

# Gnumeric: электронная таблица для всех

И.А.Хахаев, © 2007-2010

## 2 Основы работы в Gnumeric (продолжение)

### 2.4 Автозаполнение: генерация рядов данных

Рассмотрим возможности Gnumeric по генерации исходных данных.

Исходные данные, как правило, располагаются в непрерывном блоке ячеек (столбце или строке) и затем используются для каких-либо вычислений. Для определенности будет располагать исходные данные в столбце.

Для создания последовательностей исходных данных (векторов, таблиц) можно использовать возможности заполнения блоков данных, используя пункт главного меню «Данные» (рис. 2.1).

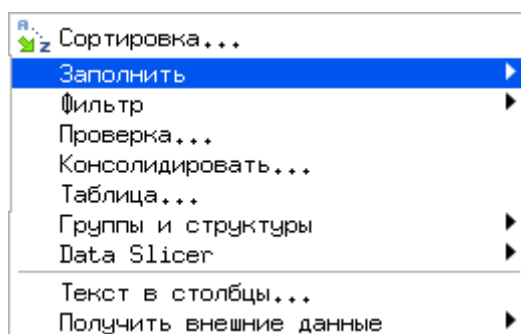


Рисунок 2.1. Пункт «Данные» главного меню Gnumeric

Вложенное меню «Заполнить» как раз и предоставляет различные способы создания блоков ячеек с исходными данными (рис. 2.2). Нужно иметь в виду, что до версии 1.9.9 вложенное меню «Заполнить» может находиться в пункте главного меню «Правка».

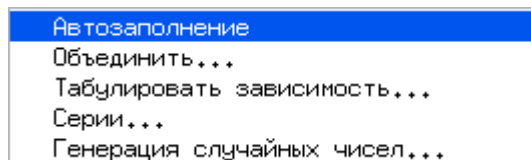


Рисунок 2.2. Вложенное меню «Заполнить»

Здесь мы рассмотрим только три варианта – генерация последовательностей («Автозаполнение»), создание прогрессий («Серии...») и создание выборок последовательностей случайных чисел с различными видами функций распределения («Генерация случайных чисел...»).

Для начала сформируем последовательность натуральных чисел от 0 до 25. Для этого введем в ячейку A3 значение «0», а в A4 – значение «1» (без кавычек!),

затем выделим ячейки от A3 до A28 (всего 25 ячеек) путем «протаскивания» мыши с нажатой левой кнопкой. Далее вызовем функцию автозаполнения ряда: «Правка/Заполнить/Автозаполнение» и пронаблюдаем результат.

Далее сформируем последовательность из 25 чисел, кратных 7, начиная с 0. По аналогии с предыдущим случаем, введем в ячейку B3 значение «0», а в ячейку B4 – значение «1», после чего повторим операции выделения и вызова функции автозаполнения и посмотрим на результат.

Такую же операцию можно провести с датами. Пусть известно, что 21 мая 2007 года и 28 мая этого же года были понедельниками. На какие даты будут приходиться понедельники в течение последующих 23 недель? (Аналогично предыдущему случаю, только вместо чисел – даты). Соответственно, вводим в ячейку C3 значение «21/05/07», в C4 – «28/05/07», выделяем снова 25 ячеек и вызываем автозаполнение. Затем устанавливаем формат дат как «dd/mm/yy» с помощью диалога «Формат ячеек». Результат показан на рис. 2.3.

Нужно отметить, что с другими (не числовыми) данными функция «Автозаполнение» не работает.

Таким образом, функция «Автозаполнение» продолжает зависимость, определенную по первым двум ячейкам диапазона, на все ячейки выделенного диапазона.

	А	В	С	D
1	<b>Использование Автозаполнения</b>			
2	через 1	через 7	даты	через 7
3	0	0	21/05/07	
4	1	7	28/05/07	
5	2	14	04/06/07	
6	3	21	11/06/07	
7	4	28	18/06/07	
8	5	35	25/06/07	
9	6	42	02/07/07	
10	7	49	09/07/07	
11	8	56	16/07/07	
12	9	63	23/07/07	
13	10	70	30/07/07	
14	11	77	06/08/07	
15	12	84	13/08/07	
16	13	91	20/08/07	
17	14	98	27/08/07	
18	15	105	03/09/07	
19	16	112	10/09/07	
20	17	119	17/09/07	
21	18	126	24/09/07	
22	19	133	01/10/07	
23	20	140	08/10/07	
24	21	147	15/10/07	
25	22	154	22/10/07	
26	23	161	29/10/07	
27	24	168	05/11/07	
28	25	175	12/11/07	

Рисунок 2.3. Результаты использования Автозаполнения

Следующая возможность автоматической генерации исходных данных – создание серий. Оно может потребоваться, если заранее неизвестно количество ячеек для автозаполнения, или если это количество велико, а также если зависимость значения следующей ячейки от предыдущей не является линейной.

Рассмотрим примеры генерации серий чисел со значениями от 1 до 100. Диалог заполнения ячеек сериями значений («Правка/Заполнить/Серии...») содержит три вкладки (рис. 2.4).

- Вкладка «Серии» позволяет установить основные параметры серии. Здесь определяется направления заполнения (по строке или по столбцу), вид заполнения (линейный или рост с заданным коэффициентом), а также начальное значение, приращение и конечной значение. Не все эти параметры обязательно указывать, однако начальное значение указывать обязательно, а конечное значение и приращение указываются в зависимости от того, известны или нет количество ячеек в серии и максимальное значение элемента серии.

- Вкладка «Параметры» активируется, если для формирования серии выбраны даты. В этом случае указывается единица приращения дат (календарный день, рабочий день, месяц или год).
- Вкладка «Вывод» позволяет указать диапазон вывода полученной серии значений. По умолчанию вывод происходит в выделенный диапазон ячеек (если он есть) или начиная с активной ячейки. Однако можно сформировать серию на новом листе или открыть новый документ Gnumeric (книгу) и создать серию значений там.

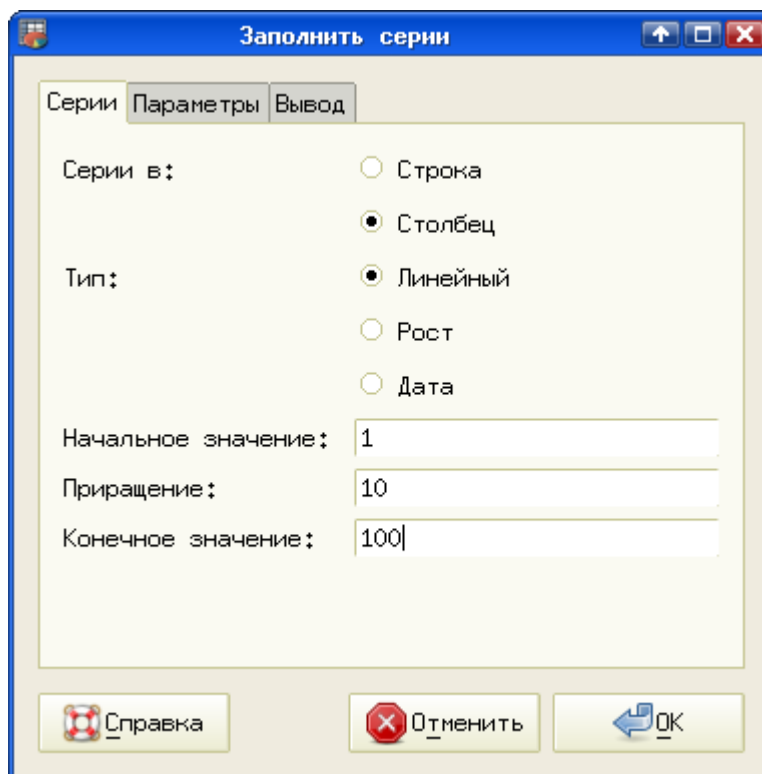


Рисунок 2.4. Настройка серий для заполнения ячеек данными

Сформируем серию значений, линейно изменяющихся по столбцу (арифметическую прогрессию). Устанавливаем соответствующие параметры на вкладке «Серии», указываем в качестве начального значения число 1, приращение 10 и конечное значение 100 (рис. 2.4). Больше никаких дополнительных настроек не требуется. После нажатия на <ENTER> (или использования кнопки «ОК» в диалоге) наблюдаем результат (рис. 2.5). Видно, что последнее значение серии меньше указанного максимального значения. Однако если провести эксперимент с начальным значением, равным 0, то последним значением в серии будет 100. Таким образом, можно сделать очевидный вывод, что конечное значение в серии никогда не превышает указанного максимального значения.

