$8;B13;D\$7;D\$2)

=-PPMT(D\$8;B13;D\$7;D\$2)

=-CUMIPMT(D\$8;D\$7;D\$2;1;B13;0)

=-CUMPRINC(D\$8;D\$7;D\$2;1;B13;0)

Эти формулы могут быть скопированы для заполнения оставшихся строк.

Обратите внимание, что без знака “-” перед вышеупомянутыми формулами, Вы увидели бы отрицательные числа — правильное отражение направления потока капитала.

Создание таблицы выплаты

Одна из самых полезных и широко используемых финансовых функций, доступных в OpenOffice Calc - **PMT**. Мы обсуждали ее выше.

Ее использование проиллюстрировано ниже в простом примере, показывающем, как вычислять платежи по кредиту с фиксированной процентной ставкой по заданному сроку

выплаты.

	A	B	C	D	E
1					
2		Исходные данные			
3		Сумма кредита	\$10 000,00		
4		Ставка (ежегодная)	7,25%		
5		Период (месяцы)	1		
6		Срок кредита (месяцы)	36		
7					
8		Выходные данные			
9		Сумма платежа	\$309,92	=PMT(C4/12;C6;-C3;0)	
10		Общая сумма платежей	\$11 156,95	=C9*C6	
11		Сумма выплаченных процентов	\$1 156,95	=C10-C3	
12					

Преимущество использования электронной таблицы таким образом, заключается в том, что мы можем включить различные параметры в уравнение и немедленно увидеть, как они влияют на суммы выплаты, которую нам необходимо будет сделать.

Теперь я расширю эту возможность, показав, как создать таблицу платежей для диапазона сумм кредита и процентных ставок. В первую очередь мы создадим пустую таблицу с требуемыми суммами кредита сбоку, и процентными ставками — сверху.

В ячейке G4, мы введем в следующую формулу:

=PMT(G\$2/12;\$C\$6;-\$F3;0)

	F	G	H	I	J	K
1						
2		7,00%	7,25%	7,50%	7,75%	
3	\$8 000,00	\$247,02				
4	\$9 000,00					
5	\$10 000,00					
6	\$11 000,00					
7	\$12 000,00					
8	\$13 000,00					
9	\$14 000,00					
10	\$15 000,00					
11						

Использование “гибридного” режима адресации позволяет получить в ООо Calc правильные результаты когда мы применяем автозаполнение формул вниз в первом столбце.

	F	G	H	I	J	K
1						
2		7,00%	7,25%	7,50%	7,75%	
3	\$8 000,00	\$247,02				
4	\$9 000,00	\$277,89				
5	\$10 000,00	\$308,77				
6	\$11 000,00	\$339,65				
7	\$12 000,00	\$370,53				
8	\$13 000,00	\$401,40				
9	\$14 000,00	\$432,28				
10	\$15 000,00	\$463,16				
11						

Таким же образом, мы можем использовать автозаполнение для всей таблицы, растянув первую колонку поперек за указатель заполнения как показано ниже.

	F	G	H	I	J	K
1						
2		7,00%	7,25%	7,50%	7,75%	
3	\$8 000,00	\$247,02	\$247,93	\$248,85	\$249,77	
4	\$9 000,00	\$277,89	\$278,92	\$279,96	\$280,99	
5	\$10 000,00	\$308,77	\$309,92	\$311,06	\$312,21	
6	\$11 000,00	\$339,65	\$340,91	\$342,17	\$343,43	
7	\$12 000,00	\$370,53	\$371,90	\$373,27	\$374,65	
8	\$13 000,00	\$401,40	\$402,89	\$404,38	\$405,88	
9	\$14 000,00	\$432,28	\$433,88	\$435,49	\$437,10	
10	\$15 000,00	\$463,16	\$464,87	\$466,59	\$468,32	
11						

Накопление

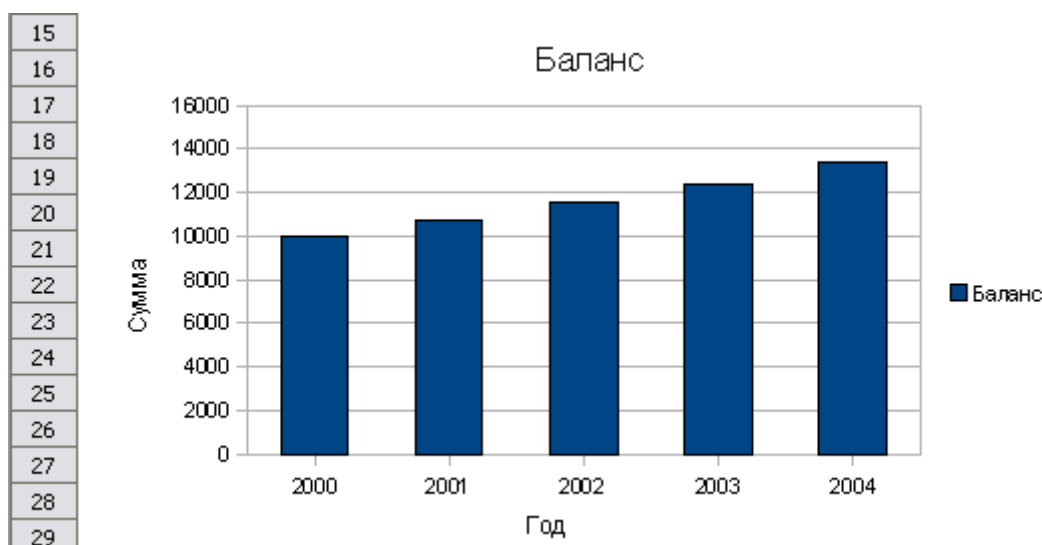
Как следует из ипотечных примеров, приведенных ранее, здесь мы рассмотрим небольшую сумму, вложенную на несколько лет по фиксированной процентной ставке.

Функция, которую мы будем использовать

=FV(rate; nper; pmt; pv)

Рассмотрим следующую ситуацию: инвестируется 10 000 руб. в счет, который имеет ежегодную процентную ставку 7,5%. Какая прибыль будет через четыре года...

	A	B	C	D	E
2					
3		Инвестируемая сумма	10 000,00 руб.		
4		Ежегодная процентная ставка	7,50%		
5		Кол-во периодов	4		
6		Конечная сумма	13 354,69 руб.	=FV(C4;C5;0;-C3)	
7					
8		Год	Баланс		
9		2000	10 000,00 руб.		
10		2001	10 750,00 руб.	=FV(\$C\$4;1;0;-C9)	
11		2002	11 556,25 руб.	=FV(\$C\$4;1;0;-C10)	
12		2003	12 422,97 руб.	=FV(\$C\$4;1;0;-C11)	
13		2004	13 354,69 руб.	=FV(\$C\$4;1;0;-C12)	
14					



Сложное накопление

В этом примере, мы рассмотрим более сложное накопление — в нем мы делаем регулярные платежи, а также зарабатываем проценты на наших первоначальных инвестициях.

Здесь, мы определяем имена для ячеек, содержащих основные переменные — для того, чтобы формулы были более удобочитаемы.

Я инвестирую 10 000 \$ в счет, на который начисляется 7.5 % ежегодно. Сколько будет накоплено денег после четырех лет.

	A	B	C	D	E	F	G
1							
2		Текущая стоимость		\$10 000,00		PV	
3		Годовая процентная ставка		7,75%			
4		Месячная процентная ставка		0,65%		RATE	
5		Количество периодов		84		NPER	
6		Взнос		\$500,00		PMT	
7		Окончательная сумма		\$72 706,26		=FV(RATE;NPER;-PMT;-PV)	
8							
9							
10		Дата	Баланс				
11		01.2001	\$10 000,00				
12		02.2001	\$10 564,58		=FV(RATE;1;-PMT;-C11)		
13		03.2001	\$11 132,81		=FV(RATE;1;-PMT;-C12)		
14		04.2001	\$11 704,71		=FV(RATE;1;-PMT;-C13)		
15		05.2001	\$12 280,31		=FV(RATE;1;-PMT;-C14)		
16		06.2001	\$12 859,62		=FV(RATE;1;-PMT;-C15)		
17		07.2001	\$13 442,67		=FV(RATE;1;-PMT;-C16)		
18		08.2001	\$14 029,48		=FV(RATE;1;-PMT;-C17)		
19		09.2001	\$14 620,09		=FV(RATE;1;-PMT;-C18)		
20		10.2001	\$15 214,51		=FV(RATE;1;-PMT;-C19)		
21		11.2001	\$15 812,77		=FV(RATE;1;-PMT;-C20)		

Конечная цель — проанализировать счет по месяцам, а также подтвердить точность нашей первоначальной формулы.

83		01.2007	\$61 545,57		=FV(RATE;1;-PMT;-C82)		
84		02.2007	\$62 443,05		=FV(RATE;1;-PMT;-C83)		
85		03.2007	\$63 346,33		=FV(RATE;1;-PMT;-C84)		
86		04.2007	\$64 255,44		=FV(RATE;1;-PMT;-C85)		
87		05.2007	\$65 170,43		=FV(RATE;1;-PMT;-C86)		
88		06.2007	\$66 091,32		=FV(RATE;1;-PMT;-C87)		
89		07.2007	\$67 018,16		=FV(RATE;1;-PMT;-C88)		
90		08.2007	\$67 950,99		=FV(RATE;1;-PMT;-C89)		
91		09.2007	\$68 889,84		=FV(RATE;1;-PMT;-C90)		
92		10.2007	\$69 834,75		=FV(RATE;1;-PMT;-C91)		
93		11.2007	\$70 785,76		=FV(RATE;1;-PMT;-C92)		
94		12.2007	\$71 742,92		=FV(RATE;1;-PMT;-C93)		
95		01.2008	\$72 706,26		=FV(RATE;1;-PMT;-C94)		
96							

Пример Шарового кредита

Шаровая ипотека — краткосрочный ипотечный кредит, длительностью пять — семь лет, но платежи основываются на сроке 30 лет. Шаровые кредиты часто имеют более низкую процентную ставку, и может быть, как правило, легче получен чем традиционный 30-летний ипотечный кредит. В конце срока вашего кредита Вам необходимо будет погасить вашу

задолженность по взносам. Это обычно означает рефинансирование, продажу дома или переход к обычной ипотеке с текущей процентной ставкой.

В электронной таблице ниже, мы используем функции **PMT** и **FV** для вычисления суммы платежа и окончательного платежа за 7-летний 8%-ный шаровой кредит в 100 000 \$.

	A	B	C	D	E	F	G
1			Шаровый ипотечный кредит				
2							
3		Входные параметры					
4		Сумма кредита		\$100 000			
5		Процентная ставка		8,00%			
6		Кол-во периодов (лет)		7			
7							
8		Сводка по кредиту					
9		Ежемесячный платеж		\$733,76	=PMT(\$D\$5/12; 360; -\$D\$4; 0; 0)		
10		Шаровой платеж		\$92 477,43	=FV(\$D\$5/12; \$D\$6*12; \$D\$9; -\$D\$4; 0)		
11							
12		Суммарные платежи		\$61 636,22	=\$D\$9*\$D\$6*12		
13		Суммарные выплаты процентов		\$54 113,66	=\$D\$12-(\$D\$4-\$D\$10)		
14							

Позвольте нам теперь модифицировать этот рабочий лист со сценарием “а что, если” — столь легким в ООо Calc. Допустим мы увеличим ежемесячную выплату и увидим, как это воздействует на нашу причитающуюся заключительную выплату.

	A	B	C	D	E	F	G
1			Шаровый ипотечный кредит				
2							
3		Входные параметры					
4		Сумма кредита		\$100 000			
5		Процентная ставка		8,00%			
6		Кол-во периодов (лет)		7			
7		Дополнительная оплата		\$100			
8							
9		Сводка по кредиту					
10		Ежемесячный платеж		\$833,76	=PMT(\$D\$5/12; 360; -\$D\$4; 0; 0)+\$D\$7		
11		Шаровой платеж		\$81 266,10	=FV(\$D\$5/12; \$D\$6*12; \$D\$10; -\$D\$4; 0)		
12							
13		Суммарные платежи		\$70 036,22	=\$D\$10*\$D\$6*12		
14		Суммарные выплаты процентов		\$51 302,33	=\$D\$13-(\$D\$4-\$D\$11)		
15							

Функции электронных таблиц

ADDRESS

Функция **ADDRESS** возвращает ссылку на ячейку в виде текста, заданную номером столбца и строки.

Синтаксис функции:

=ADDRESS(row; column; mode; style; sheet)

где:

- **row** — число, определяющее строку;
- **column** — число (а не буква), определяющее столбец;
- **mode** — режим (необязательное число) определяет, абсолютный ли адрес ячейки или относительный. Если опущен, он, как предполагают, равняется 1.

Режим	Строка	Столбец	style = TRUE	style = FALSE
1	абсолютный	абсолютный	\$A\$1	R1C1
2	абсолютный	относительный	A\$1	R1C[1]
3	относительный	абсолютный	\$A1	R[1]C1
4	относительный	относительный	A1	R[1]C[1]

- **style** - необязательный параметр, если **FALSE** или 0 определяет стиль R1C1, любое другое значение или если опущен определяет стиль A1. Хотя стиль записи R1C1 не поддерживается Oo Calc, все же, разрешается импортированным электронным таблицам вычислить формулы с его использованием;
- **sheet** — необязательная текстовая строка, определяющая лист.

Функция **ADDRESS** возвращает адрес ячейки (ссылку) в виде текста. Если задается имя листа параметром **sheet**, имя листа возвращается в тексте, сопровождаемое “.” и ссылкой на столбец/строку. Столбцы в возвращаемой строке идентифицируются с использованием заглавных букв.

	A	B	C	D	E
1					
2		Лист1.C\$4	=ADDRESS(4;3;2;1;"Лист1")		
3					
4		C4	=ADDRESS(4; 3; 4)		
5					
6		\$C\$4	=ADDRESS(4; 3)		
7					

Excel представляет некоторые ссылки на ячейку по-другому, нежели Calc, таким образом эта функция не всегда переносима. Например, **ADDRESS(1; 1; 4; "Sheet2")**, возвращает **Sheet2.A1** в Calc; эквивалент в Excel возвращает **Sheet2!A1**.

Excel допускает стиль ссылки R1C1; это будет работать в Calc начиная с Oo3.0.

Проект стандарта ODFD говорит, что **mode** по умолчанию (если **mode** опущен) должен быть 4. Calc и Excel оба по умолчанию используют значение 1, таким образом могут возникнуть проблемы с ODFD.

AREAS

Функция **AREAS** возвращает количество областей в заданной ссылке.

Синтаксис функции:

=AREAS(reference)

где:

- **reference** — ссылка, может состоять из нескольких диапазонов.

Функция **AREAS** возвращает количество диапазонов. Она главным образом полезна со ссылками на именованные диапазоны; несколько диапазонов могут быть введены в диалоговом окне **Вставка — Названия — Определить...**, вручную объединяя диапазоны в поле ввода **Назначен на**.

	A	B	C	D	E
1					
2					
3					
4					
5					

В ячейке **B2** возвращается **1**. Ссылка **A5:B7** — одиночный диапазон.

В ячейке **B4** возвращается **2**. Именованный диапазон **myref** был определен в диалоговом окне **Вставка — Названия — Определить...** как двойной диапазон (например, введением **\$Лист1.\$A\$1:\$A\$5~\$Лист1.\$C\$1:\$C\$5** в поле **Назначен на**).

CHOOSE

Функция **CHOOSE** возвращает значение из списка, задаваемое числовым индексом.

Синтаксис функции:

=CHOOSE(index; value1; value2; ... value30)

где:

- **index** — числовой индекс, определяющий какое из значений **value1...value30** будет возвращено; если 1 — возвращается **value1**, если 2 — возвращается **value2**, и т.д. ;
- **value1; value2; ... value30** — до 30 значений, каждое из которых может быть текстом, числом, логическим значением, ссылкой или формулой.

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

В ячейке **B2** возвращается текстовая строка “собака”.

В ячейке **B4** возвращается числовое значение 65,4.

В ячейке **B6** возвращается ссылка на ячейку **A6**, так что **=CHOOSE(3; "собака"; 65,4; A6)** в

ячейке отображает содержимое ячейки **A6** в этой ячейке (она эквивалентна **=A6**).

В ячейке **B8** возвращается сумма чисел в диапазоне **A2:A4**.

В ячейке **B10** возвращается 3, квадратный корень 9.

Функция **CHOOSE** пригодится в том случае, когда есть несколько вариантов для конкретного значения, выбирающиеся на основе результата выражения. При этом отпадает необходимость строить сложные формулы на основе вложенных **IF**.

В примере ниже, функция **CHOOSE** используется, для выбора варианта текста дня недели — согласно числового значения, возвращаемого функцией **WEEKDAY**.

	A	B	C	D	E	F	G
1							
2		18.08.08	Пн				
3							
4							

=CHOOSE(WEEKDAY(B2);"Вс";"Пн";"Вт";"Ср";"Чт";"Пт";"Сб")

COLUMN

Функция **COLUMN** возвращает номер(а) столбца, задаваемого ссылкой.

Синтаксис функции:

=COLUMN(reference)

где:

- **reference** — ссылка на ячейку или диапазон, может задаваться либо явно (например, **A3:B5**), либо по имени (например, **myref**).

Функция **COLUMN** возвращает номер столбца ссылки, где “A” — 1, “B” — 2, и так далее. Если **reference** опущена, возвращается номер столбца текущей ячейки (содержащей формулу). Если ссылка — диапазон ячеек, а формула введена как формула массива (нажатием *Ctrl-Shift-Enter*, а не *Enter*), возвращается однострочковый массив номеров столбцов в ссылке.

	A	B	C	D	E
1					
2			2		
3					
4			2		
5					
6			4		
7					
8			4	5	6
9			{=COLUMN(D4:F5)}		
10					

В ячейке **B2** возвращается 2. Ячейка **B3** находится во втором столбце (столбец B).

В ячейке **B4** возвращается 2. Формула введена в ячейке B4, которая находится во втором столбце.

В ячейке **B6** возвращается 4. Это не формула массива, таким образом возвращается номер столбца первой ячейки **D2**.

В ячейках **B6:D6** возвращается 4, 5 и 6. Возвращается однострочковый массив {4; 5; 6}, поскольку это формула массива.

COLUMNS

Функция **COLUMNS** возвращает количество столбцов в заданной ссылке.

Синтаксис функции:

=COLUMNS(reference)

где:

- **reference** — ссылка на ячейку или диапазон, может задаваться либо явно (например, **A3:B5**), либо по имени (например, **myref**).

Возможно определить имя, которое представляет более чем один диапазон (например, введя **\$Лист1.\$A\$1:\$A\$5~\$Лист1.\$C\$1:\$C\$5** в поле **Назначен на** диалогового окна **Вставка — Названия — Определить...**). В этом случае **COLUMNS** суммирует количество столбцов в каждом диапазоне, в независимости от любого дублирования столбцов.

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					

2	=COLUMNS(A3:B5)
4	=COLUMNS(G4:H4)
6	=COLUMNS(D8)
8	=COLUMNS(myref)

В ячейке **B2** возвращается **2**; есть два столбца в заданном диапазоне, а именно, столбцы **A** и **B**.

В ячейке **B4** возвращается **2**; есть два столбца в заданном диапазоне, а именно, столбцы **G** и **H**.

В ячейке **B6** возвращается **1**; **D8** интерпретируется как диапазон, состоящий из одной ячейки, с одним столбцом (столбец **D**).

В ячейке **B8** возвращается **3**, именованный диапазон **myref** был определен как диапазон, состоящий из нескольких областей, **\$A\$1:\$A\$3~\$C\$1:\$D\$4**.

В ООo3 функция **COLUMNS** с диапазонами, состоящими из нескольких областей, возвращает значение ошибки **Err:504**.

DDE

Функция **DDE** возвращает данные на основе DDE связи.