Теперь в соседних столбцах создадим серии с ростом в 2 и в 3 раза при тех же начальных и конечных значениях (именно поэтому в качестве начального значение выбрана 1, а не 0). Коэффициент роста указывается как приращение.

Результаты показаны на рис. 2.5).

	А	В	С	D
1	<b>Заполнение сериями</b>			
2	Больше на 10	Рост в 2 раза	Рост в 3 раза	
3	1	1	1	
4	11	2	3	
5	21	4	9	
6	31	8	27	
7	41	16	81	
8	51	32		
9	61	64		
10	71			
11	81			
12	91			
13				

Рисунок 2.5. Результат заполнения сериями

С заполнением диапазона ячеек серией дат читателям предлагается разобраться обстоятельно.

Следующая интересная возможность генерации исходных данных – заполнение диапазона ячеек случайными числами с различными функциями плотности распределения. Всего предлагается около 30 вариантов. Для использования этой возможности полезно иметь выделенный диапазон ячеек (блок). Можно реализовать двумерное поле случайных чисел, если выделить не часть строки или столбца, а прямоугольный диапазон и сделать соответствующие настройки в диалоге «Генерация случайных чисел» («Правка/Заполнить/Генерация случайных чисел...», рис. 2.6).

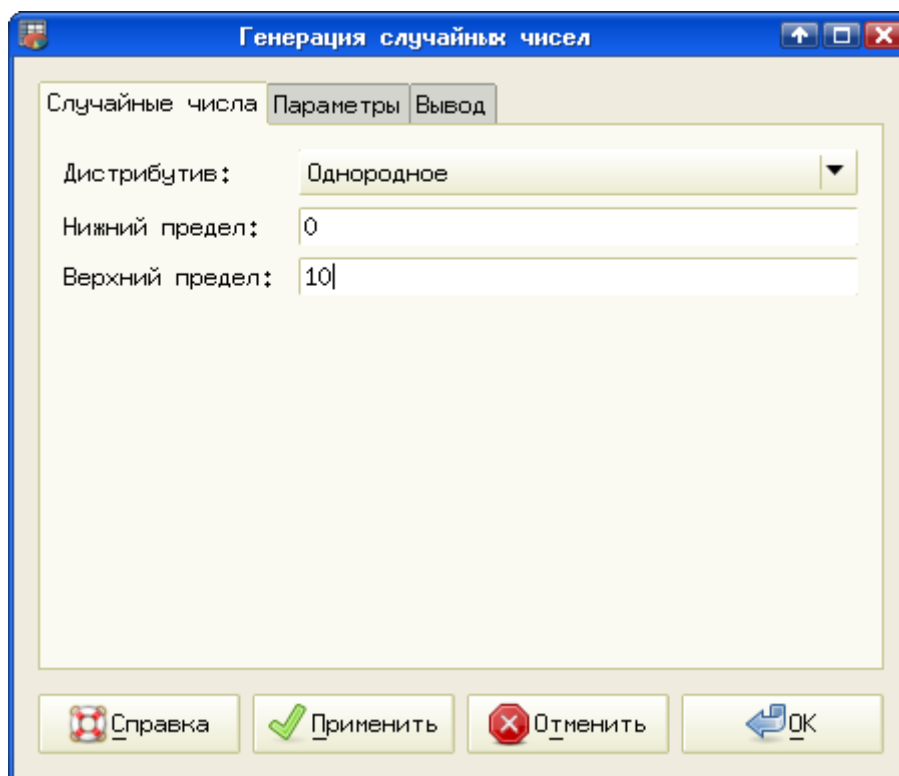


Рисунок 2.6. Диалог генерации случайных чисел (равномерное распределение)

Диалог «Генерация случайных чисел» также имеет три вкладки.

- Вкладка «Случайные числа» позволяет определить вид функции плотности распределения (неудачно названный словом «дистрибутив»), а также параметры этой функции. Для разных распределений количество и значения параметров будут разными.
- Вкладка «Параметры» позволяет определить размер выборки (сколько чисел нужно) и количество переменных (1-мерная последовательность или 2-мерное поле). Если предварительно выделен диапазон ячеек, размер выборки устанавливается автоматически.
- Вкладка «Вывод» позволяет (как и в предыдущем случае) указать диапазон вывода полученной серии значений. По умолчанию вывод происходит в выделенный диапазон ячеек (если он есть) или начиная с активной ячейки. Однако можно сформировать серию на новом листе или открыть новый документ Gnumeric (книгу) и создать серию значений там.

В качестве примера создадим три вектора из 100 случайных чисел с разными распределениями: равномерным в диапазоне от 0 до 10 (как на рис. 2.6), нормальным со средним значением 5 и стандартным отклонением 2, а также с распределением  $\chi^2$  со значением параметра  $\nu=2$ . Для полученных результатов установим формат с 4-мя десятичными знаками (рис. 2.7)

	А	В	С	D
1	<b>Генерация случайных чисел</b>			
2	Равномерное	Нормальное	Хи-квадрат	
3	3,5215	4,1070	1,0284	
4	2,2043	2,9831	0,8274	
5	5,4872	6,9444	1,1866	
6	9,6423	5,1187	0,1518	
7	1,6817	6,7977	0,6176	
<b>8</b>	7,0084	8,4337	<b>2,0553</b>	
9	7,2888	9,3311	0,9406	
10	0,7312	1,9206	0,2733	
11	5,7347	3,9104	0,7215	
12	2,2740	4,8930	1,4979	
13	7,4336	3,2952	0,6764	
14	4,5522	5,7443	0,1311	
15	4,2461	2,8635	2,5090	
16	8,8224	6,4100	2,9347	
17	0,2124	4,9169	1,1188	
18	4,4424	4,7765	0,4385	
19	7,4021	4,5417	2,7815	
20	5,3035	6,1519	0,6666	
21	7,9344	2,3908	2,6562	
22	4,1669	4,9635	8,2444	

Рисунок 2.7. Три вектора случайных чисел

Рассмотренные возможности по генерации исходных данных будут полезны при дальнейшей работе с пакетом Gnumeric.

## 2.5 Формулы. Абсолютная и относительная адресация.

Формулы в электронных таблицах предназначены для вычислений значений в ячейках таблицы на основе данных, записанных в другие ячейки. Результатом работы формулы может быть число, дата, текст или отсутствие данных (т.е. после вычислений можно получать пустые ячейки). Формула записывается в той ячейке, в которой должен быть результат. Ввод формулы начинается с нажатия на символ «=» на клавиатуре, после чего вводятся числа, адреса ячеек и знаки операций. Например, формула «=3+5» даст всегда результат «9», а формула «=A3/12» будет давать различные результаты при изменении значения в ячейке A3. Таким образом, при изменении содержимого ячеек, адреса которых используются в формулах, результаты пересчитываются автоматически. Эта особенность является ключевой для всех электронных таблиц. Из-за этого свойства крайне не рекомендуется использовать в формулах конкретные значения, если только они не являются неотъемлемой частью правил вычисления (например, если площадь круга вычисляется в евклидовой геометрии как  $\pi R^2$ , то двойка может использоваться в формуле вычисления площади круга, а вот конкретное значение радиуса – нет).

Электронная таблица с хорошо составленными формулами представляет из себя мощное средство решения вычислительных задач и, по сути, является программой, выдающей результаты на основе содержимого ячеек с исходными данными.

Теперь рассмотрим некоторые примеры использования формул и адресации данных в таблицах Gnumeric.

Начнем вычисления в ЭТ с простой задачи. Пусть имеется список из 7 человек, для каждого из которых известны фамилия, инициалы, должность, оклад за день работы и число отработанных дней. Требуется вычислить заработок каждого лица.

Формируем таблицу, начиная с ячейки А3, в соответствии с рис. 2.8. Для исправления ошибок в ячейках электронной таблицы используется режим редактирования строки ввода, который включается клавишей <F2>. Завершение редактирования обеспечивается клавишами <ENTER> (с сохранением изменений) или <ESC> (без сохранения изменений).

Фамилия, инициалы	Должность	Оклад за день	Дней работы	Заработок
Сионов К.С.	Бухгалтер	50	22	
Чернышевская С.П.	Швея	43	25	
Чкалов В.П.	Плотник	45	15	
Романов П.А.	Мореплаватель	70	31	
Штек Е.А.	Уборщица	22	25	
Зубарева О.А.	Ветеринар	34	8	
Жеребцова Н.Н.	Швея	40	25	
Зубов П.Р.	Директор	85	30	

Рисунок 2.8. Исходные данные для задачи о заработках

Если при вводе информации ширина ячейки представляется недостаточной, ее можно скорректировать после завершения ввода всех данных. Как уже упоминалось выше, всегда есть возможность подобрать ширину столбца автоматически двойным щелчком мышкой на правой границе столбца в строке имен столбцов.

Для вычисления заработка нужно просто перемножить попарно числа из третьей (столбец С) и четвертой (столбец D) колонок. Результаты вычислений должны быть в пятой колонке (столбец E). С учетом возможностей ЭТ, формулу (т.е. правила) для вычислений можно написать один раз, а потом скопировать. Формулу надо писать там, где должен появиться первый результат (в нашем примере – в ячейке E4, под заголовком «Заработок»). Переводим указатель активной ячейки в клетку E4 и нажимаем клавишу «=» (указание на начало ввода формулы). После этого щелкаем левой кнопкой по ячейке, в которой записан оклад за день (C4), нажимаем на клавиатуре знак операции (умножение – «\*») и щелкаем левой кнопкой по ячейке с количеством отработанных дней (D4), после чего нажимаем <ENTER>. В ячейке E4 появляется результат (число 1100), а переместив указатель активной ячейки на E4, в строке ввода можно увидеть формулу =C4\*D4

Теперь скопируем эту формулу в оставшиеся ячейки. Поместив указатель активной ячейки на E4, наведем мышку на нижний правый угол указателя активной ячейки (там есть маленький черный квадратик). Двойной щелчок мышкой в этой позиции приведет к автоматическому копированию формулы до конца блока



исходных данных. Результаты показаны на рис. 2.9.

Фамилия, инициалы	Должность	Оклад за день	Дней работы	Заработок
Сионов К.С.	Бухгалтер	50	22	1100
Чернышевская С.П.	Швея	43	25	1075
Чкалов В.П.	Плотник	45	15	675
Романов П.А.	Мореплаватель	70	31	2170
Штек Е.А.	Уборщица	22	25	550
Зубарева О.А.	Ветеринар	34	8	272
Жеребцова Н.Н.	Швея	40	25	1000
Зубов П.Р.	Директор	85	30	2550

Рисунок 2.9. Результат расчёта заработка

Если изменить какие-то числа в столбцах С и D, то числа в столбце Е будут автоматически пересчитываться.

Перемещая указатель активной ячейки по столбцу Е, можно заметить, что адреса ячеек в расчетной формуле изменяются. Это происходит потому, что в нашей формуле использованы **относительные** адреса ячеек. Формула просто перемножает содержимое ячеек, находящихся слева от ячейки с результатом. Таким образом, формула «запомнила» взаимное расположение ячеек с данными и с результатом и при копировании это взаимное расположение сохраняется. Это очень полезное свойство ЭТ, избавляющее от необходимости писать одну и ту же формулу много раз.

Если в какой-либо ячейке расчетного столбца (столбца «Заработок») перейти в режим редактирования (<F2>), то можно увидеть формулу. При перемещении текстового курсора по формуле будут подсвечиваться ячейки, содержащие данные для формулы (рис. 2.10).

Фамилия, инициалы	Должность	Оклад за день	Дней работы	Заработок
Сионов К.С.	Бухгалтер	50	22	1100
Чернышевская С.П.	Швея	43	25	=C5*D5
Чкалов В.П.	Плотник	45	15	675
Романов П.А.	Мореплаватель	70	31	2170
Штек Е.А.	Уборщица	22	25	550
Зубарева О.А.	Ветеринар	34	8	272
Жеребцова Н.Н.	Швея	40	25	1000
Зубов П.Р.	Директор	85	30	2550

Рисунок 2.10. Просмотр формулы в режиме редактирования

На следующем этапе посчитаем налог на доходы физических лиц, который будет начислен на рассчитанные ранее значения заработка. Пусть ставка налога

фиксирована и составляет 13%. Тогда наша таблица дополняется в соответствии с рис. 2.11 (здесь и в следующих иллюстрациях к этому примеру первый столбец «обрезан»).

В	С	Д	Е	Ф
	Ставка налога	13%		
Должность	Оклад за день	Дней работы	Заработок	Сумма налога
Бухгалтер	50	22	1100	
Швея	43	25	1075	
Плотник	45	15	675	
Мореплаватель	70	31	2170	
Уборщица	22	25	550	
Ветеринар	34	8	272	
Швея	40	25	1000	
Директор	85	30	2550	

Рисунок 2.11. Таблица для вычислений с параметром

Сумму налога легко сосчитать по правилу «Сумма налога = заработок\*ставка\_налога». Указав соответствующие адреса ячеек, в ячейке F4 записываем формулу =E4\*D1 и копируем ее во все оставшиеся ячейки. При этом получается неожиданный результат (рис. 2.12).