Синтаксис функции:

=DDE(server; file; item; mode)

где:

- **server** — (текстовое) имя сервера; например “**soffice**” для OpenOffice.org;
- **file** — (текстовое) полное имя или IRL (Internet Resource Locator) документа, содержащего данные; например “**c:\office\document\test1.ods**” или “**file:///documents/test2.ods**”;
- **item** — (текстовое) имя возвращаемого элемента. Например “**Лист1.B3**” — ячейка **B3** на **Лист1** электронной таблицы Calc; “**MyRange**” — имя диапазона в электронной таблице Calc; “**MyItem**” — имя раздела или таблицы в документе Writer;

- **mode** — необязательный параметр, определяющий, как возвращаются числа. Нечисловой текст возвращается неизменным:
 - 0 или опущен: данные преобразуются к числу если возможно, используя стиль ячейки по умолчанию;
 - 1: данные преобразуются к числу если возможно, в формате английский США (en_US) (например, используя “.” как десятичный разделитель);
 - 2: Данные всегда восстанавливаются как текст (не преобразуются в число).

Функция **DDE** выполняет запрос по протоколу DDE и возвращает его результат. Обычно это просто локальная операция на машине, но эта спецификация не налагает подобного ограничения.

DDE создает связь к элементу (и возвращает содержимое), таким образом любое изменение элемента отражается на результате **DDE**.

Альтернативный метод для создания DDE связи — скопировать ячейку или диапазон из исходного файла в буфера обмена, а затем используя команду меню **Правка — Вставить как**, выбрать пункт **Связь по DDE**. Связь вставляется как функция DDE, как формула массива.

Команда меню **Правка — Связи...** отображает список DDE связей и позволяет их изменять, обновлять вручную или разрывать.

DDE поддерживает “Динамический обмен данными”, который является предшественником OLE, “Связывания и встраивания объектов”. Объекты DDE связаны по ссылке с источником; они не встроены в целевой документ.

=DDE("soffice";"C:\OpenOffice\test1.ods";"Лист1.C7")

Создает связь и возвращает содержимое ячейки **C7** на листе **Лист1** файла Calc **test1.ods**.

=DDE("soffice";"C:\OpenOffice\test2.odt";"Таблица1")

Когда введена как формула массива, создает связь и возвращает содержимое таблицы по имени **Таблица1** в файле Writer **test2.odt**, в одном столбце. Каждый абзац в ячейке таблицы появляется в своей собственной ячейке. Содержимое левой верхней ячейки таблицы отображается в первую очередь, за ним следуют по порядку другие ячейки в верхней строке, а затем аналогичным образом по порядку в других строках. Функция **INDEX** может использоваться для возврата только части таблицы.

=DDE("soffice";"C:\OpenOffice\test2.odt";"Раздел1")

Когда введена как формула массива, создает связь и возвращает содержимое раздела по имени **Раздел1** в файле Writer **test2.odt** в одном столбце, один абзац в ячейке. Когда введена обычно (не как формула массива), она возвращает первый абзац раздела **Раздел1**.

=INDEX(DDE("soffice";"C:\OpenOffice\test2.odt";"Раздел1");2)

Когда введена как формула массива, создает связь и возвращает содержимое второго абзаца раздела **Раздел1** в файле Writer **test2.odt**.

Excel не имеет этой функции.

Когда используется массив для возврата данных, размер массива закрепляется при первом вычислении. Если например число строк в таблице Writer или разделе изменяется, размер массива в Calc не изменяется.

Имеются альтернативные способы связи с внешними данными.

ERRORTYPE

Функция **ERRORTYPE** возвращает число, соответствующее значению ошибки.

Синтаксис функции:

=ERRORTYPE(reference)

где:

- **reference** — ссылка на ячейку, содержащую ошибку.

Функция **ERRORTYPE** возвращает код ошибки ячейки, определяемой **reference**, или **#N/A**, если эта ячейка не содержит ошибку.

Заметьте, что, если Вы выберете ячейку, содержащую ошибку, в строке состояния отобразится более полное описание этой ошибки.

	A	B	C	D	E
1	Ошибка:504				
2		504	=ERRORTYPE(A1)		
3					

В ячейке **B2** возвращается **504**, если ячейка **A1** отображает ошибку **Err:504**.

HLOOKUP

Функция **HLOOKUP** возвращает значение из строки таблицы, в столбце которого в первой строке найдено требуемое значение.

Синтаксис функции:

=HLOOKUP(lookupvalue; datatable; rowindex; mode)

где:

- **lookupvalue** — значение (число, текст или логическое значение), разыскиваемое в первой строке диапазона/массива. Когда значение найдено в первой строке, **HLOOKUP** возвращает соответствующее значение (в том же самом столбце) в строке **rowindexth** диапазона **datatable**, где для первой строки **rowindex** = 1;
- **datatable** — ссылка, которая должна включать по крайней мере две строки;
- **rowindex** — номер строки в массиве, содержащей значения, которые должны быть возвращены. Первая строка имеет номер 1;
- **mode** — необязательный параметр, который указывает, отсортирована ли первая строка в массиве в порядке возрастания. Если **mode** 0 или **FALSE**, первая строка не отсортирована в порядке возрастания, и ищется первое точное совпадение (поиск выполняется слева направо). В неотсортированных списках, значение поиска должно быть подобрано точно. Иначе функция возвратит сообщение об ошибке: **#N/A**. Если **mode** = 1, **TRUE** или опущен, первая строка **datatable** должна быть отсортирована, с числами, в порядке возрастания располагающимися перед текстовыми значениями в алфавитном порядке. Поиск по отсортированным строкам может выполняться намного быстрее, и функция всегда возвращает значение, даже если значение поиска не было точно подобрано, если оно находится между наименьшим и наибольшим значением отсортированного списка. **HLOOKUP** решает, где в первом ряду появляется **lookupvalue**. Если есть точное совпадение, оно соответствует найденному столбцу; если есть более чем одно точное совпадение, найденный столбец — не обязательно крайний левый.

Если регулярные выражения разрешены в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, **HLOOKUP** находит точное соответствие, рассматривая **lookupvalue** как регулярное выражение. Это имеет смысл и должно использоваться только если **mode** имеет значение 0 или **FALSE**.

Если параметр **Условия поиска** = и <> должны распространяться на всю ячейку в

диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления** — установлен, **lookurvalue** должен соответствовать всему тексту в ячейке; в противном случае он может соответствовать только части текста.

Соответствие всегда не зависит от регистра — значение параметра **Учитывать регистр** в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, не употребляется.

	A	B	C	D	E	F
1	3	5	cat	mat	matter	
2	A	B	C	D	E	
3						
4		A				=HLOOKUP(3; A1:E2; 2; 0)
5						
6		#N/A				=HLOOKUP(4; A1:E2; 2; 0)
7						
8		C				=HLOOKUP("CAT"; A1:E2; 2; 0)
9						
10		C				=HLOOKUP("c.t"; A1:E2; 2; 0)
11						
12		C				=HLOOKUP(".at"; A1:E2; 2; 0)
13						
14		#N/A				=HLOOKUP("at"; A1:E2; 2; 0)
15						

В ячейке **B4** возвращается **A**. Ячейка **A1** в первой строке соответствует 3. Содержимое соответствующей ячейки во 2-ой строке (то есть, **A2**) возвращается. Первый ряд может быть отсортирован, но это не является необходимым в режиме 0.

В ячейке **B6** возвращается значение ошибки **#N/A**. Нет никакого соответствия для 4.

В ячейке **B8** возвращается **C**. Различие между нижним и верхним регистрами игнорируется, таким образом **CAT** соответствует **cat**.

В ячейке **B10** возвращается **C**, если регулярные выражения разрешены (и **#N/A** если не разрешены). Точка "." соответствует "любому одиночному символу" в регулярных выражениях, таким образом **c.t** соответствует **cat**.

В ячейке **B12** возвращается **C**, если регулярные выражения разрешены. Заданному условию поиска соответствует **cat**, а так же **mat**, но всегда находится крайнее левое совпадение в режиме 0.

В ячейке **B14** возвращается значение ошибки **#N/A**, если параметр **Условия поиска = и <>** должны распространяться на всю ячейку — установлен и **C** в противном случае. Строка **at** не соответствует никакому целому содержимому ячеек, но оно соответствует части **cat**.

	A	B	C	D	E	F
1	3	5	cat	mat	matter	
2	A	B	C	D	E	
3						
4		A				=HLOOKUP(3; A1:E2; 2; 1)
5						
6		A				=HLOOKUP(4; A1:E2; 2; 1)
7						
8		#N/A				=HLOOKUP(1; A1:E2; 2; 1)
9						
10		B				=HLOOKUP(7; A1:E2; 2; 1)
11						
12		B				=HLOOKUP("apple"; A1:E2; 2; 1)
13						
14		D				=HLOOKUP("mate"; A1:E2; 2; 1)
15						
16		E				=HLOOKUP("rat"; A1:E2; 2; 1)
17						
18		B				=HLOOKUP("at"; A1:E2; 2; 1)
19						

В ячейке **B4** возвращается **A**. Ячейка **A1** в первой строке соответствует 3. Верхняя строка должна быть отсортирована в режиме 1, с текстовыми данными, располагающимися после чисел. Текстовая строка **matter** следует за **mat**, потому что, хотя первые буквы — одинаковые, **matter** имеет больше букв.

В ячейке **B6** возвращается **A**. 4 располагалось бы между 3 и 5 в верхней строке, таким образом находится левая позиция.

В ячейке **B8** возвращается значение ошибки **#N/A**, 1 располагалось бы перед 3 в верхнем ряду, но позиции слева от нее нет в таблице.

В ячейке **B10** возвращается **B**. 7 располагалось бы между 5 и **cat** в верхнем ряду, таким образом находится левая позиция.

В ячейке **B12** возвращается **B**. Текстовая строка **apple** располагалась бы между 5 и **cat** в верхнем ряду, таким образом находится левая позиция.

В ячейке **B14** возвращается **D**. Текстовая строка **mate** располагалась бы между **mat** и **matter** в верхнем ряду, таким образом находится левая позиция.

В ячейке **B16** возвращается **E**. Текстовая строка **rat** располагалась бы после **matter** в верхнем ряду, таким образом находится левая позиция.

В ячейке **B18** возвращается **B**, даже если параметр **Условия поиска = и <>** должны распространяться на всю ячейку — не установлен. В этом случае соответствие между **at** и **cat** игнорируется в пользу рассмотрения **at** как слова, которое располагается между 5 и **cat**.

HYPERLINK

Функция **HYPERLINK** заставляет ячейку открывать гиперссылку (в другом приложении) когда на ней выполняется щелчок кнопкой мыши.

Синтаксис функции:

=HYPERLINK(linkaddress; celltext)

где:

- **linkaddress** — адрес (в виде текста), который открывается;
- **celltext** — текст, отображаемый в ячейке, если опущен, **linkaddress** отображается в ячейке.

Ячейка, содержащая функцию **HYPERLINK** откроет гиперссылку, когда по ячейке выполняется щелчок кнопкой мыши.

Чтобы отредактировать ячейку, содержащую гиперссылку, выберите соседнюю ячейку, и переместитесь к ячейке, содержащей гиперссылку, используя клавиши-стрелки клавиатуры (если Вы щелкнете непосредственно на ячейке, то это откроет гиперссылку).

Если ячейка содержит функцию **HYPERLINK** (даже если функция **HYPERLINK** не вычислена):

1. эта ячейка становится ячейкой, содержащей гиперссылку; то есть, если Вы нажмете на нее, то она попытается открыть страницу в окне браузера.

Например, в ячейке с

=IF(TRUE(); “некоторый текст”; HYPERLINK(“http://www.example.com”))

функция **HYPERLINK** никогда не вычисляется — но однако ячейка становится ячейкой, содержащей гиперссылку (хотя связь не будет устанавливаться — см. далее).

2. формула в ячейке оценена дважды.

- i) В первый раз, любая функция **HYPERLINK** возвращает **linkaddress**, и

результат формулы становится адресом для ссылки.

- ii) Во второй раз, любая функция **HYPERLINK** возвращает **celltext** (или **linkaddress**, если **celltext** не задан), и результат формулы отображается в ячейке.

Например,

= "h" & HYPERLINK("ttp://";"ot ") & HYPERLINK("www.example.com";"link")

вычисляется дважды. В первый раз "h" & "ttp://" & "www.example.com" соединяются, чтобы сформировать "http://www.example.com" — адрес для ссылки. Второй раз "h" & "ot " & "link" соединяются, чтобы сформировать "горячую связь" - текст, который появляется в ячейке.

= "h" & HYPERLINK("ttp://"; "вот - ") & HYPERLINK("www.example.com")

вычисляется сначала, чтобы объединить "h" & "ttp://" & "www.example.com", сформировав "http://www.example.com" — адрес для ссылки, и вторично объединяя "h" & "вот - " & "www.example.com", формируя "вот - www.example.com" — текст, который появляется в ячейке.

Примеры, представленные выше, иллюстрируют, как **HYPERLINK** работает — но не, как она должна использоваться.

	A	B	C	D	E	F	
1							
2		http://www.example.org	=HYPERLINK("http://www.example.org")				
3							
4		Нажмите здесь	=HYPERLINK("http://www.example.org"; "Нажмите здесь")				
5							
6		Нажмите example.org	=HYPERLINK("http://www."; "Нажмите ") & "example.org"				
7							

В ячейке **B2** отображается **http://www.example.org** и открывается гиперссылка **http://www.example.org**, когда по ячейке выполняется щелчок кнопкой мыши.

В ячейке **B4** отображается **Нажмите здесь** и открывается гиперссылка **http://www.example.org**, когда по ячейке выполняется щелчок кнопкой мыши.

В ячейке **B6** отображается **Нажмите example.org** и открывается гиперссылка **http://www.example.org**, когда по ячейке выполняется щелчок кнопкой мыши.

INDEX

Функция **INDEX** возвращает значение из таблицы, задаваемое номером столбца и строки.

Синтаксис функции:

=INDEX(datatable; row; column; areanumber)

где:

- **datatable** — диапазон или массив;
- **row** — позволяет выбрать в **datatable** строку, из которой будет возвращаться значение;
- **column** — позволяет выбрать в **datatable** столбец, из которого будет возвращаться значение;
- **areanumber** — если **datatable** имеет несколько диапазонов, определяет, какой должен использоваться. Является необязательным и по умолчанию — 1.

Функция **INDEX** возвращает значение в позиции **row**, **column** в диапазоне или массиве **datatable**. Эта функция — по существу двумерная версия функции **CHOOSE**.

Если **row** опущен или пустой параметр (две последовательные точки с запятой ;;), или 0, возвращается весь столбец заданной области **areanumber** в **datatable**. Если **column** опущен или пустой параметр (две последовательные точки с запятой ;;), или 0, возвращается вся строка заданной области **areanumber** в **datatable**. Если и **row** и **column**, опущены, пустые или 0, возвращается вся заданная область **areanumber**.

Если **datatable** — одномерный вектор столбец, **column** является необязательным параметром или может быть опущен как пустой параметр (две последовательные точки с запятой ;;).

Номер строки и столбца указываются относительно верхнего левого угла ссылки на диапазон.

Если **row** или **column** имеют значение, большее чем соответствующее измерение заданной области **areanumber**, возвращается ошибка.

	A	B	C	D	E
1		5	ИСТИНА		
2		Красный	3		
3		6	Синий		
4					
5					
6					
7					
8					
9					
10					
11					
12					

5	=INDEX(B1:C3; 1; 1)
Синий	=INDEX(B1:C3; 3; 2)
Красный	=INDEX(B1:B3; 2)
ИСТИНА	=INDEX(B1:C2; 1; 2)

Calc не может обращаться с пустыми параметрами, так например **INDEX(multirange; 3; ; 2)** будет возвращать ошибку.

Если **datatable** имеет одну строку, стандарт ODFF подразумевает что может использоваться синтаксис **INDEX(datatable; column)**. Это должно быть реализовано в Calc.

INDIRECT

Функция **INDIRECT** возвращает ссылку, заданную текстовой строкой.

Синтаксис функции:

=INDIRECT(textref)

где:

- **textref** — текстовая строка, определяющая ссылку, которая будет возвращена; ссылка может быть на единственную ячейку или на диапазон.

Ссылка может использоваться в формулах или функциях, требующих ссылку - смотрите примеры ниже.

	A	B	C	D	E
1					
2		10			
3					
4					
5					
6					
7					

10	=INDIRECT("B2")
10	=INDIRECT("B" & 2)

В ячейке **B4** возвращается содержимое ячейки **B2**. **INDIRECT("B2")** возвращает ссылку на ячейку **B2**, таким образом это то же самое, что **=B2**.

LOOKUP

Функция **LOOKUP** возвращает значение из таблицы шириной в одну ячейку, в позиции, определенной поиском по другой таблице.

Синтаксис функции:

=LOOKUP(lookupvalue; searchtable; resulttable)

где:

- **lookupvalue** — значение (число, текст или логическое значение), которое ищется в **searchtable**;
- **searchtable** - однострочный или одностолбцовый диапазон/массив, должен быть отсортирован, с числами, в порядке возрастания появляющимися перед текстовыми значениями в алфавитном порядке;
- **resulttable** — диапазон/массив из которого возвращается значение, того же размера, что и **searchtable**.

Функция **LOOKUP** возвращает значение из **resulttable** в позиции, где было найдено соответствие в **searchtable**. **LOOKUP** решает, где в **searchtable** появляется **lookupvalue**. Если есть точное соответствие, оно является найденной позицией; если есть больше чем одно точное соответствие, найденная позиция — не обязательно крайняя левая/самая верхняя. Если нет точного соответствия, находится позиция как раз перед тем местом, где **lookupvalue** появился бы; возвращается ошибка **#N/A**, если это положение находится не в **searchtable**.

Соответствие всегда нечувствительно к регистру — значение параметра **Учитывать регистр** в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления** не применяется.

Если регулярные выражения разрешены в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, **LOOKUP** найдет точные соответствие, рассматривая **lookupvalue** как регулярное выражение. Это имеет смысл и должно использоваться только если Вы ожидаете уникальное точное соответствие.

Если параметр **Условия поиска = и <>** должны распространяться на всю ячейку в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления** установлен, **lookupvalue** должен соответствовать всему тексту в ячейке; в противном случае он может соответствовать только части текста.

	A	B	C	D	E	F
1	3	5	cat	mat	matter	
2						
3	A	B	C	D	E	
4						
5		A	=LOOKUP(3; A1:E1; A3:E3)			
6		B	=LOOKUP(7; A1:E1; A3:E3)			
7						
8						
9		#N/A	=LOOKUP(1; A1:E1; A3:E3)			
10						
11		C	=LOOKUP("CAT"; A1:E1; A3:E3)			
12						
13		D	=LOOKUP("mate"; A1:E1; A3:E3)			
14						
15		C	=LOOKUP("c.t"; A1:E1; A3:E3)			
16						
17		D	=LOOKUP(".at"; A1:E1; A3:E3)			
18						

В ячейке **B5** возвращается **A**. Ячейка **A1** соответствует **3**. Возвращается содержимое соответствующей ячейки из диапазона **A3:E3**.

В ячейке **B7** возвращается **B. 7** появилось бы между **5** и **cat**, таким образом находится левая позиция.

В ячейке **B9** возвращается **#N/A. 1** появилось бы перед **3** в верхней строке, однако нет позиции слева от нее в таблице.

В ячейке **B11** возвращается **C**. Различие между нижним и верхним регистрами игнорируется, таким образом **CAT** соответствует **cat**.

В ячейке **B13** возвращается **D**. Текстовое значение **mate** появилось бы между **mat** и **matter**, таким образом находится левая позиция.

В ячейке **B15** возвращается **C**, если регулярные выражения разрешены (и **B** если не разрешены). Точка '.' поддерживает 'любой одиночный символ' в регулярном выражении, таким образом **c.t** соответствует **cat**.

В ячейке **B17** регулярное выражение **“.at”** соответствует и **cat** и **mat**, если регулярные выражения разрешены. Оно также соответствует **matter**, если параметр **Условия поиска = и <>** должны распространяться на всю ячейку выключен.