В	С	Д	Е	Ф
	Ставка налога	13%		
Должность	Оклад за день	Дней работы	Заработок	Сумма налога
Бухгалтер	50	22	1100	143
Швея	43	25	1075	0
Плотник	45	15	675	=E6*D3
Мореплаватель	70	31	2170	47740
Уборщица	22	25	550	13750
Ветеринар	34	8	272	4080
Швея	40	25	1000	31000
Директор	85	30	2550	63750

Рисунок 2.12. Неправильный результат вычислений с параметром

В этом случае использование относительной адресации привело к ошибке – забывчивость взаимное расположение ячеек результата и исходных данных (заработка первого в списке и ставки налога) программа ЭТ повторяет это взаимное расположение для остальных строк списка (в чем можно убедиться, войдя в режим редактирования, как показано на рис. 2.12). Чтобы не создавать дополнительный столбец с одним и тем же значением ставки налога, в соответствующей формуле надо использовать **абсолютный** адрес ячейки, содержащей параметр (в данном случае – значение ставки налога). Для указания абсолютного адреса к букве столбца или номеру строки добавляется префикс "\$" и формула для расчета суммы налога

приобретает вид =E4\*\$D\$1 (для добавления символов "\$" при редактировании формулы можно использовать клавишу <F4>). Отредактировав формулу в ячейке F4, копируем ее снова в оставшиеся ячейки и получаем правильный результат (рис. 2.13).

	В	С	Д	Е	Ф
		Ставка налога	13%		
Должность	Оклад за день	Дней работы	Заработок	Сумма налога	
Бухгалтер	50	22	1100	143	
Швея	43	25	1075	139,75	
Плотник	45	15	675	=E6*\$D\$1	
Мореплаватель	70	31	2170	282,1	
Уборщица	22	25	550	71,5	
Ветеринар	34	8	272	35,36	
Швея	40	25	1000	130	
Директор	85	30	2550	331,5	

Рисунок 2.13. Правильные вычисления с параметром

В режиме редактирования теперь видно, что во всех ячейках при вычислении суммы налога происходит обращение к ячейке, содержащей ставку налога, независимо от строки таблицы (вспомним, что одинаковое количество знаков после запятой во всех ячейках можно получить, изменив формат представления чисел).

Итак, абсолютный адрес указывает программе ЭТ, что нужно всегда обращаться к одной и той же ячейке (если поставлено два префикса \$), строке (если \$ поставлен перед номером строки) или столбцу (если \$ – перед буквой столбца). Использование абсолютных адресов позволяет работать с условно-постоянными величинами (ставка налога, курс валюты, текущая дата и пр.), причем их значения заносятся в таблицу только один раз, что экономит время и место.

Нужно заметить, что формулы позволяют связывать между собой не только ячейки в пределах одного листа, но и ячейки на разных листах документа. В результате можно строить «трехмерные» электронные таблицы.

Полезная и часто используемая возможность электронных таблиц – автосуммирование. Для использования этой возможности нужно установить указатель активной ячейки в позицию, в которой нужно получить результат и нажать на панели инструментов Gnumeric кнопку  $\Sigma$  (знак суммы). Программа автоматически определит непрерывный блок ячеек выше или слева от целевой и предложит вариант функции для вычисления результата. Обратите внимание, что диапазон ячеек указывается с использованием двоеточия, как показано на рис. 2.14.

	Заработок	Сумма налога
2	1100	143
5	1075	139,75
5	675	87,75
1	2170	282,1
5	550	71,5
8	272	35,36
5	1000	130
0	2550	331,5
		=sum(E4:E12)

Рисунок 2.14.  
Автосуммирование по столбцу

В качестве упражнения посчитайте аналогичным образом сумму всех налогов.

## 2.6 Функции. Селектор функций и помощник по формулам.

Для проведения математических, тригонометрических, экономических или инженерных вычислений четырех действий арифметики явно недостаточно. Поэтому все электронные таблицы имеют большое число встроенных функций, которые могут включаться в состав формул. Каждая функция имеет имя и список аргументов, которые помещаются в скобки сразу за именем функции. Могут быть функции с пустым множеством аргументов, такие как  $\text{pi}()$  или  $\text{today}()$ , а могут быть функции с неограниченным количеством аргументов (такие как  $\text{sum}(\dots)$  или  $\text{average}(\dots)$ ), однако для большинства функций количество аргументов фиксировано. В зависимости от назначения и синтаксиса функции, аргументом функции может быть число, текст, дата или логическое выражение (булево выражение). Чаще всего в качестве аргумента (или в составе аргумента) используются адреса ячеек. В случае нескольких аргументов разделителем аргументов является точка с запятой (символ «;»). Если в качестве аргумента используется текст, то он должен быть заключен в кавычки (символ «"»). Кроме того, в аргументах функций могут использоваться другие функции и арифметические выражения (например, возможны формулы типа « $=2*\sin(3*A3*\text{pi}()/4)$ »).

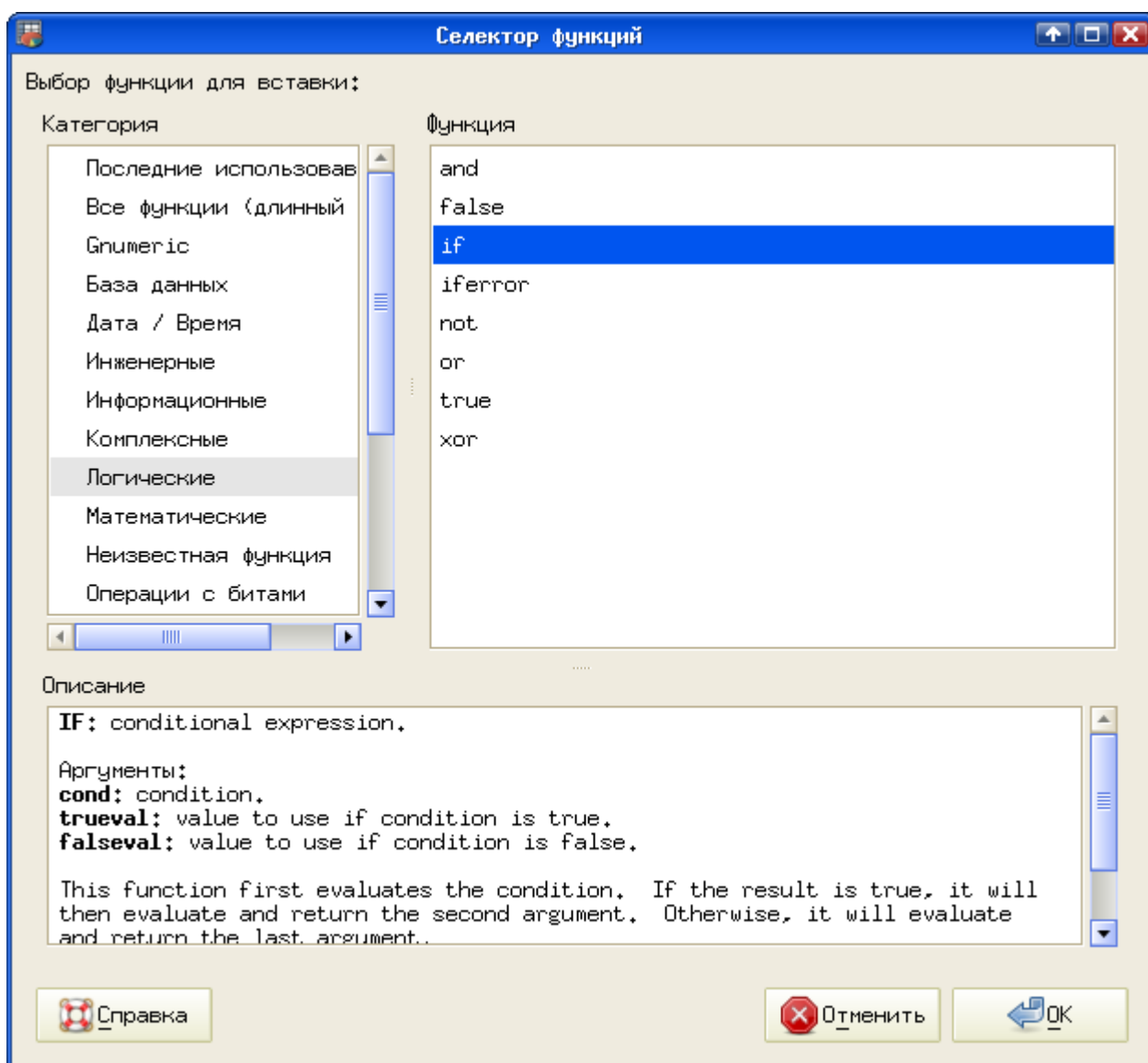


Рисунок 2.15. Диалог выбора функции (селектор функций)

Для вызова селектора встроенных функций используется либо главное меню («Вставка/Функция»), либо кнопка  $f(x)$  в панели инструментов. После этого появляется диалог выбора функции. На рис. 2.15 показан пример для логической функции  $if()$ , которая подробнее будет рассмотрена ниже.