В Excel, любые логические значения в **searchtable** должны появляться после любых текстовых значений. В Calc, логические значения, как предполагается, имеют числовые значения **0 (FALSE)** и **1 (TRUE)**.

Гибкость и мощность регулярных выражений и запрещение параметра соответствовать всей ячейке, делают эту функцию намного более сложной. Самый безопасный и самый легкий способ использовать эту функцию состоит в том, чтобы запретить регулярные выражения и разрешить соответствие всей ячейке.

MATCH

Функция **MATCH** возвращает позицию найденного элемента в одностроковой или одностолбцовой таблице.

Синтаксис функции:

=MATCH(searchitem; searchregion; matchtype)

где:

- **searchitem** — разыскиваемое значение;
- **searchregion** — одностроковый или одностолбцовый диапазон, в котором производится поиск;
- **matchtype** — необязательный параметр, определяющий тип поиска.

Если **matchtype** = 1 или опущен, предполагается, что **searchregion** отсортирован в порядке возрастания, с меньшими числами перед большими, меньшими текстовыми значениями перед большими (например, “A” перед “B”, а “B” перед “BA”), и **FALSE** перед **TRUE**. **MATCH** в этом случае возвращает положение наибольшего значения в **searchregion**, который меньше или равно **searchitem**.

Если **matchtype** = -1, предполагается, что **searchregion** отсортирован в порядке убывания. **MATCH** в этом случае возвращает положение наименьшего значения в **searchregion**, которое больше или равно **searchitem**.

Если **matchtype** = 0, **MATCH** возвращает положение первого точного соответствия. **searchregion** может быть неотсортирован. В этом случае **searchitem** может содержать регулярное выражение, которое соответствующим образом интерпретируется, если регулярные выражения разрешены в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**.

Если ничего не найдено, **MATCH** возвращает ошибку **#N/A**.

	A	B	C	D	E	F	G
1		3					
2		4		2	=MATCH(5; B1:B4; 1)		
3		8					
4		9					
5							

В ячейке **D5** возвращается **2**. Второе значение в диапазоне **B1:B4** — **4**, оно является наибольшим значением, которое меньше или равно **5**.

	A	B	C	D	E	F	G
1		9					
2		8		2	=MATCH(5; B1:B4; -1)		
3		4					
4		3					
5							

В ячейке **D5** возвращается **2**. Второе значение в диапазоне **B1:B4** — **8**, оно является наименьшим значением, которое больше или равно **5**.

	A	B	C	D	E	F	G
1		4					
2		7		3	=MATCH(5; B1:B4; 0)		
3		5					
4		2					
5							

В ячейке **D5** возвращается **3**, поскольку значение **5** — третий элемент в диапазоне **B1:B4**.

	A	B	C	D	E	F	G
1		blue					
2		red		2	=MATCH("r.d"; B1:B4; 0)		
3		green					
4		pink					
5							

В ячейке **D5** возвращается **2** (если регулярные выражения разрешены). Регулярное выражение "**r.d**" соответствует **red**. Если регулярные выражения запрещены, возвращается ошибка **#N/A**.

MATCH не может принять множество в качестве второго параметра, как требует разрабатываемый международный стандарт ODF. Например **MATCH(2;{1;2;3};0)** возвратит ошибку **Err:504**.

OFFSET

Функция **OFFSET** возвращает измененную ссылку, задаваемую ссылкой, смещением и требуемым размером.

Синтаксис функции:

=OFFSET(reference; row_offset; col_offset; new_height; new_width)

где:

- **reference** — заданная ссылка, которая может быть диапазоном;
- **row_offset** — смещение относительно заданной ссылки вниз, допустимы отрицательные числа;
- **col_offset** — смещение относительно заданной ссылки вправо, допустимы отрицательные числа;

- **new_height** — устанавливают высоту новой ссылки на диапазон, необязательный параметр, по умолчанию берется высота исходной ссылки;
- **new_width** — устанавливают ширину новой ссылки на диапазон, необязательный параметр, по умолчанию берется ширина исходной ссылки.

Измененная ссылка должна быть в действительном диапазоне, начинающемся со столбца/строки не превышающих максимальных столбца/строки. Если задаются новая ширина или высота диапазона, они должны быть больше нуля.

	A	B	C	D	E
1		Ячейка B1			
2			Ячейка C2		
3					
4		Ячейка C2	=OFFSET(A1; 1; 2)		
5					
6		Ячейка B1	=OFFSET(B3; -2; 0; 1; 1)		
7					

В ячейке **B4** возвращается содержимое ячейки **C2**. Параметры **new_height** и **new_width** были опущены.

В ячейке **B6** возвращается содержимое ячейки **B1**. Допускаются отрицательное смещения.

	A	B	C	D	E	F
1			7	8	9	
2						
3			1	2	3	
4			4	5	6	
5						
6		21	=SUM(OFFSET(A1; 2; 2; 2; 3))			
7						
8		24	=SUM(OFFSET(C1:G1; 0; 0; 1; 3))			
9						

В ячейке **B6** возвращается сумма содержимого ячеек диапазона **C3:E4**. 2 строки и 2 столбца добавляются к ссылке на ячейку **A1**, чтобы сместить ее на **C3**, а затем новый размер расширяет диапазон до **E4**.

В ячейке **B8** возвращается сумма содержимого ячеек диапазона **C1:E1**.

Параметр **new_height** не может быть пустым.

OFFSET(A1;{0|1};0;1;1) в формуле массива возвращает неожиданные результаты.

ROW

Функция **ROW** возвращает номер(а) строки, задаваемый ссылкой.

Синтаксис функции:

=ROW(reference)

где:

- **reference** — ссылка на ячейку или диапазон, может задаваться любой явно (например, **A3:B5**), либо по имени (например, **myref**).

Функция **ROW** возвращает номер строки ссылки. Если ссылка опущена, возвращается номер строки текущей ячейки (содержащей формулу). Если ссылка — диапазон ячеек, а формула введена как формула массива (нажатием *Ctrl-Shift-Enter*, а не *Enter*), возвращается однострочный массив номеров строк.

	A	B	C	D	E
1					
2		3	=ROW(B3)		
3					
4		4	=ROW()		
5					
6		5	=ROW(D5:D8)		
7					
8		5	{=ROW(D5:D8)}		
9		6			
10		7			
11		8			
12					

В ячейке **B2** возвращается **3**. Ячейка **B3** находится в третьей строке.

В ячейке **B4** возвращается **4**. Формула введена в ячейке **B4**, которая находится в четвертой строке.

В ячейке **B6** возвращается **5**. Это не формула массива, таким образом возвращается номер строки первой ячейки диапазона **D5:D8**, т. е. ячейки **D5**.

В ячейках **B8:B11** возвращается **5, 6, 7 и 8**. Возвращается одностолбцовый массив **{5|6|7|8}**, поскольку это формула массива.

ROWS

Функция **ROWS** возвращает количество строк в заданной ссылке.

Синтаксис функции:

=ROWS(reference)

где:

- **reference** — ссылка на ячейку или диапазон, может задаваться либо явно (например, **A3:B5**), либо по имени (например, **myref**).

Возможно определить имя для представления более чем одного диапазона (например, введя **\$Лист1.\$A\$1:\$A\$5~\$Лист1.\$C\$1:\$C\$5** в поле **Назначен на** диалогового окна **Вставка — Названия — Определить...**). В этом случае **ROWS** суммируют количество строк в каждом диапазоне, независимо от любого дублирования строк.

	A	B	C	D	E
1					
2		3	=ROWS(A3:B5)		
3					
4		3	=ROWS(G4:G6)		
5					
6		1	=ROWS(D8)		
7					
8		7	=ROWS(myref)		
9					

В ячейке **B2** возвращается **3**, есть три строки в диапазоне **A3:B5**, а именно, строки **3, 4 и 5**.

В ячейке **B4** возвращается **3**, есть три строки в диапазоне **G4:G6**, а именно, строки **4, 5 и 6**.

В ячейке **B6** возвращается **1**, **D8** интерпретируется как диапазон, состоящий из одной ячейки, с одной строкой (строка 8).

В ячейке **B8** возвращается **7**, именованный диапазон **myref** был определен как диапазон, состоящий из нескольких областей, **\$A\$1:\$A\$3~\$C\$1:\$D\$4**.

В OOo3 функция **ROWS** с диапазонами, состоящими из нескольких областей, возвращает значение ошибки **Err:504**.

SHEET

Функция **SHEET** возвращает номер листа, задаваемого ссылкой.

Синтаксис функции:

=SHEET(reference)

где:

- **reference** — ссылка на ячейку.

Функция **SHEET** возвращает номер листа, задаваемого ссылкой. Если ссылка опущена, возвращается номер текущего листа (содержащий формулу).

	A	B	C	D	E
1					
2		2	=SHEET(Лист2.C4)		
3					
4		1	=SHEET()		
5					

В ячейке **B2** возвращается **2**, если Вы имеете 3 листа — **Лист1**, **Лист2**, **Лист3** - в этом порядке, то **Лист2** — второй лист.

В ячейке **B4** возвращается **1**, номер текущего листа (листа, содержащего формулу).

SHEETS

Функция **SHEETS** возвращает количество листов в заданной ссылке.

Синтаксис функции:

=SHEETS(reference)

где:

- **reference** — ссылка на ячейку, может задаваться либо явно (например, **Лист1.A1:Лист3.G12**), либо по имени (например, **myref**).

Если именованная ссылка содержит несколько диапазонов, возвращается сумма листов в каждом диапазоне, независимо от любого дублирования.

	A	B	C	D	E
1					
2		3	=SHEETS(Лист1.A1:Лист3.G12)		
3					
4		1	=SHEETS(G4:H6)		
5					
6		1	=SHEETS(D8)		
7					

В ячейке **B2** возвращается **3**, если **Лист3** следует за **Лист2**, который следует за **Лист1**; в этом диапазоне, таким образом, есть три листа.

В ячейке **B4** возвращается **1**; **G4:H6** интерпретируется как диапазон на одном (текущем) листе.

В ячейке **B6** возвращается **1**; **D8** интерпретируется как диапазон, состоящий из одной ячейки, на одном (текущем) листе.

В ООоЗ функция **SHEETS** с диапазонами, состоящими из нескольких областей, возвращает значение ошибки **Err:504**.

STYLE

Функция **STYLE** применяет стиль (например цвет) к ячейке.

Синтаксис функции:

=STYLE(style; timesecs; finalstyle)

где:

- **style** — начальный стиль, применяемый к ячейке;
- **timesecs** — временной интервал в секундах, в течение которого применяется стиль **style**;
- **finalstyle** — заключительный стиль, применяемый к ячейке.

Функция **STYLE** применяет начальный стиль **style** к ячейке в течение **timesecs** секунд, и после этого применяет заключительный стиль **finalstyle**. Оба стиля задаются как текст. Стили содержатся (и могут быть созданы) в меню **Формат - Стили**.

Начальный стиль применяется в течение **timesecs** секунд, после чего ячейка вычисляется повторно. Пожалуйста отметьте, что ручной перерасчет (клавиша **F9** или **Сервис — Содержимое ячейки — Пересчитать**), не вызывает применение начального стиля.

Параметры **timesecs** и **finalstyle** могут быть опущены вместе; тогда постоянно применяется стиль **style**.

Функция **STYLE** возвращает числовое значение 0, таким образом она может прибавляться к существующему расчету в ячейке, не изменяя его результат. См. примеры ниже.

Другой способ применения стилей может быть найден в меню **Формат — Условное Форматирование...**

Не существует никакого прямого способа определить (в формуле), какое форматирование было применено к ячейке.

	A	B	C	D	E	F	G
1							
2		8					
3							
4		3					
5							
6	5	5					
7							
8	2	2					
9							
10		Некоторый текст					
11							
12		4					
13							

В ячейке **B2** отображается **8** (результат сложения 3 и 5) на красном фоне в течение 3 секунд, и со стилем Базовый после этого.

В ячейке **B4** отображается **3** (квадратный корень 9) постоянно на красном фоне.

В ячейке **B6** отображается число, содержащееся в ячейке **A6**, на красном фоне, если **A6** больше чем **3**. Здесь, функция **IF** возвращает имя стиля для использования в функции **STYLE** (“Красный” или “Базовый”), в зависимости от значения ячейки **A6**.

В ячейке **B8** отображается число, содержащееся в ячейке **A8** на красном фоне, если **A8** больше чем 3.

В ячейке **B10** отображается текст **Некоторый текст** на красном фоне. Функция **STYLE** всегда возвращает 0, таким образом **T(0)** возвратит строку нулевой длины, которая может быть объединена с существующей строкой, не затрагивая ее.

В ячейке **B10** отображается результат вычисления формулы **SQRT(16)** на красном фоне. Этот вариант может быть полезен, если Вы не можете предсказать, возвратит ли формула текст или число (например с **HLOOKUP**).

Функция **STYLE** уникальна для Calc.

VLOOKUP

Функция **VLOOKUP** возвращает значение из столбца таблицы, в строке, удовлетворяющей условиям поиска в первом столбце.

Синтаксис функции:

=VLOOKUP(lookupvalue; datatable; columnindex; mode)

где:

- **lookupvalue** — значение (число, текст или логическое значение), разыскиваемое в первом столбце диапазона/массива. Когда значение найдено в первом столбце, **VLOOKUP** возвращает соответствующее значение (в той же самой строке) в столбце **columnindex** диапазона **datatable**, где для первой строки **columnindex** = 1.
- **datatable** — ссылка, которая должна включать по крайней мере два столбца
- **columnindex** — номер столбца в массиве, содержащего значения, которые должны быть возвращены. Первый столбец имеет номер 1.
- **mode** — необязательный параметр, который указывает, отсортирован ли первый столбец в массиве в порядке возрастания. Если **mode** = 0 или **FALSE**, первый столбец не отсортирован в порядке возрастания, и ищется первое точное совпадение (поиск выполняется сверху вниз). В неотсортированных списках, значение поиска должно быть подобрано точно. Иначе функция возвратит сообщение об ошибке: **#N/A**. Если **mode** = 1, **TRUE** или опущен, первый столбец **datatable** должен быть отсортирован, с числами, в порядке возрастания располагающимися перед текстовыми значениями в алфавитном порядке. Поиск по отсортированным столбцам может выполняться намного быстрее, и функция всегда возвращает значение, даже если значение поиска не было точно подобрано, если оно находится между наименьшим и наибольшим значением отсортированного списка. **VLOOKUP** решает, где в первом столбце появляется **lookupvalue**. Если есть точное совпадение, оно соответствует найденной строке; если есть более чем одно точное совпадение, найденная строка — не обязательно самая верхняя.

Если регулярные выражения разрешены в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, **VLOOKUP** находит точное соответствие, рассматривая **lookupvalue** как регулярное выражение. Это имеет смысл и должно использоваться только если **mode** имеет значение 0 или **FALSE**.

Если параметр **Условия поиска** = и <> должны распространяться на всю ячейку в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления** — установлен, **lookupvalue** должно соответствовать всему тексту в ячейке; в противном случае он может соответствовать только части текста.

Соответствие всегда не зависит от регистра — значение параметра **Учитывать регистр** в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, не

употребляется.

	A	B	C	D	E	F	G	H
1	3	A						
2	5	B		A	=VLOOKUP(3; A1:B5; 2; 0)			
3	cat	C						
4	mat	D		#N/A	=VLOOKUP(4; A1:B5; 2; 0)			
5	matter	E						
6				C	=VLOOKUP("CAT"; A1:B5; 2; 0)			
7								
8				C	=VLOOKUP("c.t"; A1:B5; 2; 0)			
9								
10				C	=VLOOKUP(".at"; A1:B5; 2; 0)			
11								
12				#N/A	=VLOOKUP("at"; A1:B5; 2; 0)			
13								

В ячейке **D2** возвращается **A**. Ячейка **A1** в левом столбце соответствует **3**. Возвращается содержимое соответствующей ячейки в 2-ом столбце (то есть, **B1**). Левый столбец может быть отсортирован, но это не требуется в **mode = 0**.

В ячейке **D4** возвращается ошибка **#N/A**. Нет соответствия для 4.

В ячейке **D6** возвращается **C**. Различие между нижним и верхним регистрами игнорируется, таким образом **CAT** соответствует **cat**.

В ячейке **D8** возвращается **C**, если регулярные выражения разрешены (и значение ошибки **#N/A** если запрещены). Точка “.” соответствует “любому одиночному символу” в регулярном выражении, таким образом **c.t** соответствует **cat**.

В ячейке **D10** возвращается **C**, если регулярные выражения разрешены. Заданному условию поиска соответствует **cat**, а так же **mat**, но всегда находится самое верхнее совпадение при **mode = 0**.

В ячейке **D10** возвращается значение ошибки **#N/A**, если параметр **Условия поиска = и** <> должны распространяться на всю ячейку — установлен и **C** в противном случае. Строка **at** не соответствует никакому целому содержимому ячеек, но оно соответствует части **cat**.

	A	B	C	D	E	F	G	H
1	3	A						
2	5	B		A	=VLOOKUP(3; A1:B5; 2; 1)			
3	cat	C						
4	mat	D		A	=VLOOKUP(4; A1:B5; 2; 1)			
5	matter	E						
6				#N/A	=VLOOKUP(1; A1:B5; 2; 1)			
7								
8				B	=VLOOKUP(7; A1:B5; 2; 1)			
9								
10				B	=VLOOKUP("apple"; A1:B5; 2; 1)			
11								
12				D	=VLOOKUP("mate"; A1:B5; 2; 1)			
13								
14				E	=VLOOKUP("rat"; A1:B5; 2; 1)			
15								
16				B	=VLOOKUP("at"; A1:B5; 2; 1)			
17								

В ячейке **D2** возвращается **A**. Ячейка **A1** в левом столбце соответствует **3**. Левый столбец должен быть отсортирован при **mode = 1**, с текстовыми данными, располагающимися после чисел. Текстовая строка **matter** следует за **mat**, потому что, хотя первые буквы — одинаковые, **matter** имеет больше букв.

В ячейке **D4** возвращается **A**. **4** появилось бы между **3** и **5** в левом столбце, таким образом находится верхняя позиция.

В ячейке **D6** возвращается ошибка **#N/A**. **1** появилась бы перед **3** в левом столбце, но позиция выше нее в таблице отсутствует.

В ячейке **D8** возвращается **B**. **7** появился бы между **5** и **cat** в левом столбце, таким образом находится верхняя позиция.

В ячейке **D10** возвращается **B**. Текстовое значение **apple** появилось бы между **5** и **cat** в левом столбце, таким образом находится верхняя позиция.

В ячейке **D12** возвращается **D**. возвращений появился бы между циновкой и вопросом в левом столбце, таким образом положение выше найдено. Текстовое значение **mate** появилось бы между **mat** и **matter** в левом столбце, таким образом находится верхняя позиция.

В ячейке **D14** возвращается **E**. Текстовое значение **rat** появилась бы после **matter** в левом столбце, таким образом находится верхняя позиция.

В ячейке **D16** возвращается **B**, даже если параметр **Условия поиска = и <>** должны **распространяться на всю ячейку** — не установлен. В этом случае соответствие между **at** и **cat** игнорируется в пользу рассмотрения **at** как слова, которое располагается между **5** и **cat**.

Вот простой пример классификации для иллюстрирования того, как **VLOOKUP** может сделать для Вас жизнь легче. В формуле, последний аргумент **1 (TRUE)** говорит OpenOffice Calc, что столбец поиска упорядоченный, а точное соответствие не требуется. Возвращается самое близкое соответствие, которое не превышает значение поиска — это именно то, чего мы хотим..