При выборе функции сначала выбирается категория функций (в левой части диалога), а затем – конкретная функция (в правой части диалога). В нижней части диалога приводится объяснение назначения и структуры функции. После нажатия на кнопку «Вставка» появляется диалог определения аргументов выбранной функции (помощник по формулам, рис. 2.16). Для приведенной в примере функции  $if()$  нужно определить три аргумента. В данном случае проверяется ячейка C3. Если в ней содержится ненулевое значение, должно быть выведено сообщение «Не ноль!», в противном случае – сообщение «Ноль»

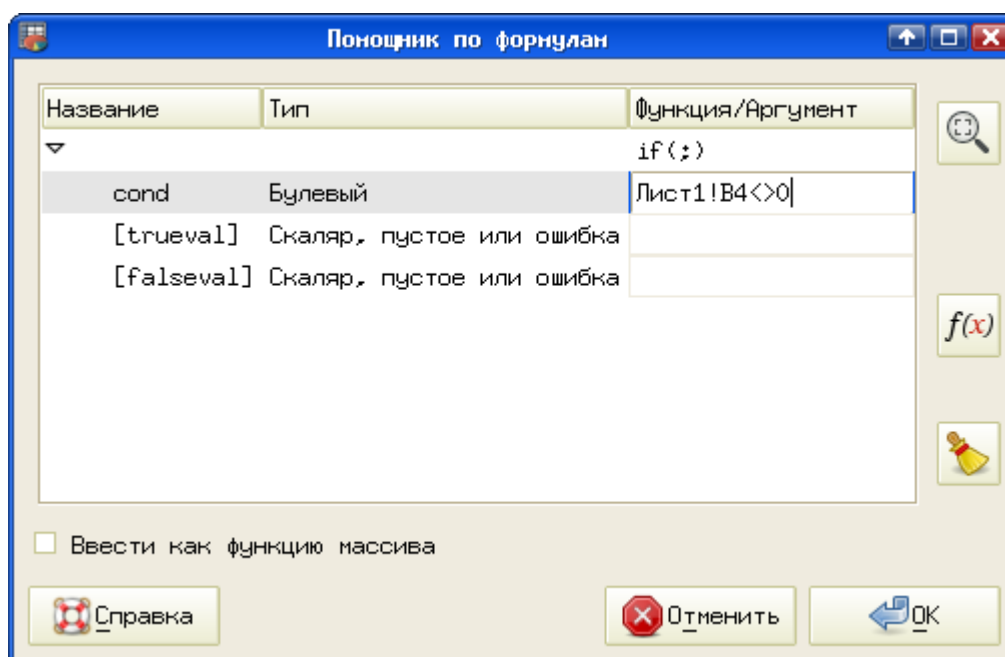
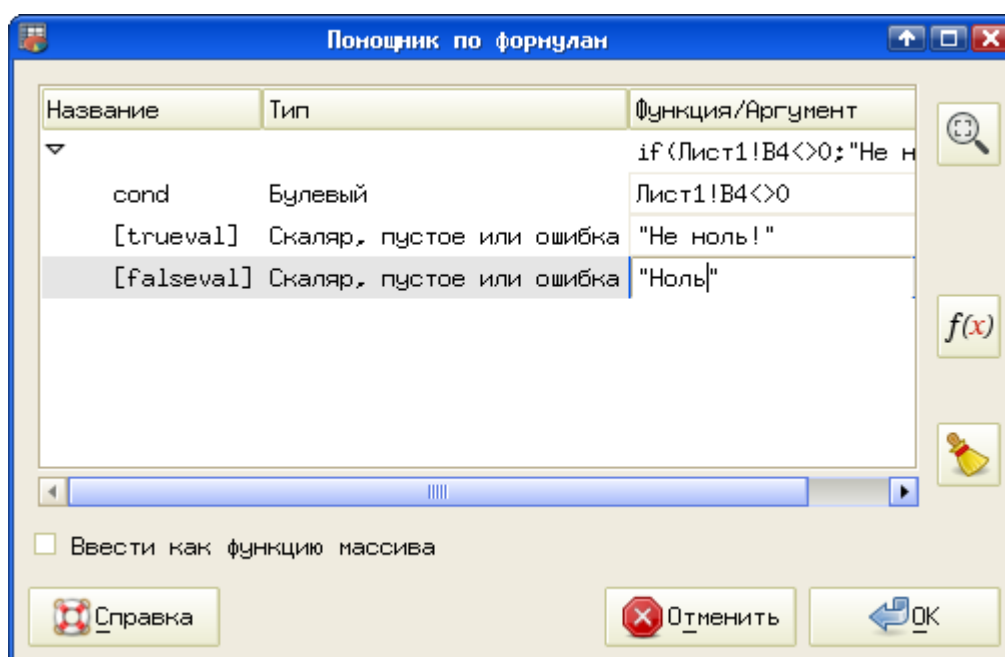


Рисунок 2.16. Составление формулы

Для определения аргументов нужно щелкнуть мышкой в поле ввода (в столбце «Функция/Аргумент», указать ячейку со значением аргумента (можно щелкнуть мышкой по нужной ячейке) и при необходимости дописать остальное. Завершается определение аргумента нажатием на <ENTER>, после чего с помощью щелчка мышкой переходим к определению следующего аргумента. По мере определения аргументов функция дописывается. Окончательный результат построения функции `if()` показан на рис. 2.17.

Рисунок 2.17. Функция `if()` с аргументами

Кнопка  $f(x)$  в этом диалоге позволяет вставить функцию в качестве аргумента формируемой функции. При необходимости можно редактировать ссылки типа «Лист1!B4», превращая их в ссылки типа «B4».

После окончательного определения всех аргументов и нажатия на кнопку «ОК» можно наблюдать результат работы функции. При необходимости можно редактировать функцию либо в строке ввода по нажатию клавиши <F2>, либо снова вызвать диалог определения аргументов функции, используя кнопку  $f(x)$ , когда ячейка с результатом функции является активной.

Далее кратко рассмотрим несколько основных групп функций, а потом перейдем к конкретным примерам.

### ▪ **Математические функции.**

Gnumeric содержит около 70 встроенных математических функций. Описывать все нет необходимости (использование и особенности функций типа  $\text{abs}()$ ,  $\text{sin}()$ ,  $\text{tan}()$  или  $\text{pi}()$  достаточно очевидны). Приведем здесь краткое описание некоторых более редких математических функций.

Таблица 1. Некоторые неочевидные математические функции

Название, аргументы	Назначение
$\text{atan2}(b1, b2)$	Вычисляет арктангенс отношения $b2/b1$ без учета знаков аргументов.
$\text{beta}(a, b)$	Вычисляет бета-функцию (интеграл Эйлера I рода) для положительных целых аргументов.
$\text{betaIn}(a, b)$	Вычисляет натуральный логарифм бета-функции для положительных целых аргументов.
$\text{combin}(n, k)$	Вычисляет количество уникальных комбинаций из $n$ по $k$ .
$\text{expm1}(x)$	Вычисляет $\text{exp}(x)-1$ с высокой точностью.
$\text{factdouble}(m)$	Вычисляет двойной факториал положительного числа $m$ , т.е. $m!!$ . Если $m$ не целое, дробная часть обрезается.
$\text{hypot}(a, b, \dots)$	Вычисляет квадратный корень из суммы квадратов аргументов.
$\text{ln1p}(x)$	Вычисляет $\ln(1+x)$ с высокой точностью.

Название, аргументы	Назначение
<code>multinomial(x1, x2, ...)</code>	Вычисляет отношение факториала суммы аргументов к произведению их факториалов.
<code>quotient(a, b)</code>	Вычисляет целую часть результата деления a на b.
<code>roman(m, тип)</code>	Преобразует натуральное число в «римскую» форму записи. Необязательный аргумент «тип» определяет вариант нотации (классический или краткий).
<code>seriesum(X, n, m, коэффициенты)</code>	Вычисляет сумму степенного ряда вида $a \cdot X^n + b \cdot X^{n+m} + c \cdot X^{n+2m} + d \cdot X^{n+3m} \dots$ , т.е. X – основание степени, n – начальное значение показателя степени, m – инкремент показателя степени, a коэффициенты a, b, c... - коэффициенты при каждом члене ряда, записанные в блоке ячеек таблицы.
<code>sqrtpi(m)</code>	Вычисляет квадратный корень из $m \cdot \pi()$ ( <code>sqrt(m*pi())</code> ).
<code>sumsq(x1, x2, ...)</code>	Вычисляет сумму квадратов аргументов.
<code>sumx2my2(вектор1, вектор2)</code>	Вычисляет сумму разностей квадратов соответствующих элементов векторов. Вектор1 и вектор2 – блоки ячеек одинаковой длины.
<code>sumx2py2(вектор1, вектор2)</code>	Вычисляет сумму сумм квадратов соответствующих элементов векторов.
<code>sumxmy2(вектор1, вектор2)</code>	Вычисляет сумму квадратов разностей соответствующих элементов векторов.

### ▪ **Логические функции.**

Логических функций вообще немного, поэтому рассмотрим их все. Каждая логическая функция может быть аргументом другой, если это позволено синтаксисом.



Таблица 2. Логические функции в Gnumeric

Название, аргументы	Назначение
<code>false()</code>	Аргументов не имеет. Всегда выдает логическое значение «ЛОЖЬ» (FALSE).
<code>true()</code>	Аргументов не имеет. Всегда выдает логическое значение «ИСТИНА» (TRUE).
<code>and(условие1, условие2, ...)</code>	Имеет неограниченное количество аргументов (более одного). Выдает логическое значение «ИСТИНА» (TRUE), если выполняются все условия, приведенные в аргументах. Пример синтаксиса: <code>and(c1&gt;0,b4=3,d2&lt;b4)</code> , где <code>c1</code> , <code>b4</code> и <code>d2</code> – адреса ячеек.
<code>or(условие1, условие2, ...)</code>	Имеет неограниченное количество аргументов (более одного). Выдает логическое значение «ИСТИНА» (TRUE), если выполняются хотя бы одно из условий, приведенных в аргументах. Синтаксис аналогичен функции <code>and()</code> .
<code>not(условие)</code>	Аргументом является условие или результат работы логической функции (логическое TRUE или FALSE). Выдает логическое значение «ИСТИНА» (TRUE), если условие не выполняется или аргумент установлен в FALSE.
<code>xor(условие1, условие2, ...)</code>	Имеет неограниченное количество аргументов (более одного). Выдает логическое значение «ИСТИНА» (TRUE), если выполняется нечетное количество условий (исключающее ИЛИ).
<code>if(условие, действие1, действие2)</code>	Проверяет условие, и если оно выполняется, производится действие1 (возможна проверка еще какого-то условия, выполнение арифметических операций или вычисление по формуле с функциями, а также вывод текста). В противном случае выполняется действие2 (с теми же особенностями).

Важным свойством функции `if()` является возможность использования результата работы одной функции `if()` в качестве аргумента другой функции `if()`. Однако количество таких вложений ограничивается максимально допустимой

длиной текста в ячейке, а также здравым смыслом (формулы с количеством проверок условий более 7-8 крайне трудно анализировать в случае ошибок). Также в качестве условий в `if()` могут использоваться любые синтаксически допустимые комбинации логических функций.

### ■ **Функции комплексного переменного.**

Функций работы с комплексными числами в `Gnumeric` насчитывается более сорока, поэтому здесь рассмотрим только основные (и наиболее интересные с точки зрения автора).

Таблица 3. Функции работы с комплексными числами

Название, аргументы	Назначение
<code>complex(x1, x2, символ)</code>	Формирует комплексное число из двух вещественных. Третий необязательный аргумент (символ) позволяет изменить обозначение мнимой единицы. Если он не указан, будет сформировано комплексное число вида $x1+x2i$ .
<code>imabs(complex)</code>	Вычисляет модуль комплексного числа. Например, если в ячейке E3 записано комплексное число $5+3i$ (как результат функции <code>complex()</code> ), то <code>imabs(E3)</code> выдаст 5,83095.
<code>imargument(complex)</code>	Вычисляет аргумент комплексного числа (показатель степени при экспоненциальном представлении). Для примера $5+3i$ выдаст значение 0,54042.
<code>imreal(complex)</code>	Выдает вещественную часть комплексного числа.
<code>imaginary(complex)</code>	Выдает мнимую часть комплексного числа.
<code>imconjugate(complex)</code>	Вычисляет комплексно сопряженное число.
<code>imdiv(complex1, complex2)</code>	Вычисляет целую часть результата деления двух комплексных чисел.
<code>iminv(complex)</code>	Выполняет преобразование $1/z$ .
<code>impower(complex, power)</code>	Возводит комплексное число в степень, которая тоже может быть комплексным числом.

Название, аргументы	Назначение
<code>improduct(complex1, complex2, ...)</code>	Вычисляет произведение комплексных чисел (обычная операция умножения не работает!).

Далее рассмотрим использование логических и математических функций, а также функций комплексного переменного на классическом примере вычисления корней квадратного уравнения с произвольными коэффициентами.

Итак, заданы три коэффициента  $A$ ,  $B$  и  $C$  квадратного уравнения вида

$$A \cdot x^2 + B \cdot x + C = 0 \quad (1)$$

Требуется вычислить корни  $x_1$  и  $x_2$ , которые в общем случае могут быть комплексными. Перед вычислением корней вычислим дискриминант  $D$ :

$$D = B^2 - 4 \cdot A \cdot C \quad (2)$$

И затем воспользуемся формулой для вычисления корней:

$$x_{1,2} = \frac{-B \pm \sqrt{D}}{2 \cdot A} \quad (3)$$

Однако при отрицательном дискриминанте корни будут комплексными. Такое комплексное число будет иметь вещественную часть ( $-B/2A$ ) и мнимую часть (с точностью до знака)

$$\Im(x_{1,2}) = \pm \frac{\sqrt{|D|}}{2 \cdot A} \quad (4)$$

В то же время для неотрицательных значений дискриминанта будут работать обычные правила вычисления корней (в соответствии с формулой (3)). Таким образом, в формуле для вычисления корня должна присутствовать проверка дискриминанта на отрицательность, и при отрицательном дискриминанте должно быть сформировано комплексное число. При неотрицательном дискриминанте используются обычные функции и арифметические действия.