	A	B	C	D	E	F
1	Нижний предел	Оценка				
2	0	F				
3	50	E				
4	60	D				
5	70	C				
6	80	B				
7	90	A				
8						
9	Студент	Очки	Оценка			
10	Чарли	73	C	=VLOOKUP(B10;\$A\$2:\$B\$7;2;1)		
11	Лиза	64	D	=VLOOKUP(B11;\$A\$2:\$B\$7;2;1)		
12	Мишель	92	A	=VLOOKUP(B12;\$A\$2:\$B\$7;2;1)		
13	Гарри	44	F	=VLOOKUP(B13;\$A\$2:\$B\$7;2;1)		
14	Роджер	55	E	=VLOOKUP(B14;\$A\$2:\$B\$7;2;1)		
15	Патрик	80	B	=VLOOKUP(B15;\$A\$2:\$B\$7;2;1)		
16						

Другое полезное применение функции **VLOOKUP** — использование OpenOffice Calc для выписки счетов. На одном листе мы имеем каталог — где мы можем хранить цены и ассортимент товаров в одном месте.

	A	B	C	D	E
1	Каталог				
2	Код Товара	Описание	Цена		
3	0001	Болт 1/4"	1,50		
4	0002	Болт 1/3"	1,80		
5	0003	Шестигранная гайка 1"	1,30		
6	0004	Шестигранная гайка 1/2"	1,15		
7					
8					

Лист1 | **Catalog** | Лист3

Функция **VLOOKUP** вступает в игру на листе счета — как проиллюстрировано ниже. Здесь, таблица соответствий (каталог) находится на другом листе.

	A	B	C	D	E	F
2	Код по каталогу	Кол-во	Описание	Цена за штуку	Сумма	
3	0003	5	Шестигранная гайка 1"	1,30	6,50	=B3*D3
4	0004	12	Шестигранная гайка 1/2"	1,15	13,80	=B4*D4
5						
6			Итого		20,30	=SUM(E3:E5)
7						
8						
9		C3	=VLOOKUP(A3;\$Catalog.\$A\$3:\$C\$6;2)			
10		C4	=VLOOKUP(A4;\$Catalog.\$A\$3:\$C\$6;2)			
11						
12		D3	=VLOOKUP(A3;\$Catalog.\$A\$3:\$C\$6;3)			
13		D4	=VLOOKUP(A4;\$Catalog.\$A\$3:\$C\$6;3)			
14						

Вот другое простое применение функции **VLOOKUP**. Рассмотрим торгового агента, которому платят комиссионные от продаж на основе скользящей шкалы.

	A	B	C	D	E	F	G
1		VLOOKUP					
2		Справочная таблица					
3		Продажи	Ставка				
4		0	25,00%				
5		50	22,50%				
6		100	20,00%				
7		500	18,00%				
8		1000	15,00%				
9							
10		Продажи	Комиссионные				
11		350	70 р.	=VLOOKUP(B11;\$B\$4:\$C\$8;2)*B11			
12		360	72 р.	=VLOOKUP(B12;\$B\$4:\$C\$8;2)*B12			
13		700	126 р.	=VLOOKUP(B13;\$B\$4:\$C\$8;2)*B13			
14							

При продажах до 50 \$, платится комиссионные в 25 %, между 50 \$ и 100 \$, комиссионные уменьшаются до 22.5 % — и так далее. При использовании таблицы соответствий и функции **VLOOKUP**, правильная ставка комиссионного вознаграждения может быть применена к каждой сделке.

Функции преобразования чисел

Функции преобразования чисел выполняют преобразования чисел между различными представлениями чисел. Двоичные, восьмеричные и шестнадцатеричные числа могут быть представлены в виде текста или числами, с использованием только обычных цифр (0-1 для двоичных, 0-7 для восьмеричных); это только представление, и арифметика (например сложение) работать не будет. Функции Calc возвращают текст для двоичных, восьмеричных и шестнадцатеричных чисел.

ARABIC

Функция **ARABIC** возвращает арабское число (например, 14), соответствующее заданному римскому числу (например, XIV).

Синтаксис функции:

=ARABIC(text)

где:

- **text** — римское число в виде текста, ограниченное диапазоном 1—3999.

Функция **ARABIC** выполняет преобразование заданного в виде текста римского числа в число. Данная функция является обратной по отношению к функции **ROMAN**. Для получения информации о значении отдельных символов римских цифр см. описание функции **ROMAN**. Любой римский символ слева от большего символа (прямо или косвенно) уменьшает окончательное значение количеством символа, иначе, это увеличивает заключительное количество количеством символа. Регистр игнорируется.

Поддерживается следующее тождество: **ARABIC(ROMAN(x; any)) = x**, в том случае если **ROMAN(x; any)** — не возвращает ошибку.

	A	B	C	D	E
1					
2		1014	=ARABIC("MXIV")		
3					
4		2002	=ARABIC("MMII")		
5					
6		3999	=ARABIC("mcmxcix")		
7					

BASE

Функция **BASE** возвращает текстовое представление заданного числа в указанной системе счисления.

Синтаксис функции:

=BASE(number; radix; minlength)

где:

- **number** — заданное число (положительное целое число);
- **radix** — основание системы счисления (целое число от 2 до 36 включительно);
- **minlength** — (необязательный) определяет минимальное число возвращаемых символов; нули добавляются слева в случае необходимости.

Функция **BASE** выполняет преобразование заданного числа **number** в соответствующее текстовое представление в заданной системе счисления. Используются символы 0-9, а также A-Z (в верхнем регистре).

Если параметр **minlength** не указан, возвращаемый текст использует наименьшее количество символов (то есть, ведущие нули не добавляются). Если **minlength** указан, и получающийся текст меньше, чем **minlength**, добавляются ведущие нули, чтобы вернуть текст длиной точно **minlength** символов. Если текст длиннее чем аргумент **minlength**, параметр **minlength** игнорируется.

	A	B	C	D	E
1					
2			10001	=BASE(17; 2)	
3					
4			17	=BASE(17; 10)	
5					
6			00FF	=BASE(255; 16; 4)	
7					
8			ИСТИНА	=ISTEXT(BASE(17; 10))	
9					

В ячейке **B6** возвращается **00FF** в виде текста. Добавляются ведущие нули, чтобы сформировать 4 цифры.

В ячейке **B8** возвращается **TRUE**. Функция **BASE** возвращает текст, а не число.

Эта функция в настоящее время отображается в категории Текстовые *Мастера функций*.

BIN2DEC

Функция **BIN2DEC** выполняет преобразование двоичного числа в десятичное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=BIN2DEC(binarynumber)

где:

- **binarynumber** — двоичное число, может быть в виде текста или числа, содержащего только цифры 1 и 0 (таким образом число кажется двоичным, хотя и не является им). **binarynumber** может иметь до десяти битов в дополнительном двоичном представлении; положительные числа — от 0 до 111111111 (девять бит, представляющих десятичные числа от 0 до 511) и отрицательные числа — от 111111111 до 1000000000 (десять бит, представляющих десятичные числа от -1 до -512).

Функция **BIN2DEC** возвращает (десятичное) число, соответствующее заданному двоичному числу, со старшей 10-ой цифрой, являющейся знаковым битом (используется дополнительное двоичное представление). Если задан текст, текст считается представлением двоичного числа. Если задано число, цифры числа несмотря на то, что соответствуют десятичной системе считаются цифрами эквивалентно-представленного двоичного числа. Если какая-нибудь из цифр — от 2 до 9, возвращается ошибка.

	A	B	C	D	E
1					
2		6	=BIN2DEC("110")		
3					
4		6	=BIN2DEC(110)		
5					
6		-2	=BIN2DEC("111111110")		
7					
8		Ошибка:502	=BIN2DEC(150)		
9					

Подобное использование 10-ого бита как знакового очень странно, но оно широко распространено; это реализовано в Excel2003 и OpenOffice.org 2.0.3, и почти наверняка у многих других. Таким образом **=BIN2DEC("1011100100")** возвращает отрицательное число -284 вместо предполагаемого значения 740 . Это очень спорная практика, но она чрезвычайно широко распространена.

BIN2HEX

Функция **BIN2DEC** выполняет преобразование двоичного числа в шестнадцатеричное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=BIN2HEX(binarynumber; numdigits)

где:

- **binarynumber** — двоичное число, может быть в виде текста или числа, содержащего только цифры 1 и 0 (таким образом число кажется двоичным, хотя и не является им). **binarynumber** может иметь до десяти битов в дополнительном двоичном представлении; положительные числа — от 0 до 11111111 (девять бит, представляющих десятичные числа от 0 до 511) и отрицательные числа — от 111111111 до 1000000000 (десять бит, представляющих десятичные числа от -1 до -512);
- **numdigits** — дополнительное число, определяющее количество возвращаемых цифр.

Функция **BIN2HEX** возвращает представление шестнадцатеричного числа, соответствующее заданному двоичному числу. Если задан текст, текст считается представлением двоичного числа. Если задано число, цифры числа несмотря на то, что соответствуют десятичной системе считаются цифрами эквивалентно-представленного двоичного числа. Если какая-нибудь из цифр — от 2 до 9, возвращается ошибка.

Возвращаемое значение — шестнадцатеричное значение, содержащее до 10 шестнадцатеричных цифр, самый старший бит (40-ой бит) — знаковый, и в дополнительной двоичной форме. Цифры от A до F — в верхнем регистре. Если входное значение имеет 10-ый установленный бит (отрицательное значение), аргумент **numdigits** игнорируется; в противном случае, **numdigits** указывает число цифр в возвращаемом значении, с ведущими цифрами 0, добавляемыми по мере необходимости, чтобы довести количество цифр в результате до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D	E
1					
2		1D	=BIN2HEX("11101")		
3					
4		1D	=BIN2HEX(11101)		
5					
6		01A6	=BIN2HEX("110100110"; 4)		
7					
8		FFFFFFFFFE	=BIN2HEX("111111110")		
9					

В ячейке **B2** возвращается **1D** в виде текста.

В ячейке **B4** возвращается **1D** в виде текста. Число 11101 содержит только цифры 1 и 0, и может быть прочитано как двоичное.

В ячейке **B6** возвращается **01A6** в виде текста. **BIN2HEX** добавляет ведущий ноль, чтобы сформировать 4 цифры.

В ячейке **B8** возвращается **FFFFFFFFFE** в виде текста (представление в дополнительной двоичной форме десятичного числа -2).

BIN2OCT

Функция **BIN2OCT** выполняет преобразование двоичного числа в восьмеричное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=BIN2OCT(binarynumber; numdigits)

где:

- **binarynumber** — двоичное число, может быть в виде текста или числа, содержащего только цифры 1 и 0 (таким образом число кажется двоичным, хотя и не является им). **binarynumber** может иметь до десяти битов в дополнительном двоичном представлении; положительные числа — от 0 до 111111111 (девять бит, представляющих десятичные числа от 0 до 511) и отрицательные числа — от 111111111 до 100000000 (десять бит, представляющих десятичные числа от -1 до -512).
- **numdigits** — необязательное число, определяющее количество возвращаемых цифр.

Функция **BIN2OCT** возвращает текстовое представление восьмеричного числа, соответствующее заданному двоичному числу. Если задан текст, текст считается представлением двоичного числа. Если задано число, цифры числа несмотря на то, что соответствуют десятичной системе считаются цифрами эквивалентно-представленного двоичного числа. Если какая-нибудь из цифр — от 2 до 9, возвращается ошибка.

Возвращаемое значение — восьмеричное значение, содержащее до 10 восьмеричных цифр, самый старший бит (30-ый бит) — знаковый, и в дополнительной двоичной форме. Цифры от A до F — в верхнем регистре. Если входное значение имеет 10-ый установленный бит (отрицательное значение), аргумент **numdigits** игнорируется; в противном случае, **numdigits** указывает число цифр в возвращаемом значении, с ведущими нулями, добавляемыми по мере необходимости, чтобы довести количество цифр в результате до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D	E
1					
2		14	=BIN2OCT("1100")		
3					
4		14	=BIN2OCT(1100)		
5					
6		014	=BIN2OCT("1100"; 3)		
7					
8		777777776	=BIN2OCT("111111110")		
9					

В ячейке **B2** возвращается **14** в виде текста.

В ячейке **B4** возвращается **14** в виде текста. Число 1100 содержит только цифры 1 и 0, и может быть прочитано как двоичное.

В ячейке **B6** возвращается **014** в виде текста. **BIN2HEX** добавляет ведущий ноль, чтобы сформировать 3 цифры.

В ячейке **B8** возвращается **777777776** в виде текста (представление в дополнительной двоичной форме десятичного числа -2).

DEC2BIN

Функция **DEC2BIN** выполняет преобразование десятичного числа в двоичное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=DEC2BIN(number; numdigits)

где:

- **number** — десятичное число, заданное в виде текста или числа, которое должно быть от -512 и до 511 включительно. Если задан текст, текст считается десятичным представлением числа, и может иметь ведущий знак минус;
- **numdigits** — необязательное число, определяющее количество возвращаемых цифр.

Функция **DEC2BIN** возвращает двоичное число с десятью битами, с самым старшим битом (10-ый бит) — знаковым и в дополнительном двоичном представлении; положительные числа — от 0 до 11111111 (девять битов, представляющие десятичные числа от 0 до 511) и отрицательные числа — от 11111111 до 100000000 (десять битов, представляющих десятичные числа от -1 до -512). Если **number** отрицательно, аргумент **numdigits** игнорируется; в противном случае, **numdigits** указывает число цифр в возвращаемом значении, с ведущими нулями, добавляемыми по мере необходимости, чтобы довести количество цифр в результате до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D
1				
2		1001	=DEC2BIN(9)	
3				
4		1001	=DEC2BIN("9")	
5				
6		00001001	=DEC2BIN(9; 8)	
7				
8		111111110	=DEC2BIN(-2)	
9				

В ячейке **B2** возвращается **1001** в виде текста.

В ячейке **B4** возвращается **1001** в виде текста. **DEC2BIN** принимает десятичное число, заданное в виде текста.

В ячейке **B6** возвращается **00001001** в виде текста. Добавляются ведущие нули, чтобы сформировать 8 цифр.

В ячейке **B8** возвращается **111111110** в виде текста (представление в дополнительной двоичной форме десятичного числа -2).

DEC2HEX

Функция **DEC2HEX** выполняет преобразование десятичного числа в шестнадцатеричное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=DEC2HEX(number; numdigits)

где:

- **number** — десятичное число, заданное в виде текста или числа, которое должно быть от -2^{39} до $2^{39}-1$ включительно. Если задан текст, текст считается десятичным представлением числа, и может иметь ведущий знак минус;
- **numdigits** — необязательное число, определяющее количество возвращаемых цифр.

Функция **DEC2HEX** возвращает шестнадцатеричное число, содержащее до 10 шестнадцатеричных цифр, самый старший бит (40-ой бит) — знаковый, и в дополнительной двоичной форме. Цифры от A до F — в верхнем регистре. Если входное значение — отрицательное, аргумент **numdigits** игнорируется; в противном случае, **numdigits** указывает число цифр в возвращаемом значении, с ведущими нулями, добавляемыми по мере необходимости, чтобы довести количество цифр в результате до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D
1				
2		1E	=DEC2HEX(30)	
3				
4		1E	=DEC2HEX("30")	
5				
6		001E	=DEC2HEX(30; 4)	
7				
8		FFFFFFFFFE	=DEC2HEX(-2)	
9				

В ячейке **B2** возвращается **1E** в виде текста.

В ячейке **B4** возвращается **1E** в виде текста. **DEC2HEX** принимает десятичное число, заданное в виде текста.

В ячейке **B6** возвращается **001E** в виде текста. Добавляются ведущие нули, чтобы сформировать 4 цифры.

В ячейке **B8** возвращается **FFFFFFFFFE** в виде текста (представление в дополнительной двоичной форме десятичного числа -2).

DEC2OCT

Функция **DEC2OCT** выполняет преобразование десятичного числа в восьмеричное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=DEC2OCT(number; numdigits)

где:

- **number** — десятичное число, заданное в виде текста или числа, которое должно быть от -2^{29} до $2^{29}-1$ включительно. Если задан текст, текст считается десятичным представлением числа, и может иметь ведущий знак минус;
- **numdigits** — необязательное число, определяющее количество возвращаемых цифр.

Функция **DEC2OCT** возвращает восьмеричное число, содержащее до 10 восьмеричных цифр, самый старший бит (30-ый бит) — знаковый, и в дополнительной двоичной форме. Если входное значение — отрицательное, аргумент **numdigits** игнорируется; в противном случае, **numdigits** указывает число цифр в возвращаемом значении, с ведущими нулями, добавляемыми по мере необходимости, чтобы довести количество цифр в результате до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D
1				
2		23	=DEC2OCT(19)	
3				
4		23	=DEC2OCT("19")	
5				
6		023	=DEC2OCT(19; 3)	
7				
8		777777776	=DEC2OCT(-2)	
9				

В ячейке **B2** возвращается **23** в виде текста.

В ячейке **B4** возвращается **23** в виде текста. **DEC2OCT** принимает десятичное число, заданное в виде текста.

В ячейке **B6** возвращается **023** в виде текста. Добавляются ведущие нули, чтобы сформировать 3 цифры.

В ячейке **B8** возвращается **777777776** в виде текста (представление в дополнительной двоичной форме десятичного числа -2).

DECIMAL

Возвращает десятичное число, соответствующее заданному текстовому представлению и основанию системы счисления.

Синтаксис функции:

=DECIMAL(text; radix)

где:

- **text** — текстовое представление заданного числа, буквы, если таковые вообще имеются, могут быть в верхнем или нижнем регистре. Любые ведущие пробелы и табуляции игнорируются;
- **radix** — основание системы счисления (целое число от 2 до 36 включительно).

Если основание системы счисления — 16 (шестнадцатеричная система), любые ведущие **0x**, **0X**, **x** или **X** игнорируются, как и любые замыкающие **h** или **H**.

Если основание системы счисления — 2 (двоичная система счисления), любые замыкающие **b** или **B** игнорируются.

	A	B	C	D	E
1					
2		255	=DECIMAL("00FF"; 16)		
3					
4		5	=DECIMAL("101b"; 2)		
5					

В ячейке **B2** возвращается 255 в виде числа (00FF в шестнадцатеричной системе).

В ячейке **B2** возвращается 5 в виде числа (101 в двоичной системе).

HEX2BIN

Функция **HEX2BIN** выполняет преобразование шестнадцатеричного числа в двоичное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=HEX2BIN(hexadecimalnumber; numdigits)

где:

- **hexadecimalnumber** — шестнадцатеричное число, заданное в виде текста или числа (воспринимаемое как шестнадцатеричное, хотя и не является им). Если задан текст, текст считается шестнадцатеричным представлением числа; если его 40-ой бит — 1, оно считается отрицательным числом;
- **numdigits** — необязательное число, определяющее количество возвращаемых цифр.

Возвращаемое двоичное число может иметь до десяти битов представленное в дополнительной двоичной форме; положительные числа — от 0 до 11111111 (девять битов, представляющих десятичные числа от 0 до 511) и отрицательные числа - от 11111111 до 100000000 (десять битов, представляющих десятичные числа от -1 до -512). Поэтому **hexadecimalnumber** также должен лежать в этом диапазоне, и задаваться в дополнительной двоичной форме.

Если входное значение — отрицательное (40-ой бит — 1), аргумент **numdigits** игнорируется; в противном случае, **numdigits** указывает число цифр в возвращаемом значении, с ведущими нулями, добавляемыми по мере необходимости, чтобы довести количество цифр в результате до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D	E
1					
2		11101	=HEX2BIN("1D")		
3					
4		10000	=HEX2BIN(10)		
5					
6		0010	=HEX2BIN("2"; 4)		
7					
8		111111110	=HEX2BIN("FFFFFFFFE")		
9					

В ячейке **B2** возвращается **11101** в виде текста.

В ячейке **B4** возвращается **10000** в виде текста. **HEX2BIN** воспринимает число 10 как шестнадцатеричное.

В ячейке **B6** возвращается **0010** в виде текста. **HEX2BIN** добавляет ведущие нули, чтобы сформировать 4 цифры.

В ячейке **B8** возвращается **111111110** в виде текста (представление в дополнительной

двоичной форме десятичного числа -2).