После столь долгих рассуждений пора показать таблицу и формулы для вычислений (рис. 2.18).

	А	В	С
1	<b>Решатель квадратных уравнений</b>		
2			
3	Коэффициенты: А		1
4		В	2
5		С	3
6	Дискриминант: D		-8
7	Корни:	X1	-1+2,82842712474619i
8		X2	-1-2,82842712474619i
9			

Рисунок 2.18. Решение квадратного уравнения

Формулы для вычислений приведены ниже:

Таблица 4. Формулы для решения квадратного уравнения

Адрес ячейки, назначение	Формула
C6: Дискриминант	=C4^2-4*C3*C5
C7: Корень X1	=if(C6<0;complex(-C4/(2*C3);sqrt(abs(C6)));(-C4+sqrt(C6))/(2*C3))
C8: Корень X2	=if(C6<0;complex(-C4/(2*C3),-sqrt(abs(C6)));(-C4-sqrt(C6))/(2*C3))

### ■ **Календарные функции.**

Календарные функции в Gnumeric находятся в категории «Дата/Время», всего таких функции более 20. Основными являются функции «разложения» даты на составляющие – выделения из даны номера дня в месяце (функция `day()`), номера месяца в году (`month()`) и года (`year()`) – и функция обратного преобразования `date(year(), month(), day())`, которая конструирует данные типа «дата» из номера года, месяца и дня.

Также часто используются функции автоматического определения дня недели (`weekday()`), текущей даты (`today()`) и текущего момента времени (`now()`).

Имеются также интересные (и полезные) функции преобразования дат – `date2unix()` и обратная ей `unix2date()`. Они выполняют преобразование даты в количество секунд, прошедших с начала «эры UNIX» и наоборот («эра UNIX» началась в полночь 1 января 1970 года).

Для знакомства с основными календарными функциями рассмотрим следующую задачу.

Дан список лиц, для которых известны фамилии, пол и даты рождения. Определить:

- День недели, на который приходится день рождения каждого человека в

текущем году. Если день рождения приходится на выходные дни, вывести текст «УРА!», в остальных случаях вывести текст «УВЫ...».

b. Возраст на настоящий момент.

c. Дату выхода на пенсию для каждого человека.

Для этой задачи воспользуемся исходными данными (списком фамилий) из задачи про доходы и налоги. Пол установим в соответствии с фамилиями, даты рождения введем произвольно (как уже упоминалось, удобно вводить дату в виде 23/11/88, а программа приведет ее в нужный вид).

Для получения решения по пункту "а" необходимо проделать некоторые промежуточные вычисления. Сначала нужно для каждого лица сформировать дату рождения в текущем году на основании дня и месяца рождения, а также номера текущего года. Тогда первая формула (в ячейке D4 на рис. 2.19) будет иметь вид:

```
=date(year(today()),month(C4),day(C4))
```

Очевидно, что функция today() просто выдает текущую дату, а функция date() формирует дату из номера года, номера месяца и номера дня в месяце, определяемых соответственно, с помощью функций year(), month() и day(). Естественно, при желании вместо результатов работы функций можно использовать адреса ячеек, содержащих числа или просто числа. Соответственно, формула для окончательного результата по пункту «а» (в ячейке E4) будет иметь вид:

```
=if(or(weekday(D4,2)=6,weekday(D4,2)=7),"УРА!","УВЫ..."), где
```

функция weekday() вычисляет день недели для даты, указанной в первом аргументе. Второй аргумент функции weekday() определяет правила нумерации дней недели. В данном случае первый день недели соответствует понедельнику.

Для обработки всего списка просто копируем эти формулы вниз.

Для получения решения по пункту «b» воспользуемся функцией вычисления разности дат – datedif(). Эта функция имеет три аргумента – начальную дату, конечную дату и строку-параметр, задающую единицы измерения разности. Значения параметра «y», «m» или «d» позволяют найти разность дат соответственно, в полных годах, полных месяцах и в днях. С учетом того, что возраст определяется в полных годах, запишем формулу для возраста в следующем виде:

```
=datedif(C4,today(),'y')
```

Для получения решения по пункту «с» снова нужно формировать даты, используя значения параметров возраста выхода на пенсию. Эти возрасты на момент написания пособия составляют 60 лет для мужчин и 55 для женщин, однако они могут в любой момент измениться, поэтому конкретные числа в формулу записывать не будем. Дата выхода на пенсию формируется с использованием условия проверки пола. Итак, получаем формулу:

```
=if(B4="М",date(year(C4)+$G$1,month(C4),day(C4)),date(year(C4)+$G$2,month(C4),day(C4)))
```

Результирующая таблица показана на рис. 2.19.

А	В	С	Д	Е	Ф	Г
	Сегодня:	03/04/10			Мужчины	60
					Женщины	55
Фамилия, инициалы	Пол	Дата рожд.	Д/р в этом году		Возраст	На пенсию
Сионов К.С.	М	11/03/56	11/03/10	УВЫ...	54	11/03/2016
Чернышевская С.П.	Ж	03/11/65	03/11/10	УВЫ...	44	03/11/2020
Чкалов В.П.	М	22/08/99	22/08/10	УРА!	10	22/08/2059
Романов П.А.	М	08/01/88	08/01/10	УВЫ...	22	08/01/2048
Штек Е.А.	Ж	06/06/66	06/06/10	УРА!	43	06/06/2021
Зубарева О.А.	Ж	05/05/55	05/05/10	УВЫ...	54	05/05/2010
Жеребцова Н.Н.	Ж	29/09/77	29/09/10	УВЫ...	32	29/09/2032
Зубов П.Р.	М	16/01/68	16/01/10	УРА!	42	16/01/2028

Рисунок 2.19. Пример вычислений с датами

Последний столбец (дата выхода на пенсию) приведен в формате «dd/mm/yyyy» для иллюстрации корректного выполнения вычислений.

### ■ **Функции поиска соответствий.**

В тех случаях, когда использование функции `if()` становится неудобным по причине большого количества вложений или большой длины формулы, а также для повышения эффективности вычислений, в электронных таблицах используются функции поиска соответствий. К таким функциям относятся `lookup()`, `vlookup()`, `hlookup()`, `match()` и `index()`, находящиеся в категории «Поиск».

Понимание работы этих функций может сначала вызывать некоторые затруднения, однако они (вместе с `if()`) обеспечивают мощные средства обработки данных, поэтому их освоение стоит затрат времени и интеллекта.

Функции `lookup()`, `vlookup()` и `hlookup()` в качестве одного из аргументов используют так называемые «ассоциативные массивы» (справочники). Ассоциативный массив – это структура данных, оформленная в виде таблицы, первый столбик которой содержит «ключи» – данные, которые участвуют в формировании условий. В следующих столбиках ассоциативного массива содержатся значения, соответствующие ключам. Таким образом, по значению ключа можно однозначно получить какие-то другие данные (в языках программирования такие структуры называются «хэш-таблицы»).

В программах ЭТ ассоциативные массивы реализуются как блоки ячеек (справочные таблицы), содержащие минимум два столбца. Первый столбец содержит ключи, второй – значения, соответствующие ключам.