HEX2DEC

Функция **HEX2DEC** выполняет преобразование шестнадцатеричного числа в десятичное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=HEX2DEC(hexadecimalnumber)

где:

- **hexadecimalnumber** — шестнадцатеричное число, заданное в виде текста или числа (воспринимаемое как шестнадцатеричное, хотя и не является им). Если задан текст, текст считается шестнадцатеричным представлением числа; если его 40-ой бит — 1, оно считается отрицательным числом.

Функция **HEX2DEC** выполняет преобразование заданного шестнадцатеричного числа в эквивалентное десятичное. Входное значение должно содержать от 1 до 10 (включительно) шестнадцатеричных цифр. Положительные числа — от 0 до 7FFFFFFFFF (представляющие десятичные числа от 0 до $2^{39}-1$) и отрицательные числа — от FFFFFFFFFF до 8000000000 (представляющие десятичные числа от -1 до -2^{39}).

Возвращаемое значение — десятичное число.

	A	B	C	D	E
1					
2		30	=HEX2DEC("1E")		
3					
4		32	=HEX2DEC(20)		
5					
6		-2	=HEX2DEC("FFFFFFFFE")		
7					

В ячейке **B2** возвращается **30** в виде (десятичного) числа.

В ячейке **B4** возвращается **32** в виде (десятичного) числа. **HEX2DEC** воспринимает число 20 как шестнадцатеричное.

В ячейке **B6** возвращается **-2** в виде (десятичного) числа.

HEX2OCT

Функция **HEX2OCT** выполняет преобразование шестнадцатеричного числа в восьмеричное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=HEX2OCT(hexadecimalnumber; numdigits)

где:

- **hexadecimalnumber** — шестнадцатеричное число, заданное в виде текста или числа (воспринимаемое как шестнадцатеричное, хотя и не является им). Если задан текст, текст считается шестнадцатеричным представлением числа; если его 40-ой бит — 1, оно считается отрицательным числом;
- **numdigits** — необязательное число, определяющее количество возвращаемых цифр.

Функция **HEX2OCT** возвращает текстовое представление восьмеричного числа, являющееся эквивалентом заданного шестнадцатеричного числа **hexadecimalnumber**. Если задан текст,

текст считается шестнадцатеричным представлением числа; если его 40-ой бит — 1, оно считается отрицательным числом.

Возвращаемое восьмеричное число может иметь до десяти восьмеричных цифр, самый старший бит (30-ый бит) — знаковый, и представлено в дополнительной двоичной форме; положительные числа — от 0 до 3777777777, а отрицательные — от 7777777777 до 4000000000 (диапазон десятичных чисел от -2^{29} до $2^{29}-1$). Входное значение **hexadecimalnumber** должно поэтому также лежать в этом диапазоне, и быть представлено в дополнительной двоичной форме с десятью цифрами. Если входное значение отрицательно (40-ой бит — 1), аргумент **numdigits** игнорируется; в противном случае **numdigits** указывает количество цифр в возвращаемом значении, с ведущими нулями, добавляемыми по мере необходимости, чтобы довести количество цифр в возвращаемом значении до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D	E
1					
2		35	=HEX2OCT("1D")		
3					
4		20	=HEX2OCT(10)		
5					
6		035	=HEX2OCT("1D"; 3)		
7					
8		777777776	=HEX2OCT("FFFFFFFFFE")		
9					

В ячейке **B2** возвращается **35** в виде текста.

В ячейке **B4** возвращается **20** в виде текста. **HEX2OCT** воспринимает число 10 как шестнадцатеричное.

В ячейке **B6** возвращается **035** в виде текста. **HEX2OCT** добавляет ведущие нули, чтобы сформировать 3 цифры.

В ячейке **B8** возвращается **777777776** в виде текста (представление в дополнительной двоичной форме десятичного числа -2).

OCT2BIN

Функция **OCT2BIN** выполняет преобразование восьмеричного числа в двоичное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=OCT2BIN(octalnumber; numdigits)

где:

- **octalnumber** — восьмеричное число, заданное в виде текста или числа (воспринимаемое как восьмеричное, хотя и не является им). Если задан текст, текст считается восьмеричным представлением числа; если его 30-ый бит — 1, оно считается отрицательным числом;
- **numdigits** — необязательное число, определяющее количество возвращаемых цифр.

Функция **OCT2BIN** возвращает текстовое представление двоичного числа, являющееся эквивалентом заданного восьмеричного числа **octalnumber**, который может быть текстом, или числом (воспринимаемое как восьмеричное, хотя им и не является).

Возвращаемое двоичное число может иметь до десяти битов представленное в дополнительной двоичной форме; положительные числа — от 0 до 11111111 (девять битов,

представляющие десятичные числа от 0 до 511) и отрицательные числа — от 111111111 до 1000000000 (десять битов, представляющие десятичные числа от -1 до -512). Входное значение **octalnumber** поэтому должно также лежать в этом диапазоне, и задаваться в дополнительной двоичной форме. Если входное значение отрицательно (30-ый бит — 1), аргумент **numdigits** игнорируется; в противном случае **numdigits** указывает количество цифр в возвращаемом значении, с ведущими нулями, добавляемыми по мере необходимости, чтобы довести количество цифр в возвращаемом значении до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D	E
1					
2			1100	=OCT2BIN("14")	
3					
4			1100	=OCT2BIN(14)	
5					
6			0010	=OCT2BIN("2"; 4)	
7					
8			111111110	=OCT2BIN("777777776")	
9					

В ячейке **B2** возвращается **1100** в виде текста.

В ячейке **B4** возвращается **1100** в виде текста. **OCT2BIN** воспринимает число 14 как восьмеричное.

В ячейке **B6** возвращается **0010** в виде текста. **OCT2BIN** добавляет ведущие нули, чтобы сформировать 4 цифры.

В ячейке **B8** возвращается **111111110** в виде текста (представление в дополнительной двоичной форме десятичного числа -2).

OCT2DEC

Функция **OCT2DEC** выполняет преобразование восьмеричного числа в десятичное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=OCT2DEC(octalnumber)

где:

- **octalnumber** — восьмеричное число, заданное в виде текста или числа (воспринимаемое как восьмеричное, хотя и не является им). Если задан текст, текст считается восьмеричным представлением числа; если его 30-ый бит — 1, оно считается отрицательным числом.

Функция **OCT2DEC** возвращает (десятичное) число, соответствующее заданному восьмеричному числу **octalnumber**. **octalnumber** может иметь до десяти цифр представленное в дополнительной двоичной форме; положительные числа — от 0 до 3777777777 (представляющие десятичные числа от 0 до $2^{29}-1$) и отрицательные числа — от 7777777777 до 4000000000 (представляющие десятичные числа от -1 до -2^{29}).

Возвращаемое значение — десятичное число.

	A	B	C	D	E
1					
2		25	=OCT2DEC("31")		
3					
4		25	=OCT2DEC(31)		
5					
6		-2	=OCT2DEC("777777776")		
7					

В ячейке **B2** возвращается **25** в виде (десятичного) числа.

В ячейке **B4** возвращается **25** в виде (десятичного) числа. **OCT2DEC** воспринимает число 31 как восьмеричное.

В ячейке **B6** возвращается **-2** в виде (десятичного) числа.

OCT2HEX

Функция **OCT2DEC** выполняет преобразование восьмеричного числа в шестнадцатеричное.

Эта функция доступна, только если установлен *пакет Анализа*.

Синтаксис функции:

=OCT2HEX(octalnumber; numdigits)

где:

- **octalnumber** — восьмеричное число, заданное в виде текста или числа, содержащим только цифры от 0 до 7 (воспринимаемое как восьмеричное, хотя и не является им). Если задан текст, текст считается восьмеричным представлением числа; если его 30-ый бит — 1, оно считается отрицательным числом;
- **numdigits** — необязательное число, определяющее количество возвращаемых цифр.

Функция **OCT2DEC** возвращает текстовое представление шестнадцатеричного числа, являющееся эквивалентом заданного восьмеричного числа **octalnumber**. Входное значение **octalnumber** может иметь до десяти цифр представленное в дополнительной двоичной форме; положительные числа — от 0 до 3777777777 (представляющие десятичные числа от 0 до $2^{29}-1$) и отрицательные числа — от 7777777777 до 4000000000 (представляющие десятичные числа от -1 до -229).

Возвращаемое шестнадцатеричное число может содержать до 10 шестнадцатеричных цифр, самый старший бит (40-ой бит) — знаковый, и в дополнительной двоичной форме. Цифры от A до F — в верхнем регистре. Если входное значение — отрицательное, аргумент **numdigits** игнорируется; в противном случае, **numdigits** указывает число цифр в возвращаемом значении, с ведущими нулями, добавляемыми по мере необходимости, чтобы довести количество цифр в результате до заданного количества. Если имеется больше цифр, чем требует параметр **numdigits**, возвращается ошибка **Err:502**.

	A	B	C	D	E
1					
2		1D	=OCT2HEX("35")		
3					
4		1D	=OCT2HEX(35)		
5					
6		001D	=OCT2HEX("35"; 4)		
7					
8		FFFFFFFFFE	=OCT2HEX("777777776")		
9					

В ячейке **B2** возвращается **1D** в виде текста.

В ячейке **B4** возвращается **1D** в виде текста. **OCT2DEC** воспринимает число 35 как восьмеричное.

В ячейке **B6** возвращается **001D** в виде текста. Добавляются ведущие нули, чтобы сформировать 4 цифры.

В ячейке **B8** возвращается **FFFFFFFFFE** в виде текста (представление в дополнительной двоичной форме десятичного числа -2).

ROMAN

Функция **ROMAN** возвращает римское число (например, XIV) в виде текста, соответствующее заданному числу.

Синтаксис функции:

=ROMAN(number; mode)

где:

- **number** — число в диапазоне 1-3999, которое должно быть преобразовано в римское число;
- **mode** — (необязательный параметр, число в диапазоне 0-4) указывает степень упрощения. Чем выше значение, тем больше упрощение римского числа.

Функция **ROMAN** возвращает число в римском представлении соответствующее заданному числу. Параметр **mode** определяет уровень лаконичности, и по умолчанию — 0 (если опущен), что соответствует классическому представлению, с увеличением значения требует увеличения лаконичности.

Данная функция является обратной по отношению к функции **ARABIC**. Поддерживается следующее тождество: **ARABIC(ROMAN(x; any)) = x**, в том случае если **ROMAN(x; any)** — не возвращает ошибку.

Ниже представлены значения отдельных римских цифр; римские цифры, которые предшествуют (прямо или косвенно) большему по значению римскому числу, вычитают свое значение из окончательного значения.

Римская цифра	Значение
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Значение параметра **mode** соответствует одному из следующих:

mode	Значение
0	Вычитается только степень 10, а не L или V, и только если следующее число не более чем в 10 раз больше. Число после большего должно быть меньшим чем вычтенное число. Также известно как <i>классический</i> .
1	Степень 10, а также L и V могут быть вычтены, только если следующее число не более чем в 10 раз больше. Число после большего должно быть меньшим чем вычтенное число.

mode	Значение
2	Степень 10 и L могут быть вычтены, но не V, также если следующее число более чем в 10 раз больше. Число после большего должно быть меньшим чем вычтенное число.
3	Степень 10, а также L и V могут быть вычтены, также если следующее число более чем в 10 раз больше. Число после большего должно быть меньшим чем вычтенное число.
4	Производит наименьшее количество возможных римских цифр. Также известно как “сокращённый способ”.

	A	B	C	D	E
1					
2			CMXCIX	=ROMAN(999)	
3					
4			CMXCIX	=ROMAN(999; 0)	
5					
6			LMVLIV	=ROMAN(999; 1)	
7					
8			XMIX	=ROMAN(999; 2)	
9					
10			VMIV	=ROMAN(999; 3)	
11					
12			IM	=ROMAN(999; 4)	
13					

Эта функция в настоящее время отображается в категории Текстовые Мастера функций.

Текстовые функции

Текстовые функции манипулируют текстовыми значениями или возвращают их.

Текстовое значение (также называемое строковым значением) — последовательность из нуля или большего количества символов. Ячейка с надписью имеет этот тип, также, как и константы, заключенные в двойные кавычки. Текстовое значение нулевой длины называют пустой строкой.

Некоторые функции, такие как **FIND**, ссылаются на позицию символа в текстовом значении. Первая позиция текстового значения имеет номер 1, а не 0.

ВАНТТЕХТ

Функция **ВАНТТЕХТ** возвращает тайский текст, включающий наименование тайской валюты, соответствующий заданному числу.

Синтаксис функции:

=ВАНТТЕХТ(number)

где:

- **number** — любое число.

Функция **ВАНТТЕХТ** возвращает строку тайских символов, с добавлением “бат” к целой части числа и “сатанг” — к десятичной части числа.

	A	B	C	D
1				
2		สองหมื่นสี่พันเจ็ดสิบเจ็ดบาทถ้วน	=ВАНТТЕХТ("12.65")	
3				

Возвращается строка тайских символов со значением “Двенадцать бат и шестьдесят пять сатанг”.

Функция **ВАНТТЕХТ** не включена в разрабатываемый международный стандарт ODF. Она (по крайней мере частично) реализована в Excel.

CHAR

Функция **CHAR** возвращает один текстовый символ в соответствии с заданным кодом символа.

Синтаксис функции:

=CHAR(number)

где:

- **number** - код символа, в диапазоне 1-255.

Функция **CHAR** использует системную таблицу кодов символов (например iso-8859-1, iso-8859-2, Windows 1252, Windows 1251), чтобы определить возвращаемый символ. Коды, больше чем 127, возможно, являются не переносимыми.

	A	B	C	D
1				
2		H	=CHAR(72)	
3				

Возвращается символ **H**, соответствующий коду 72.

Международный стандартный ODFFF введет более универсальные функции **UNICODE** и **UNICHAR**, которые будут реализованы в Calc.

CLEAN

Функция **CLEAN** возвращает текстовую строку с удаленными непечатаемыми символами.

Синтаксис функции:

=CLEAN(text)

где:

- **text** — заданная текстовая строка.

Функция **CLEAN** возвращает текстовую строку с удаленными всеми непечатаемыми символами. Пробелы не удаляются.

	A	B	C	D	E
1					
2		Кошка	=CHAR(7) & "Кошка" & CHAR(8)		
3					
4		Кошка	=CLEAN(B2)		
5					
6			5	=LEN(CLEAN(B2))	
7					

В ячейке **B4** возвращается “Кошка”. Два непечатаемых символа удалены.

В ячейке **B6** возвращается 5, число символов в строке “Кошка”.

CODE

Функция **CODE** возвращает числовой код для первого символа в текстовой строке.

Синтаксис функции:

=CODE(text)

где:

- **text** — заданная текстовая строка.

Функция **CODE** возвращает числовой код для первого символа текстовой строки, в диапазоне 0-255. Коды, большие чем 127 могут зависеть от системной таблицы кодов символов (например, iso-8859-1, iso-8859-2, Windows 1251, Windows 1250), и, следовательно, не могут быть переносимыми.

Функция **CODE** является обратной по отношению к функции **CHAR**, таким образом, поддерживается следующее тождество: **CODE(CHAR(N))** возвращает **N** для любого **N** >= 0.

	A	B	C	D
1				
2		72	=CODE("H")	
3				
4		72	=CODE("Hello")	
5				

В ячейке **B2** возвращается 72, код символа для "H". В ячейке **B4** также возвращается 72.

Международный стандартный ODFFF введет более универсальные функции **UNICODE** и **UNICHAR**, которые будут реализованы в Calc.

CONCATENATE

Функция **CONCATENATE** объединяет несколько текстовых строк в одну.

Синтаксис функции:

=CONCATENATE(text1; text2; ... text30)

где:

- **text1; text2; ... text30** — текстовые строки, могут быть также ссылками на одиночные ячейки.

Функция **CONCATENATE** объединяет до 30 текстовых строк по порядку в единое текстовое значение.

	A	B	C	D	E	F
1						
2		altogether	=CONCATENATE("al"; "tog"; "ether")			
3						
4		key				
5		board				
6		keyboard	=CONCATENATE(B4; B5)			
7						

DOLLAR

Функция **DOLLAR** возвращает текстовое представление числа в локальном денежном формате.

Синтаксис функции:

=DOLLAR(number; decimals)

где:

- **number** — заданное число;
- **decimals** — (необязательный, предполагается равным 2 если опущен) устанавливает количество десятичных знаков.

Валюта по умолчанию устанавливается в диалоговом окне **Сервис — Параметры — Языковые настройки — Языки - “Валюта по умолчанию”** (обычно валюта в соответствии с вашими региональными настройками).

	A	B	C	D	E
1					
2		255,00 руб.	=DOLLAR(255)		
3					
4		367,46 руб.	=DOLLAR(367,456; 2)		
5					

Поскольку она зависит от региональных настроек, функция **DOLLAR** не должен расцениваться как переносимая.

EXACT

Функция **EXACT** возвращает **TRUE**, если две текстовых строки идентичны.

Синтаксис функции:

=EXACT(text1; text2)

где:

- **text1; text2** — сравниваемые текстовые строки, могут быть также ссылками на

одиночные ячейки.

Функция **EXACT** возвращает **TRUE**, если две текстовых строки являются идентичными (в том числе учитывая регистр).

	A	B	C	D	E
1					
2			ИСТИНА	=EXACT("red car"; "red car")	
3					
4			ЛОЖЬ	=EXACT("red car"; "Red Car")	
5					

FIND

Функция **FIND** возвращает положение строки текста в пределах другой строки.

Синтаксис функции:

=FIND(findtext; texttosearch; startposition)

где:

- **findtext** — искомый текст;
- **texttosearch** — текст, в котором осуществляется поиск;
- **startposition** — (необязательный) позиция, с которой начинается поиск.

Функция **FIND** возвращает позицию символа первого появления строки **findtext** в пределах **texttosearch**. Поиск осуществляется с учетом регистра, и никакие групповые символы или другие инструкции не принимаются во внимание в **findtext**. Если строка не найдена, возвращается ошибка **#VALUE!**.

	A	B	C	D	E
1					
2			3	=FIND("yo"; "Yoyo")	
3					
4			5	=FIND("cho"; "choochoo"; 2)	
5					

FIXED

Функция **FIXED** возвращает текстовое представление числа, округленного до указанного числа десятичных знаков.

Синтаксис функции:

=FIXED(number; decimals; omitseparators)

где:

- **number** — число для округления;
- **decimals** — целое число, задающее количество десятичных знаков, отображаемых в результате;
- **omitseparators** — (необязательный) если этот параметр установлен в **TRUE**, то в результате не будут отображаться разделители тысяч, если он будет установлен в **FALSE** или опущен разделители тысяч в результате будут отображаться.

Функция **FIXED** округляет значение **number** до **decimals** десятичных знаков (после десятичного разделителя), и возвращает результат, отформатированный в виде текста, используя региональные настройки. Если необязательный параметр **omitseparators** (который по умолчанию **FALSE**) **TRUE**, то разделители тысяч не включаются в итоговую строку.

Используемый символ разделителя тысяч зависит от ваших региональных настроек; см. **Сервис — Параметры — Настройки языка — Языки**.

	A	B	C	D	E
1					
2		1 234 567,890	=FIXED(1234567,89; 3)		
3					
4		1234567,890	=FIXED(1234567,89; 3; TRUE())		
5					

В ячейке **B2** возвращается 1 234 567,890 в виде текста, если ваши региональные настройки использует пробел в качестве разделителя тысяч.

LEFT

Функция **LEFT** возвращает текст с левой стороны текстовой строки.

Синтаксис функции:

=LEFT(text; number)

где:

- **text** — заданная текстовая строка, содержащая символы, которые требуется извлечь;
- **number** — количество возвращаемых символов (по умолчанию — 1, если опущено).

Функция **LEFT** возвращает **number** первых символов текстовой строки. Если строка **text** имеет меньше символов чем **number**, функция возвращает всю строку **text**. Это означает, что, если **text** — пустая строка (длина которой равна нулю) или параметр **number** — 0, **LEFT()** будет всегда возвращать пустую строку. Отметьте это, если **number** меньше 0, возвращается ошибка.

	A	B	C	D
1				
2		out	=LEFT("output"; 3)	
3				

LEN

Функция **LEN** возвращает длину текстовой строки.

Синтаксис функции:

=LEN(text)

где:

- **text** — заданная текстовая строка.

Функция **LEN** возвращает количество символов в тексте, включая пробелы.

	A	B	C	D
1				
2		7	=LEN("red car")	
3				
4		5	=LEN(123,4)	
5				

В ячейке **B2** возвращается 7.

В ячейке **B4** возвращается 5, число предварительно преобразуется в текст.

LOWER

Функция **LOWER** преобразует символы текстовой строки к нижнему регистру.

Синтаксис функции:

=LOWER(text)

где:

- **text** — заданная текстовая строка.

Функция **LOWER** возвращает текст со всеми символами, преобразованными к нижнему регистру (строчным буквам).

	A	B	C	D	E
1					
2		доброе утро	=LOWER("Доброе УТРО")		
3					

MID

Функция **MID** возвращает текст из середины текстовой строки.

Синтаксис функции:

=MID(text; start; number)

где:

- **text** — заданная текстовая строка;
- **start** — позиция первого символа, извлекаемого из текста. Первый символ в заданной текстовой строке имеет позицию 1 и так далее;
- **number** — определяет число возвращаемых символов.

Функция **MID** возвращает **number** символов из **text**, начиная с позиции **start**. Если имеется меньше символов чем **number**, начиная со **start**, функция возвращает столько символов, сколько имеется начиная со **start**. В частности если **start** больше чем длина заданной строки, возвращается пустая строка (""). Если **start** меньше нуля, возвращается ошибка. Если **start** больше или равно нулю, а **number** равно нулю, возвращается пустая строка.

	A	B	C	D
1				
2		ХО	=MID("выход"; 3; 2)	
3				

PROPER

Функция **PROPER** возвращает текст со словами в нижнем регистре с заглавной первой буквой.

Синтаксис функции:

=PROPER(text)

где:

- **text** — заданная текстовая строка.

Функция **PROPER** возвращает заданную текстовую строку, измененную следующим образом:

- Если первый символ каждого слова — буква, она преобразуется к своему заглавному эквиваленту; в противном случае, возвращается исходный символ;

- Если букве предшествует не-буква, она преобразуется к своему заглавному эквиваленту;
- Если букве предшествует буква, она преобразуется к своему строчному эквиваленту.

	A	B	C	D	E
1					
2		=PROPER("доброЕ утРО")			
3					

REPLACE

Функция **REPLACE** заменяет часть текстовой строки на другую текстовую строку.

Синтаксис функции:

=REPLACE(originaltext; startposition; length; newtext)

где:

- **originaltext** — заданная текстовая строка;
- **start** — позиция первого символа замещаемого текста. Первый символ в заданной текстовой строке имеет позицию 1 и так далее;
- **length** — определяет длину замещаемой подстроки;
- **newtext** — заменяющий текст.

Функция **REPLACE** в **originaltext**, удаляет **length** символов, начиная с символа **startposition**, заменяет их на **newtext**, и возвращает результат.

startposition и **length** должны быть 1 или более.

Функция **REPLACE** очень похожа на функцию **SUBSTITUTE**.

При использовании **REPLACE**, мы указываем позицию и длину подстроки, которые будут заменены, что это за подстрока, мы не знаем.

При использовании **SUBSTITUTE**, мы знаем то, что мы заменяем, только не уверены точно, где оно находится в заданной строке.

	A	B	C	D	E
1					
2		=REPLACE("mouse"; 2; 3; "ic")			
3					

В ячейке **B2** возвращается **mice**. Начиная с символьной позиции 2, удаляются 3 символа (**ous**) и заменяются на **ic**.

Excel и разрабатываемый стандарт ODFD допускают, что **length** может быть нулем.

REPT

Функция **REPT** повторяет указанную строку или символ заданное число раз.

Синтаксис функции:

=REPT(text; number)

где:

- **text** — заданная текстовая строка;
- **number** — положительное число, определяющее, сколько раз нужно повторить текст.

Функция **REPT** возвращает **number** копий **text**, объединенных вместе. Если **number** — ноль, возвращается пустая строка. Если **number** меньше нуля, возвращается ошибка.

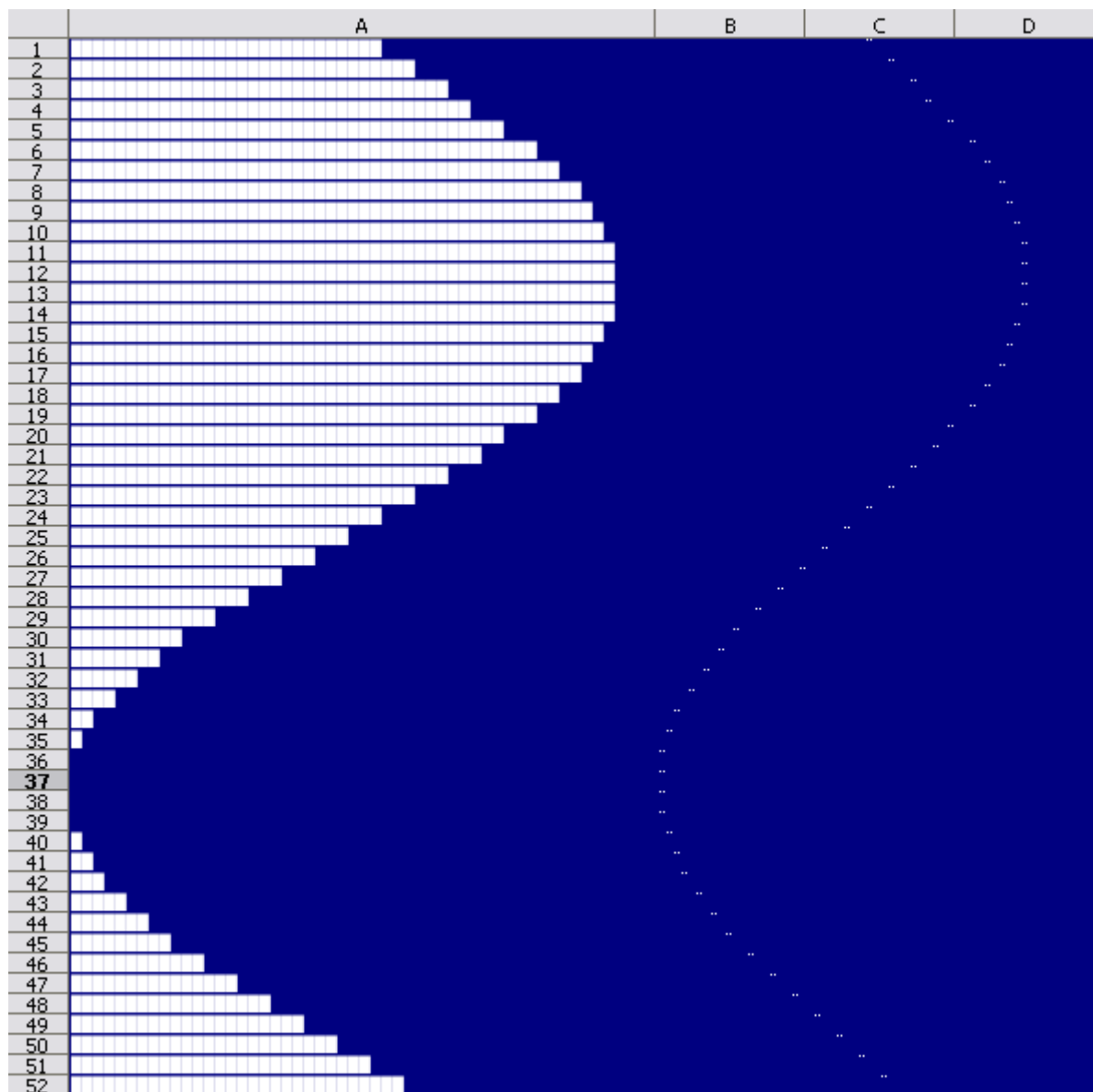
Функция **REPT** может использоваться для создания простых диаграмм без использования инструмента построения диаграмм OOo Calc — хотя не ясно, было ли это подлинным назначением функции.

Построим гистограмму на основе функции синуса. В столбце **A** мы имеем

=REPT("█";INT((SIN(ROW()/8)+1)*25))

а в столбце **B**

=CONCATENATE(REPT(" ";INT((SIN(ROW()/8)+1)*25));"*")



RIGHT

Функция **RIGHT** возвращает текст с правой стороны текстовой строки.

Синтаксис функции:

=RIGHT(text; number)

где:

- **text** — заданная текстовая строка;
- **number** — количество извлекаемых знаков.

Функция **RIGHT** возвращает **number** символов с правой стороны текстовой строки **text**. По умолчанию **number** — 1, если опущено. Если **text** имеет меньше символов чем **number**,

возвращается вся текстовая строка **text**. Это означает, что, если **text** - пустая строка (которая имеет длину 0) или параметр **number** — 0, функция **RIGHT()** будет возвращать пустую строку. Заметьте это, если **number** меньше нуля, возвращается ошибка.

	A	B	C	D
1				
2		put	=RIGHT("output"; 3)	
3				

SEARCH

Функция **SEARCH** возвращает положение строки текста в пределах другой строки.

Синтаксис функции:

=SEARCH(findtext; texttosearch; startposition)

где:

- **findtext** — искомая текстовая строка;
- **texttosearch** — текст, в котором осуществляется поиск;
- **startposition** — (необязательный) позиция, с которой начинается поиск.

Функция **SEARCH** возвращает положение символа из первого появления **findtext** в пределах **texttosearch**, начиная с символьной позиции **startposition**. По умолчанию **startposition** — 1, если опущен. Поиск не учитывает регистр.

Поиск будет использовать регулярные выражения, если они разрешены (**Сервис — Параметры — OpenOffice.org Calc — Вычисления**).

Если найти строку **findtext** не удалось, возвращается ошибка **#VALUE!**.

Установка параметра **Условия поиска = и <>** должны распространяться на всю ячейку в диалоговом окне **Сервис — Параметры — OpenOffice.org Calc — Вычисления**, не оказывает на поиск никакого эффекта.

	A	B	C	D	E
1					
2		1	=SEARCH("yo"; "Yoyo")		
3					
4		5	=SEARCH("cho"; "choochoo"; 2)		
5					
6		3	=SEARCH("t.n"; "often")		
7					

В ячейке **B2** возвращается 1. Поиск выполняется без учета регистра.

В ячейке **B6** возвращается 3, если регулярные выражения разрешены. “.” означает любой одиночный символ в регулярном выражении, таким образом “t.n” соответствует “ten”.

SUBSTITUTE

Функция **SUBSTITUTE** заменяет новым текстом старый текст в текстовой строке.

Синтаксис функции:

=SUBSTITUTE(originaltext; oldtext; oldtext; which)

где:

- **originaltext** — текст, в котором Вы хотите выполнить замену символов;
- **oldtext** — текст, который Вы хотите заменить;

- **oldtext** — текст, на который Вы хотите заменить **oldtext**;
- **which** — (необязательный) номер появления, которое Вы хотите заменить.

Функция **SUBSTITUTE** в **originaltext** удаляет **oldtext**, вставляет **newtext** на его место, и возвращает результат. **oldtext** и **newtext** могут иметь различную длину.

Параметр **which** (необязательный) — число, которое определяет какое появление **oldtext** заменить (счет выполняется слева направо). Если опущен, заменяются все появления.

Эта функция должна использоваться, когда Вы хотите заменить все появления некоторого текста независимо от их положения в тексте. Если **which** оставить пустым, каждое появление **oldtext** заменяется на **newtext**. Функция **SUBSTITUTE** очень похожа на функцию **REPLACE**.

При использовании **REPLACE**, мы указываем позицию и длину подстроки, которые будут заменены, что это за подстрока, мы не знаем.

При использовании **SUBSTITUTE**, мы знаем то, что мы заменяем, только не уверены точно, где оно находится в заданной строке.

	A	B	C	D	E	F
1						
2		cave	=SUBSTITUTE("castle"; "stl"; "v")			
3						
4		a white white rose	=SUBSTITUTE("a red red rose"; "red"; "white")			
5						
6		a red and white rose	=SUBSTITUTE("a red red rose"; "red"; "and white"; 2)			
7						

В ячейке **B4** возвращается “a white white rose”. Все появления **red** заменяются если параметр **which** опущен.

В ячейке **B6** возвращается “a red and white rose”. Заменяется второе появление **red**.

T

Функция **T** возвращает заданный текст или пустую текстовую строку, если задан не текст.

Синтаксис функции:

=**T**(value)

где:

- **value** — входное значение.

Если **value** — текст, возвращается он. Если **value** — число или логическое значение, возвращается пустая текстовая строка “”.

	A	B	C	D
1				
2		red car	=T("red car")	
3				
4			=T(123,4)	
5				
6		ИСТИНА	=ISTEXT(T(123,4))	
7				
8		#Н/Д	=T(NA())	
9				

В ячейке **B2** возвращается текстовая строка “red car”.

В ячейке **B4** возвращается пустая текстовая строка.

В ячейке **B6** возвращается **TRUE**.

В ячейке **B8** возвращается значение ошибки **#N/A**. Ошибки передаются.

TEXT

Функция **TEXT** преобразует число в текст согласно заданному формату.

Синтаксис функции:

=TEXT(number; format)

где:

- **number** — либо числовое значение, либо формула, вычисление которой дает числовое значение, либо ссылка на ячейку, содержащую числовое значение;
- **format** — строка формата.

Функция **TEXT** возвращает число **number**, преобразованное в текст, согласно коду формата, указанному строкой **format**. Примера кодов формата могут быть найдены в диалоговом окне **Формат — Ячейки...**, в поле Код формата.

Результат этой функции может изменяться в зависимости от региональных настроек. Если затрагиваются разделители, такие как десятичный или разделитель групп, преобразование может дать неожиданные результаты, если разделители не соответствуют разделителям в текущих региональных настройках. Переносимые документы не должны использовать эту функцию.

	A	B	C	D	E
1					
2		12,35	=TEXT(12,34567;"###,##")		
3					

В ячейке **B2** возвращается текст **12,35**, если для ваших региональных настроек формат **###,##** — код формата значение “до трех цифр целой части числа и самое большее два десятичных знака после запятой”.

На первый взгляд, функция **TEXT**, возможно, не кажется очень полезной. Но вот несколько простых ее применений.

	A	B	C	D	E	F
1						
2		10 435,00 р.	Всего: 10435			
3			=Всего: " & B2			
4						
5			Всего: 10 435,00 р.			
6			=Всего: " & TEXT(B2; "# ##0,00 [\$p.-419]")			
7						
8						
9		16.08.2008	Дата - 16.08.08			
10			=Дата - " & TEXT(B9; "DD.MM.YY")			
11						

В верхнем примере, без функции **TEXT**, общее количество не форматируется правильно (как денежное значение). Однако, функция **TEXT** позволяет нам применять правильный формат к числовому компоненту строки.

Во втором примере, мы видим, как функция **TEXT** позволяет нам переформатировать существующие данные — такие, как даты.

TRIM

Функция **TRIM** удаляет лишние пробелы из текстовой строки.

Синтаксис функции:

=TRIM(text)

где:

- **text** — заданная текстовая строка, из которой удаляются пробелы.

Функция **TRIM** возвращает текст с удаленными ведущими или замыкающими пробелами и замененными с множественными пробелами на одиночные пробелы.

	A	B	C	D	E
1					
2		Доброе Утро	=TRIM(" Доброе Утро ")		
3					

UPPER

Функция **UPPER** преобразует символы текстовой строки к верхнему регистру.

Синтаксис функции:

=UPPER(text)

где:

- **text** — заданная текстовая строка.

Функция **UPPER** возвращает текст со всеми символами, преобразованными к верхнему регистру (прописным буквам).

	A	B	C	D	E
1					
2		ДОБРОЕ УТРО	=UPPER("Доброе Утро")		
3					

VALUE

Функция **VALUE** возвращает число, соответствующее заданному текстовому представлению.

Синтаксис функции:

=VALUE(text)

где:

- **text** — текстовая строка или ссылка на ячейку, содержащую текст, который нужно преобразовать.

Функция **VALUE** преобразует текстовую строку **text** в число. Текстовая строка может представлять: число, включая дроби, экспоненциальное представление и (в зависимости от региональных настроек) десятичный разделитель и разделитель тысяч; процент; дату или время (возвращается дата/время в числовом формате).

	A	B	C	D	E
1					
2		12	=VALUE("12")		
3					
4		1234,56	=VALUE("1 234,56")		
5					
6		1,5	=VALUE("1 1/2")		
7					
8		1000	=VALUE("1E3")		
9					
10		0,5	=VALUE("50%")		
11					
12		39906	=VALUE("2009-04-03")		
13					

В ячейке **B2** возвращается 12 в виде числа.

В ячейке **B4** возвращается 1234.56, если ваши региональные настройки признают в качестве разделителя тысяч “ ” (пробел), а в качестве десятичного разделителя “,”.

В ячейке **B6** возвращается 1.5 (дробь).

В ячейке **B8** возвращается 1000 (экспоненциальное представление).

В ячейке **B10** возвращается 0,5 (процентное представление).

В ячейке **B12** возвращается 39906, дата/время в числовом формате для 3 апреля 2009 г.

Функции массивов

Функции массивов работают с и возвращают массивы, которые являются прямоугольными блоками данных. Некоторые из этих функций требуют, чтобы формула была введена как “формула массива”.

Использование массивов

Множества могут быть полезными в сокращении количества используемых Вами ячеек, но они могут быть немного сложны для понимания. До некоторой степени, они расширяют то, что электронная таблица может сделать сверх того, для чего были предназначены электронные таблицы. Решение почти всех задачи возможно без их использования.

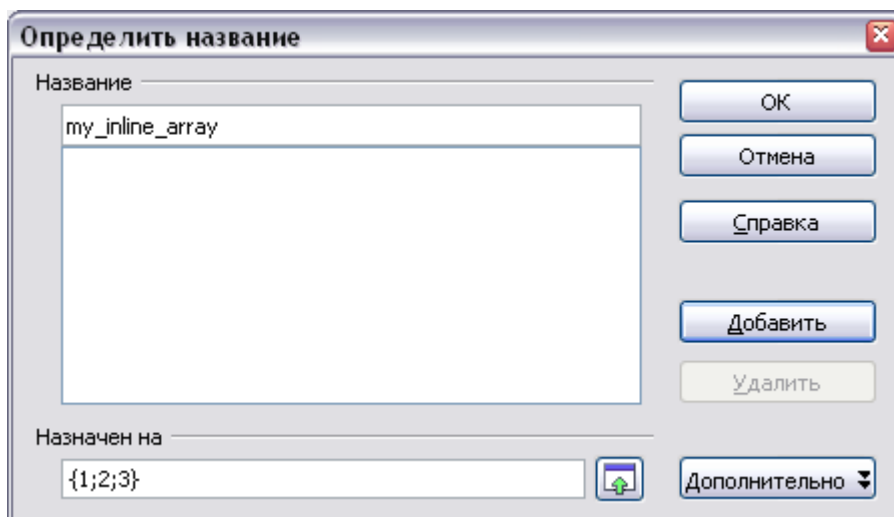
Массив — просто прямоугольный информационный блок, которым Calc может манипулировать в формуле — то есть, это — информация, организованная в строки и столбцы. Массив может быть ячейками в электронной таблице, или может содержаться внутри Calc.

Есть 2 метода определения массива в формуле:

- как диапазон - например **A2:C3**;
- как “встроенный массив”, например **{1; 5; 3 | 6; 2; 4}** (они полностью функциональны начиная с OOo2.4, но существуют и в более ранних версиях — см. *Проблемы*). Вы вводите фигурные скобки **{}** вокруг встроенного массива. Элементы в строке разделяются точкой с запятой “;”, а строки — символом вертикальной черты “|”. Каждая строка должна иметь одно и то же число элементов (неправильно записать **{1; 2; 3 | 4; 5}**, потому что есть 3 элемента в первой строке и только 2 в следующей). Встроенный массив может иметь смешанное содержимое, например **{4; 2; “собака” | -22; “кот”; 0}**. Однако он не может содержать ссылки (например, **A4**), или формулы (например, **PI ()**, **2 * 3**), или проценты (например, **5%**).

Вы можете дать имя к диапазону ячеек: выберите диапазон и выполните **Вставка — Названия — Определить...**

Вы можете дать имя встроенному массиву: выберите **Вставка — Названия — Определить...**; введите массив (например, **{1; 3; 2}**, включая фигурные скобки) в поле **Назначен на**.



Функции, которые понимают массивы в качестве параметров

Некоторые функции, такие как **SUM()**, **AVERAGE()**, **MATCH()**, **LOOKUP()**, принимают один или более своих параметров в виде массива.

Например:

- **SUM(A2:C3)** возвращает сумму чисел в диапазоне **A2:C3**;
- **SUM({3; 2; 4})**, возвращает **9**, сумму чисел в встроенном массиве **{3; 2; 4}**;
- **SUM** также ожидает/понимает одиночные значения (“скаляр”) — **SUM(B5; 7)**, возвращает сумму значения ячейки **B5** и **7**.

Функции, не ожидающие в качестве параметров массивы

Некоторые функции, такие как **ABS()**, **SQRT()**, **COS()**, **LEN()** ожидают, что их параметры “скаляры” — то есть, одиночные значения, такие как **354** или “собака” или содержимое ячейки, например, **B5**.

Например:

- **SQRT(4)** возвращает **2**;
- **LEN(“собака”)** возвращает **6**.

Однако, Вы можете использовать массив там, где ожидается одиночное значение — например, **SQRT({9; 4})**. Если Вы введете формулу “обычно”, нажав *Enter*, то Calc в этом случае вычислит формулу, используя одиночное значение из массива следующим образом:

- Если это встроенный массив, Calc будет использовать первое значение (“верхнее, левое”).

Пример:

=SQRT({9; 4 | 25; 16}), после нажатия *Enter* возвращает **3**, квадратный корень первого элемента в массиве (**9**).

- Если это диапазон:
 1. Calc возвратит ошибку, если массив содержит не одну строку или один столбец;
 2. Для диапазона из одной строки или одного столбца, Calc будет использовать значение, где столбец/строка ячейки формулы пересекается с массивом (или возвратит ошибку, если такого пересечения нет).

Например:

	A	B	C	D	E
1					
2		-4			
3		2		2	
4		-5			
5		3			
6					

Формула **=ABS(B2:B5)** введена “обычно” в ячейке **D3**, которая находится в строке 3. Строка 3 пересекает **B2:B5** в ячейке **B3**, таким образом вычисляется формула **=ABS (B3)**.

	A	B	C	D	E
1		2			
2					
3					
4					
5		до	над	пересечение	
6					
7					

Формула **=LEN(B5:D5)** введена “обычно” в ячейке **B1**, которая находится в столбце B. Столбец B пересекает диапазон **B5:D5** в ячейке **B5**, таким образом вычисляется формула **=LEN (B5)**.

Формулы массивов

Действительная мощь массивов проявляется, когда Вы вводите формулу специальным способом, как “формулу массива”. Вы делаете это, нажимая *Ctrl-Shift-Enter* вместо *Enter* (или помечая флажок **Массив**, если Вы используете *Мастер функций*).

Если в ячейке **B1** Вы вводите **={3; 4}**, “обычно” нажимая *Enter*, первое значение **3** отображается в ячейке.

Если в ячейке **B2** Вы вводите **={3; 4}**, но нажимаете *Ctrl-Shift-Enter* вместо *Enter*, ячейка становится “формулой массива”. Формула теперь возвращает весь массив **{3; 4}**. Ячейка **B2** отображает **3**, а ячейка **C2** — **4**.

Заметьте, что, если Вы ввели формулу, используя клавишу *Enter*, просто выбрав ячейку и нажав *Ctrl-Shift-Enter*, Вы не преобразуете ячейку в выражение массива — Вы должны выполнить фактическое редактирование (например, добавив, а затем удалив символ), или запустить *Мастера Функций* и пометить флажок **Массив**.

Если Вы теперь попробуете отредактировать ячейку **B2**, Вам скажут, что “*Вы не можете изменить только часть массива*”. Чтобы отредактировать массив, Вы должны выбрать весь массив, или при помощи мыши или нажав *Ctrl-/* (удерживая клавишу *Ctrl* нажать клавишу */*).

The screenshot shows the Excel interface. The formula bar at the top displays **B2:C2** and the formula **= {3;4}**. Below the formula bar, a grid of cells is visible. Cell **B2** contains the value **3**, and cell **C2** contains the value **4**. The cells **B2** and **C2** are highlighted with a blue background, indicating they are part of the array result.

Панель формул указывает, что формула — формула массива, заключая ее во фигурные скобки **{}**. Вы не должны вводить их — они исчезнут, когда Вы редактируете формулу, и Calc отобразит им снова, когда Вы закончите редактирование.

Вычисления формулы массива

Когда Calc вычисляет формулу массива, он рассматривает “непредвиденный массив” как ряд значений (вместо того, чтобы использовать одиночное значение), вычисляя результат для каждого из элементов массива, и возвращая массив результатов.

Например:

=SQRT({16; 4; 25}), когда введено нажатием *Ctrl-Shift-Enter* вместо *Enter*, возвращает массив результатов, с 1 строкой и 3 столбцами — **{4; 2; 5}**. Если формула находится в ячейке **B2**, Calc помещает результаты в ячейки **B2:D2**. **4** помещается в **B2**, **2** — в **C2** и **5** — в **D2**.

	A	B	C	D	E
1					
2		4	2	5	
3					

Процесс в действительности работает следующим образом:

1. Все “непредвиденные массивы” в одном и том же вычислении массива должны быть одинакового размера;
2. Результат будет возвращен в виде массива того же размера;
3. Вычисление выполняется для каждого элемента по очереди, с результатом, возвращаемым в соответствующем элементе производимого массива.

Например, с формулой массива **=SQRT({16; 4; 25})**:

1. Есть только один массив, с 1 строкой и 3 столбцами;
2. Результат будет возвращен в массиве с 1 строкой и 3 столбцами;
3. Вычисление выполняется сначала для 16, затем для 4, и затем для 25, давая в результате массив **{4; 2; 5}**.

Например, с формулой множества **=SQRT({8 | 18} * 2)** в ячейке A5:

1. Есть только один массив, с 2 строками и 1 столбцом;
2. Результат будет возвращен в массиве с 2 строками и 1 столбцом;
3. Вычисление: первый элемент **SQRT(8*2) = 4**; второй элемент **SQRT(18*2) = 6**; результирующий массив, таким образом, **{4 | 6}** — то есть, 4 в ячейке A5 и 6 в ячейке A6.

Результат выражения массива — массив, который может использоваться в пределах формулы.

Например, **=SUM(SQRT({16; 4; 25}))** как формула массива. Вычисление **SQRT({16; 4; 25})**, как и прежде, приводит к результату **{4; 2; 5}**, таким образом давая **SUM({4; 2; 5})**, которая возвращает конечный результат в ячейке **4+2+5 = 11**.

Например, с формулой массива **=SUM(IF(A1:A4 > 0; B1:B4; 0))**

1. Два массива **A1:A4** и **B1:B4** и имеют 4 строки и 1 столбец.
2. Результатом обработки массива функцией **IF()** получается массив того же размера, который будет суммировать **SUM()**;
3. Если **A1 > 0** первый элемент — **B1**; иначе **0**. Если **A2 > 0** второй элемент — **B2**; иначе **0**... Массив предоставляемый **SUM()**, имеет значения в **B1:B4**, когда смежные значения в **A1:A4** — **>0**. Окончательный результат — сумма значений в **B1:B4**, для которых смежные значения в **A1:A4** — **>0**.

Функции массива

Некоторые функции вычисляют свой результат в виде массива. Если они должны вернуть массив, они должны быть введены как формула массива, нажатием *Ctrl-Shift-Enter* (или установкой флажка **Массив**, если Вы используете *Мастера формул*). Если, взамен, они будут введены “обычно”, клавишей *Enter*, то будет возвращен только (единственный) верхний левый элемент рассчитанного массива. Вот эти функции: **FREQUENCY**, **GROWTH**, **LINEST**, **LOGEST**, **MDTERM**, **MINVERSE**, **MMULT**, **MUNIT**, **SUMPRODUCT**, **SUMX2MY2**, **SUMX2PY2**, **SUMXMY2**, **TRANSPOSE**, **TREND**.

Например: **=MUNIT(2)** введенная как формула массива в ячейку **B1** возвращает единичную матрицу 2×2 в виде массива **{1; 0 | 0; 1}** — то есть, ячейки **B1**, **B2**, **C1**, **C2** отображают 1, 0,

0, 1. Если эта формула введена “обычно”, только “верхнее, левое” значение (1) возвращается в ячейке **B1**.

Некоторые функции (включая некоторые упомянутые выше) берут параметры, которые вынуждены оценивать как формулу массива, даже если формула вводится “обычно”: **MDETERM**, **MINVERSE**, **MMULT**, **SUMPRODUCT**, **SUMX2MY2**, **SUMX2PY2**, **SUMXMY2**, **CORREL**, **COVAR**, **FORECAST**, **FTEST**, **INTERCEPT**, **MODE**, **PEARSON**, **PROB**, **RSQ**, **SLOPE**, **STEYX**, **TTEST**.

Например: **MODE(ABS(A1:A3))** (в нормальном режиме) вынуждает **ABS(A1:A3)** оцениваться как формулу массива, возвращая массив абсолютных значений **A1:A3**, из которого **MODE** выбирает наиболее распространенное значение, которое и будет возвращено.

Например: **PROB(A1:A5; B1:B5/100; 3)** (в нормальном режиме) заставляет **B1:B5/100** оцениваться как формула массива, возвращая массив, где каждое значение является сотой частью из значений в **B1:B5** (возможно, потому, что вероятности выражаются в процентах).

Некоторые из этих функций имеют некоторые параметры (массивы), которые вынуждены оценивать в виде формулы массива, а другие параметры (одиночное значение), которые оценивают обычно.

Например: **PROB(A1:A5; B1:B5; {3; 4})** (в нормальном режиме) не оценивает **{3; 4}** как массив, потому что это означало бы, что **PROB** должна вернуть массив. Она вычисляет **PROB(A1:A5; B1:B5; 3)** — то есть, она интерпретирует **{3; 4}** в скалярном режиме, беря “верхнее левое” значение 3. Однако, если она введена как формула массива, она конечно возвратит массив.

Проблемы

Встроенные массивы работают в OOo2.3, но вызывают ошибку, если есть какие-нибудь пробелы или отрицательные числа. Это установлено для OOo2.4, хотя пробелы удаляются, а не игнорируются.

Можно будет включать в формулу массивы различного размера, как определено процесс вычисления в готовящемся международном стандарте ODF.

Некоторые функции все же не могут использоваться в формулах массивов: **COUNTIF**, **SUMIF**, **MATCH**, **ISLOGICAL**.

OFFSET имеет ограниченное поведение в пределах формулы массива.

Имена, определенные через **Вставка — Названия — Определить...**, могут использоваться в пределах формул массивов, но надписи (добавленные через **Вставка — Названия — Подписи** или автоматически распознанные в заголовке столбца), не должны использоваться.

В Excel, не все функции могут использоваться в формулах массивов. Например **CONCATENATE** работает в Calc, но не в Excel. Знайте об ограничениях Excel, если Вы планируете использовать электронную таблицу Calc в Excel.

Изменчивые функции, такие как **RAND**, **RANDBETWEEN**, **NOW** не могут повторно вычисляться в формулах массивов; например в **{=A1:A3+RAND()}** порождается только одно случайное число.

Формулы массивов могут замедлить ваш компьютер, если Вы имеете большие массивы.

FREQUENCY

Функция **FREQUENCY** возвращает массив, распределяющий по категориям значения набора данных в заданных интервалах.

Синтаксис функции:

=FREQUENCY(data; bins)

где:

- **data** — диапазон или массив, содержащий числовые данные;
- **bins** — однострочный диапазон или массив, содержащий числа в порядке возрастания, которые являются верхним пределом каждой категории.

Функция **FREQUENCY** возвращает однострочный массив, где первый элемент — счетчик значений в **data**, которые меньше или равны первому значению в **bins**, второе значение — счетчик значений в **data**, которые больше чем первое значение, но меньше или равны второму значению в **bins**, и так далее. Возвращаемый массив — на один элемент больше чем **bins**; последний элемент содержит счетчик значений в **data**, которые больше чем последнее значение в **bins**.

Чтобы вернуть массив, функция **FREQUENCY** должна быть введена как формула массива, нажатием *Ctrl-Shift-Enter* вместо клавиши *Enter* (или с установленным флажком **Массив**, если Вы используете *Мастер формул*).

	A	B	C	D	E
1	12	5	1		
2	8	10	3		
3	24	15	2		
4	11	20	3		
5	5	25	1		
6	20		1		
7	16				
8	9				
9	7				
10	16				
11	33				
12					

Когда формула **=FREQUENCY(A1:A11; B1:B5)** введена как формула массива, где значения данных в **A1:A11** — 12, 8, 24, 11, 5, 20, 16, 9, 7, 16, 33, а значения в **B1:B5** — 5, 10, 15, 20, 25 возвращается массив **{1|3|2|3|1|1}**. Первое значение — счетчик значений данных, меньших или равных 5, а последнее значение — счетчик значений данных, больших чем 25.

GROWTH

Функция **GROWTH** подбирает экспоненциальную кривую для набора данных, и возвращает точки на этой кривой.

Синтаксис функции:

=GROWTH(yvalues; xvalues; new_xvalues; allow_factor)

где:

- **yvalues** и **xvalues** — однострочные или однострочные диапазоны, определяющие точки в наборе данных;
- **new_xvalues** — однострочный или однострочный диапазон, определяющий положение точек в выходном наборе данных;
- **allow_factor** - Если **FALSE**, коэффициент **b** в уравнении задается равным 1; если отличен от нуля, **TRUE** или опущен, коэффициент **b**, вычисляется на основе данных.

Функция **GROWTH** подбирает показательную кривую вида $y=bm^x$, через заданные точки данных, используя метод линейной регрессии. Функция **GROWTH** возвращает массив

значений **y** найденной показательной кривой, соответствующих значениям **x** в **new_xvalues** (или если опущено, в **xvalues**). Она должно быть введена как формула массива (например при использовании *Ctrl-Shift-Enter*, а не просто *Enter*).

Диапазоны **yvalues** и **xvalues** должны иметь одинаковый размер. Диапазон **new_xvalues** может иметь другой размер.

	A	B	C	D
1	X	Y		
2	4	1,8	1,72134774688769	
3	5	3,2	3,31988350530959	
4	6	6,2	6,40290522862362	
5	7	12,3	12,34898613194340	
6	8	24,5	23,81691639057910	
7	9		45,93458121137000	
8				

Когда формула **=GROWTH(B2:B6; A2:A6; A2:A7)** введена как формула массива в ячейке **C2**, где значения **x** в диапазоне **A2:A7** — 4, 5, 6, 7, 8, 9 и значения **y** в диапазоне **B2:B6** — 1,80, 3,20, 6,20, 12,30, 24,50, находит наиболее соответствующую показательную кривую для этих точек, и в диапазоне **C2:C7** возвращает значения **y** на той кривой для значений **x** из диапазона **A2:A7**.

Этот пример показывает, как **GROWTH** может использоваться для предсказания будущих значений (A7, C7).

LINEST

Функция **LINEST** возвращает таблицу статистических данных для прямой линии, которая лучше всего соответствует набору данных.

Синтаксис функции:

=LINEST(yvalues; xvalues; allow_const; stats)

где:

- **yvalues** — однострочковый или одностолбцовый диапазон, определяющий координаты **y** в наборе точек данных;
- **xvalues** — соответствующий однострочковый или одностолбцовый диапазон, определяющий координаты **x**. Если **xvalues** опущен, то по умолчанию принимаются значения 1, 2, 3..., **n**. Если имеется более чем один набор переменных, **xvalues** может быть диапазоном с соответствующими несколькими строками или столбцами.
- **allow_const** — если **FALSE**, найденная прямая линия проходит через начало координат (константа — ноль; $y = bx$), если опущен, по умолчанию — **TRUE** (линия не проходит через начало координат);
- **stats** — если опущен или **FALSE**, возвращается только верхняя строка таблицы статистических данных (рассчитывается только коэффициент регрессии), если **TRUE** возвращается вся таблица.

Функция **LINEST** находит прямую линию $y = a + bx$, которая лучше всего соответствует данным, используя линейную регрессию (метод “наименьших квадратов”). С более чем одним набором переменных прямая линия имеет форму:

$$y = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

Функция **LINEST** возвращает таблицу (массив) статистических данных, приведенных ниже, и должна быть введена как формула массива (например с использованием *Ctrl-Shift-Enter*, а

не просто *Enter*).

b_n	b_{n-1}	...	b_1	a
σ_n	σ_{n-1}	...	σ_1	σ_a
r^2	σ_y			
F	df			
SS_{reg}	SS_{resid}			

где:

- $b_1 \dots b_n$ — угол наклона линии; a — точка пересечения оси Y;
- $\sigma_1 \dots \sigma_n$ — значение среднеквадратичной погрешности для угол наклона линии; σ_a — значение среднеквадратичной погрешности для точки пересечения оси Y;
- r^2 — коэффициент детерминации (RSQ); σ_y — значение среднеквадратичной погрешности для значения y;
- F — статистический критерий Фишера F (наблюдаемая величина F); df — число степеней свободы;
- SS_{reg} — сумма квадратов регрессий; SS_{resid} — остаточная сумма квадратов.

D2:E6						
f(x) Σ = {=LINEST(B2:B8;A2:A8;1;1)}						
	A	B	C	D	E	F
1	X	Y				
2	4	100		4,857142857	77,571428571	
3	5	105		0,914732034	6,659349726	
4	6	104		0,849375459	4,840306956	
5	7	108		28,195121951	5,000000000	
6	8	111		660,571428571	117,142857143	
7	9	120				
8	10	133				
9						

В примере выше, ячейки **A2:B8** содержат значения x и y для набора точек. Формула **=LINEST(B2:B8; A2:A8; 1; 1)**, возвращает статистические данные для наиболее соответствующей линии (линии регрессии), проведенной через эти точки.

E2:G6							
f(x) Σ = {=LINEST(A2:A8;B2:C8;1;1)}							
	A	B	C	D	E	F	G
1	value 000s	floor_area	num_windows				
2	377	191	10		13,55	0,84	80,77
3	412	214	11		1,78	0,12	8,95
4	422	216	12		1	2,16	#N/A
5	455	236	13		1722,91	4	#N/A
6	486	258	14		16042,81	18,62	#N/A
7	488	260	14				
8	526	270	16				
9							

В примере выше, Вы измеряете закрытую площадь и подсчитываете окна выборки зданий в районе, и делаете таблицу с соответствующей продажной стоимостью (ячейки **A2:C8**). Чтобы предсказывать стоимость других зданий в районе, Вы можете использовать:

$$value = a + b_1 * floor_area + b_2 * num_windows$$

где a , b_1 и b_2 — константы. **LINEST(A2:A8; B2:C8; 1; 1)** возвращает соответствующие статистические данные для этого уравнения.

Для правильного использования функции **LINEST** Вам необходимо хорошее понимание сложной теории статистики.

Пустые ячейки в выходном массиве отображают #N/A (в Calc и Excel).

LOGEST

Функция **LOGEST** возвращает таблицу статистических данных для показательной кривой, которая лучше всего соответствует набору данных.

Синтаксис функции:

=LOGEST(yvalues; xvalues; allow_const; stats)

где:

- **yvalues** — однострочковый или одностолбцовый диапазон, определяющий координаты *y* в наборе точек данных;
- **xvalues** — соответствующий однострочковый или одностолбцовый диапазон, определяющий координаты *x*. Если **xvalues** опущен, то по умолчанию принимаются значения 1, 2, 3..., *n*. Если имеется более чем один набор переменных, **xvalues** может быть диапазоном с соответствующими несколькими строками или столбцами.
- **allow_const** — если **FALSE**, константа задается равной единице, $y = b^x$, если опущен, по умолчанию — **TRUE** (ищется значение константы);
- **stats** — если опущен или **FALSE**, возвращается только верхняя строка таблицы статистических данных (рассчитывается только коэффициент регрессии), если **TRUE** возвращается вся таблица. Статистические данные касаются уравнения для $\ln(y)$ приведенного ниже.

Функция **LOGEST** находит показательную кривую $y = a \cdot b^x$, которая лучше всего соответствует данным. С более чем одним набором переменных кривая имеет форму

$$y = a \cdot b_1^{x_1} \cdot b_2^{x_2} \cdot \dots \cdot b_n^{x_n}$$

Для того чтобы подобрать кривую, функция **LOGEST** использует линейную регрессию (метод “наименьших квадратов”) основанный на следующем уравнении:

$$\ln(y) = \ln(a) + x_1 \ln(b_1) + x_2 \ln(b_2) + \dots + x_n \ln(b_n)$$

Функция **LOGEST** возвращает таблицу (массив) статистических данных, приведенных ниже, и должна быть введена как формула массива (например с использованием *Ctrl-Shift-Enter*, а не просто *Enter*).

b_n	b_{n-1}	...	b_1	a
σ_n	σ_{n-1}	...	σ_1	σ_a
r^2	σ_y			
F	df			
SS_{reg}	SS_{resid}			

где:

- $b_1 \dots b_n$ и a — коэффициенты для уравнения выше;
- $\sigma_1 \dots \sigma_n$ — значение среднеквадратичной погрешности для значения $\ln(b)$; σ_a — значение среднеквадратичной погрешности для значения $\ln(a)$;
- r^2 — коэффициент детерминации (RSQ); σ_y — значение среднеквадратичной погрешности для оценки $\ln(y)$;

- F — статистический критерий Фишера F (наблюдаемая величина F); df — число степеней свободы;
- SS_{reg} — сумма квадратов регрессий; SS_{resid} — остаточная сумма квадратов.

	A	B	C	D	E
1	X	Y			
2	1	0,76		2,519352739165	0,291424749009
3	2	1,8		0,010862188281	0,036025802930
4	3	4,5		0,999585590165	0,034349255341
5	4	11,9		7236,210426939820	3,000000000000
6	5	30		8,537797311056	0,003539614028
7					

В примере выше, ячейки **A2:B6** содержат значения x и y для набора точек. Формула **LOGEST(B2:B6; A2:A6; 1; 1)** возвращает статистические данные для наиболее соответствующей показательной кривой, проходящей через эти точки.

Для правильного использования функции **LOGEST** Вам необходимо хорошее понимание сложной теории статистики.

Пустые ячейки в выходном массиве отображают #N/A (в Calc и Excel).

MDETERM

Функция **MDETERM** возвращает детерминант матрицы.

Синтаксис функции:

=MDETERM(array)

где:

- **array** — квадратная матрица, которая может быть или встроенным массивом или диапазоном, содержащим все числа.

Функция **MDETERM** возвращает детерминант квадратной матрицы **array**. Функция **MDETERM** возвращает единственное значение. Она *не должна* вводиться как формула массива.

	A	B	C	D	E
1					
2				18	=MDETERM({4;1 2;5})
3					
4	4	2			
5	2	3			
6					
7				8	=MDETERM(A4:B5)
8					

В ячейке **B2** возвращается 18, то есть $4*5 - 1*2$.

В ячейке **B7** возвращается 8, детерминант квадратной матрицы, заданной диапазоном **A4:B5**, то есть $4*3 - 2*2$.

MINVERSE

Функция **MINVERSE** возвращает обратную матрицу.

Синтаксис функции:

=MINVERSE(array)

где:

- **array** — квадратная матрица, которая может быть или встроенным массивом или диапазоном, содержащим числа.

Функция **MINVERSE** возвращает обратную матрицу для квадратной матрицы **array**. Обратная матрица — такая матрица, при умножении на которую исходная матрица даёт в результате единичную матрицу. Матрица имеет обратную матрицу, тогда и только тогда, когда она невырожденная, т.е. её детерминант не равен нулю.

Функция **MINVERSE** возвращает массив, и должна быть введена как формула массива (например с использованием *Ctrl-Shift-Enter*, а не просто *Enter*).

	A	B	C	D	E
1					
2		1	-1		
3		-1.5	2		
4					

Когда формула **=MINVERSE({4;2|3;2})** введена как формула массива в ячейке **B2**, возвращается **{1; -1 | -1.5; 2}**, так, что ячейки **B2**, **C2**, **B3**, **C3** отображают 1, -1, -1,5 и 2 соответственно.

	A	B	C	D	E
1	4	2			
2	3	2			
3					
4	1	-1			
5	-1.5	2			
6					

Когда формула **=MINVERSE(A1:B2)** введена как формула массива в ячейке **A4**, где ячейки **A1**, **B1**, **A2**, **B2** содержат 4, 2, 3, 2 соответственно, возвращается 1, -1, -1,5 и 2 в ячейках **A4**, **B4**, **A5**, **B5** соответственно.

MMULT

Функция **MMULT** возвращает обычное произведение двух матриц.

Синтаксис функции:

=MMULT(array1; array2)

где:

- **array1** и **array2** — перемножаемые матрицы, могут быть или встроенным массивом или диапазоном, содержащим числа.

Функция **MMULT** перемножает матрицы **array1** и **array2** и возвращает в результате матрицу. Количество столбцов в матрице **array1** должно быть таким же, как количество строк в матрице **array2**.

Функция **MMULT** возвращает массив с таким же количеством строк, как у матрицы **array1** и количеством столбцов, как у матрицы **array2**. Функция **MMULT** должна быть введена как формула массива (например с использованием *Ctrl-Shift-Enter*, а не просто *Enter*).

		f(x) Σ = {=MMULT({1;2;3};{4;5 6;7 8;9})}				
	A	B	C	D	E	
1						
2		40	46			
3						

Когда формула **=MMULT({1;2;3};{4;5|6;7|8;9})** введена как формула массива в ячейке **B2**, возвращает 40 в ячейке **B2** и 46 в ячейке **C2**. Матрица 1×3 была умножена на матрицу 3×2 , давая результирующую матрицу 1×2 , вычисленную как

$$\mathbf{B2} = 1*4 + 2*6 + 3*8 = 40 \text{ и } \mathbf{C2} = 1*5 + 2*7 + 3*9 = 46.$$

MUNIT

Функция **MUNIT** возвращает единичную матрицу заданного размера.

Синтаксис функции:

=MUNIT(size)

где:

- **size** — размер возвращаемой матрицы, целое число, большее чем ноль.

Функция **MUNIT** возвращает единичную матрицу, также известную как единичная матрица **I**, размера **size**. Единичная матрица — квадратная матрица, элементы главной диагонали которой равны единице, а остальные — нулю.

$$[1] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Функция **MUNIT** возвращает массив, и должна быть введена как формула массива (например с использованием *Ctrl-Shift-Enter*, а не просто *Enter*).

		f(x) Σ = {=MUNIT(2)}				
	A	B	C	D	E	
1						
2		1	0			
3		0	1			
4						

Когда формула **=MUNIT(2)** введена как формула массива в ячейке **B2**, возвращается массив **{1; 0|0; 1}**, так, чтобы ячейки **B2**, **C2**, **B3**, **C3** отображали 1, 0, 0 и 1 соответственно.

SUMPRODUCT

Функция **SUMPRODUCT** возвращает сумму произведений соответствующих элементов массивов.

Синтаксис функции:

=SUMPRODUCT(array1; array2; ... array30)

где:

- **array1; array2; ... array30** — до 30 массивов или диапазонов одинакового размера, соответствующие элементы которых должны быть перемножены.

Функция **SUMPRODUCT** возвращает $\sum_i array1_i \cdot array2_i \cdot array3_i \cdot \dots$ для всех элементов массивов. Вы можете использовать функцию **SUMPRODUCT** для вычисления скалярного

произведения двух векторов.

Функция **SUMPRODUCT** расценивает каждый параметр от **array1** до **array30** как формулу массива, но не должен вводиться как формула массива. Другими словами она может быть введена клавишей *Enter*, а не *Ctrl-Shift-Enter*. См. примеры ниже.

Функция **SUMPRODUCT** может также использоваться для суммирования ячеек, которые удовлетворяют заданному условию - см. *Условный подсчет и суммирование* и пример ниже.

	A	B	C	D	E	F	G
1	1	2				5	6
2	3	4				7	8
3							
4	70 =SUMPRODUCT(A1:B2;F1:G2)						
5							

Возвращается $A1 * F1 + B1 * G1 + A2 * F2 + B2 * G2$.

	A	B	C	D	E	F
1	2					
2	-3					
3	8					
4	4					
5	-5					
6	-2					
7						

24 =SUMPRODUCT(ABS(A1:A6))

Когда формула **=SUMPRODUCT(ABS(A1:A6))** введена “обычно” (а не как формула массива), возвращает сумму абсолютных значений в ячейках **A1:A6**. Функция **SUMPRODUCT** заставляет **ABS (A1:A6)** вычисляться как формула массива.

	A	B	C	D	E	F	G
1	красный	большой	8				
2	синий	маленький	5				
3	зеленый	большой	2				
4	синий	маленький	11				
5	красный	большой	6				
6	красный	маленький	4				
7			14				
8	=SUMPRODUCT(A1:A6="красный";B1:B6="большой";C1:C6)						
9							

Формула **=SUMPRODUCT(A1:A6="красный"; B1:B6="большой"; C1:C6)** возвращает сумму ячеек в диапазона **C1:C6**, соответствующие элементы для которых в столбце А имеют значение “красный”, а в столбце В — “большой”. Этот вариант формулы не переносим в Excel, поскольку Excel игнорирует логические значения.

	A	B	C	D	E	F	G
1	красный	большой	8				
2	синий	маленький	5				
3	зеленый	большой	2				
4	синий	маленький	11				
5	красный	большой	6				
6	красный	маленький	4				
7			14				
8	=SUMPRODUCT((A1:A6="красный") * (B1:B6="большой") * C1:C6)						
9							

Вариант **SUMPRODUCT(A1:A6 = "красный"; B1:B6 = "большой"; C1:C6)**, переносима в Excel, так как логические значения преобразуются в числовые значения при умножении.

В Excel функция **SUMPRODUCT** игнорирует логические значения (то есть, она

рассматривает их как 0). В Calc TRUE — 1, и FALSE — 0.

SUMX2MY2

Функция **SUMX2MY2** возвращает сумму разности между квадратами соответствующих элементов двух матриц.

Синтаксис функции:

=SUMX2MY2(x; y)

где:

- **x** и **y** — массивы или диапазоны одинакового размера.

Функция **SUMX2MY2** вычисляет $\sum_i (x_i^2 - y_i^2)$ для всех элементов в массивах или диапазонах.

Функция **SUMX2MY2** вычисляет свои параметры **x** и **y** как формулы массива, но не должна вводиться как формула массива. Другими словами она может быть введена клавишей *Enter*, а не *Ctrl-Shift-Enter*. См. пример ниже.

	A	B	C	D	E
1	4				
2	3	20	=SUMX2MY2(A1:A2;{2 1})		
3					

Когда ячейки **A1** и **A2** содержат, соответственно, 4 и 3, формула **=SUMX2MY2(A1:A2; {2|1})** в ячейке **B2** возвращает $(4^2 - 2^2) + (3^2 - 1^2) = 20$.

	A	B	C	D	E
1	3				
2	2	20	=SUMX2MY2(A1:A2+1;{2 1})		
3					

Когда формула **=SUMX2MY2(A1:A2+1; {2|1})** введена “обычно” (а не как формула массива), и когда ячейки **A1** и **A2** содержат, соответственно, 3 и 2, возвращает $((3+1)^2 - 2^2) + ((2+1)^2 - 1^2) = 20$. Функция **SUMX2MY2** принимает аргумент (A1:A2+1), который вычисляется как формула массива (внутренне возвращающая массив {4|3}).

SUMX2PY2

Функция **SUMX2PY2** возвращает сумму квадратов всех элементов двух матриц.

Синтаксис функции:

=SUMX2PY2(x; y)

где:

- **x** и **y** — массивы или диапазоны одинакового размера.

Функция **SUMX2PY2** вычисляет $\sum_i (x_i^2 + y_i^2)$ для всех элементов в массивах или диапазонах.

Функция **SUMX2PY2** вычисляет свои параметры **x** и **y** как формулы массива, но не должна вводиться как формула массива. Другими словами она может быть введена клавишей *Enter*, а не *Ctrl-Shift-Enter*. См. пример ниже.

	A	B	C	D	E
1	4				
2	3	30	=SUMX2PY2(A1:A2;{2 1})		
3					

Когда ячейки **A1** и **A2** содержат, соответственно, 4 и 3, формула **=SUMX2PY2(A1:A2; {2|1})** в ячейке **B2** возвращает $(4^2+2^2) + (3^2+1^2) = 30$.

	A	B	C	D	E
1	4				
2	3	30	=SUMX2PY2(A1:A2+1;{2 1})		
3					

Когда формула **=SUMX2PY2(A1:A2+1; {2|1})** введена “обычно” (а не как формула массива), и когда ячейки **A1** и **A2** содержат, соответственно, 3 и 2, возвращает $((3+1)^2+2^2) + ((2+1)^2+1^2) = 30$. Функция **SUMX2PY2** принимает аргумент (A1:A2+1), который вычисляется как формула массива (внутренне возвращающая массив {4|3}).

SUMXMY2

Функция **SUMXMY2** возвращает сумму квадратов разностей между соответствующими элементами двух матриц.

Синтаксис функции:

=SUMXMY2(x; y)

где:

- **x** и **y** — массивы или диапазоны одинакового размера.

Функция **SUMXMY2** вычисляет $\sum_i (x_i - y_i)^2$ для всех элементов в массивах или диапазонах.

Функция **SUMXMY2** вычисляет свои параметры **x** и **y** как формулы массива, но не должна вводиться как формула массива. Другими словами она может быть введена клавишей **Enter**, а не **Ctrl-Shift-Enter**. См. пример ниже.

	A	B	C	D	E
1	4				
2	3	8	=SUMXMY2(A1:A2;{2 1})		
3					

Когда ячейки **A1** и **A2** содержат, соответственно, 4 и 3, формула **=SUMXMY2(A1:A2; {2|1})** в ячейке **B2** возвращает $(4 - 2)^2 + (3 - 1)^2 = 8$.

	A	B	C	D	E
1	3				
2	2	8	=SUMXMY2(A1:A2+1;{2 1})		
3					

Когда формула **=SUMXMY2(A1:A2+1; {2|1})** введена “обычно” (а не как формула массива), и когда ячейки **A1** и **A2** содержат, соответственно, 3 и 2, возвращает $((3+1) - 2)^2 + ((2+1) - 1)^2 = 8$. Функция **SUMXMY2** принимает аргумент (A1:A2+1), который вычисляется как формула массива (внутренне возвращающая массив {4|3}).

TRANSPOSE

Функция **TRANSPOSE** возвращает массив, в котором строки и столбцы меняются местами.

Синтаксис функции:

=TRANSPOSE(array)

где:

- **array** — встроенный массив или диапазон, содержащий числа.

Функция **TRANSPOSE** выполняет транспонирование матрицы, заданной встроенным

массивом или диапазоном.

Функция **TRANSPOSE** возвращает массив, и должна быть введена как формула массива (например с использованием *Ctrl-Shift-Enter*, а не просто *Enter*).

		f(x) Σ = {=TRANSPOSE({1;2;3 4;5;6})}				
	A	B	C	D	E	
1						
2		1	4			
3		2	5			
4		3	6			
5						

Когда формула `=TRANSPOSE({1;2;3|4;5;6})` введена как формула массива в ячейке **B2**, она возвращает массив `{1; 4|2; 5|3; 6}`, так, чтобы ячейки **B2**, **C2** отображали 1 и 4, ячейки **B3**, **C3** отображали 2 и 5, а ячейки **B4**, **C4** отображали 3 и 6 соответственно.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

		f(x) Σ = {=TRANSPOSE(A1:C2)}				
	A	B	C	D	E	
1	1	2	3			
2	4	5	6			
3						
4						
5		1	4			
6		2	5			
7		3	6			
8						

Когда формула `=TRANSPOSE(A1:C2)` введена как формула массива в ячейке **B5**, она возвращает массив `{1; 4|2; 5|3; 6}`, так, чтобы ячейки **B5**, **C5** отображали 1 и 4, ячейки **B6**, **C6** отображали 2 и 5, а ячейки **B7**, **C7** отображали 3 и 6 соответственно.

TREND

Функция **TREND** подгоняет прямую линию к данным, используя линейную регрессию и возвращает точки на той линии.

Синтаксис функции:

`=TREND(yvalues; xvalues; new_xvalues; type)`

где:

- **yvalues** и **xvalues** — одностроковые или одностолбцовые диапазоны, определяющие точки в наборе данных;
- **new_xvalues** — одностроковый или одностолбцовый диапазон, определяющий положение точек в выходном наборе данных;
- **type** - Если 0, то искомая прямая линия проходит через начало координат; если отличен от нуля или опущен, то будет искажаться наиболее соответствующая прямая линия.

Функция **TREND** подбирает прямую линию через заданные точки данных, используя метод линейной регрессии.

Функция **TREND** возвращает массив значений *y* найденной прямой линии, соответствующих значениям *x* в **new_xvalues** (или если опущено, в **xvalues**). Она должна вводиться как

формула массива (например с использованием *Ctrl-Shift-Enter*, а не просто *Enter*).

Диапазоны **yvalues** и **xvalues** должны быть одинакового размера. Диапазон **new_xvalues** может иметь отличающийся размер.

	A	B	C	D	E
1	X	Y	Новые Y		
2		1	2	2,01	
3		2	4	4,02	
4		3	6,1	6,03	
5		4	8	8,04	
6		5		10,05	
7		6		12,06	
8					

Когда формула **=TREND(B2:B5; A2:A5; A2:A7)** введена как формула массива в ячейке **C2**, где значения **x** в диапазоне **A2:A7** — 1, 2, 3, 4, 5 и 6, а значения **y** в диапазоне **B2:B4** — 2, 4, 6,1 и 8, возвращает массив **{2,01|4,02|6,03|8,04|10,05|12,06}**. Точки данных располагаются почти на линии $y = 2x$ (и это было бы так, если бы **B4** содержала 6 вместо 6,1). Наиболее соответствующая найденная линия, поэтому, почти $y = 2x$.

Этот пример показывает, как функция **TREND** может использоваться для предсказания будущих значений.

Список литературы

1. Open Document Format for Office Applications (OpenDocument)v1.2. Part 2: Recalculated Formula (OpenFormula) Format. Pre-Draft10, 10 Oct 2008
2. Documentation/How Tos/Calc: Calc Function Descriptions (http://wiki.services.openoffice.org/wiki/Documentation/How_Tos/Calc:_Functions_listed_by_category)
3. OpenOffice Calc Tips (<http://www.openofficetips.com/>)
4. Introduction to Statistics Using OpenOffice.org Calc (<http://www.comfsm.fm/~dleeling/statistics/notes000.html>)
5. Andrew Pitonyak. Calc As A Simple Database.
6. OpenOffice.org Руководство по Calc
7. OpenOfficeSchool.org. Learn how to use OpenOffice.org
8. Губкина Г.Е. Автоматизации операций с потоками платежей в среде OpenOffice.org.
9. Смирнова И.И. Автоматизация финансовых вычислений в среде Openoffice.org.