Для примера рассмотрим следующую задачу.

В ралли участвуют гонщики на автомобилях, для которых известны марки и расход топлива в литрах на 100 км. Дан список, в котором указаны фамилии гонщиков и марки автомобилей. Известна протяженность трассы гонок в километрах  $L$ . Определить расход топлива для каждого участника гонок.

Для решения задачи составим справочную таблицу для 7-ми разных марок а/м, и расположим ее в ячейках F3:G10 (рис. 2.20). Полезно соблюдать алфавитный порядок текстовых значений в первом столбике справочной таблицы.

E	F	G	
	Марка а/м	Расход, л/100км	
	BMW	13	
	FIAT	10,5	
	Ford	15	
	Honda	11,4	
	Nissan	14,3	
	Toyota	12,5	
	Volvo	13,3	

Рисунок 2.20. Справочник для задачи о ралли

Данные (список участников и марки их а/м) запишем в ячейки A3:B10 (рис. 2.21). Длину трассы L, которая является параметром, запишем в ячейку C1. Для вычисления полного расхода топлива используем формулу с функцией lookip( ).

A	B	C	D
	Длина трассы L	500	
	Фамилия	Марка а/м	Расход на трассу
	Шумахер	Toyota	
	Бельмондо	FIAT	
	Валигора	Nissan	
	Пороханов	Volvo	
	Коннери	BMW	
	Сигал	Ford	
	Кэмпбелл	Honda	

Рисунок 2.21. Исходные данные задачи о ралли

Формула в ячейке C4 будет выглядеть следующим образом.

=lookup(B4, \$F\$4:\$F\$10, \$G\$4:\$G\$10) \* \$C\$1/100

Функция lookup() считывает содержание ячейки, указанной в первом аргументе, ищет это значение в диапазоне ячеек (столбце), указанном во втором аргументе и выдаёт соответствие этому значению из диапазона ячеек (столбца),

указанном в третьем аргументе. Таким образом, для получения результата по названию а/м находим расход топлива на 100 км и умножаем это значение на количество сотен километров. Принципиально важно указывать абсолютные адреса блоков ячеек справочной таблицы.

Итоговая таблица показана на рис. 2.22.

А	В	С	Д
	Длина трассы L		500
Фамилия	Марка а/м	Расход на трассу	
Шумахер	Toyota	62,5	
Бельмондо	FIAT	52,5	
Валигора	Nissan	71,5	
Пороханов	Volvo	66,5	
Коннери	BMW	65	
Сигал	Ford	75	
Кэмпбелл	Honda	57	

Рисунок 2.22. Решение задача о ралли

Ограничение функции `lookup()` – только один столбец соответствий. Более «мощной» является функция `vlookup()`, в которой второй аргумент определяет весь блок ячеек, содержащих ассоциативный массив (справочник), а третий аргумент указывает, в каком столбце ассоциативного массива нужно искать соответствие ключу. Четвёртый (необязательный) аргумент определяет порядок сортировки первого («ключевого») столбца справочника. Если он не указан или равен 1 (логическая ИСТИНА) то первый столбец ассоциативного массива для функции `vlookup()` должен содержать числа, отсортированные по возрастанию, или текст, отсортированный в алфавитном порядке. Если значения в первом столбце не отсортированы, то четвертый аргумент должен быть установлен в 0. Еще одним большим достоинством функции `vlookup()` является возможность работы с диапазонами значений ключа.

Для примера рассмотрим вычисление суммы годового налога при прогрессивной налоговой шкале. Пусть при годовом доходе до 10000 у.е. ставка налога составляет 12%, до 30000 у.е. – 20%, до 50000 у.е. – 25% и при большем доходе – 35%. Для создания таблицы данных используем фамилии из предыдущего примера, а суммы годового дохода запишем такие, чтобы можно было реализовать все варианты ставок налога (рис. 2.23). Таблицу данных разместим в диапазоне A3:B10.



А	В	С
Фамилия	Годовой доход	Сумма налога
Шумахер	78000	
Бельмондо	46000	
Валигора	29000	
Пороханов	30000	
Коннери	10000	
Сигал	18400	
Кэмпбелл	5670	

Рисунок 2.23. Исходные данные задачи о налогах

Справочную таблицу размещаем в диапазоне E3:F7 (рис. 2.24), формула в ячейке С4 с использованием функции `vlookup()` выглядит следующим образом:

`=vlookup(B4, $E$4:$F$7, 2) * B4`

Е	Ф
Сумма дохода	Ставка налога
0	0,12
10000	0,2
30000	0,25
50000	0,35

Рисунок 2.24. Справочник к задаче о налогах

Во втором столбце ассоциативного массива находим ставку налога, соответствующую доходу, а затем получаем сумму налога, умножая ставку налога на величину дохода (рис 2.25).

А	В	С
Фамилия	Годовой доход	Сумма налога
Шумахер	78000	27300,00
Бельмондо	46000	11500,00
Валигора	29000	5800,00
Пороханов	30000	7500,00
Коннери	10000	2000,00
Сигал	18400	3680,00
Кэмпбелл	5670	680,40

Рисунок 2.25. Решение задачи о налогах

Важно, что `vlookup()` выбирает значения, соответствующие нижней границе диапазона значений ключей, и используется минимальное значение ключа в интервале.

Четвертый аргумент функции `vlookup()` также влияет и на возможность интервального просмотра. Если этот аргумент имеет значение ИСТИНА (1) или опущен, то интервальный просмотр работает, как описано выше. Если этот аргумент имеет значение ЛОЖЬ (0), то функция `vlookup()` ищет точное соответствие. Если такое не найдено, то возвращается значение ошибки `#N/A` (`#N/D` – «нет данных»). Таким образом, для использования возможности работы с интервалами значений в `vlookup()` первый столбец справочной таблицы (ассоциативного массива) обязательно должен быть отсортирован по возрастанию.

Функция `hlookup()` работает аналогично `vlookup()`, только порядок следования «ключей» – не сверху вниз, а слева направо.

Теперь рассмотрим формат функций `match()` и `index()`.

`match(искомое_значение, искомый_массив, тип_сопоставления)` — находит позицию (порядковый номер) искомого значения в одномерном массиве. Значение аргумента «тип сопоставления» – (-1, 0 или 1) – зависит от того, упорядочен ли массив (-1 — массив упорядочен по убыванию, находится место наименьшего значения, которое больше или равно искомому, 0 — массив может быть неупорядоченным, находится место первого значения, равного исходному, 1 — массив упорядочен по возрастанию, находится место наибольшего значения, которое меньше или равно искомому).

`index(массив, номер_строки, номер_столбца)` — находит значение элемента, находящегося в заданном массиве на пересечении заданных строки и столбца.

Далее рассмотрим пример.

Пусть на основе эксперимента получена следующая зависимость посещаемости дискотеки от входной платы:

Входная плата, у.е.	1	1,5	2	2,5	3	3,5	5
Количество посетителей	200	175	160	140	124	110	70

Определить оптимальную входную плату.

Очевидно, что оптимуму соответствует максимальная выручка. Соответственно. решение задачи состоит в том, чтобы подсчитать выручку в каждом случае, найти максимальную и написать формулу, показывающую входную плату, соответствующую максимальной выручке. То есть нужно определить номер столбика, в котором получается максимальная выручка и вывести значение входной платы.

Определяем выручку в каждом случае (умножив входную плату на количество посетителей), затем функцией `max()` находим наибольшую выручку. После этого функцией `match()` определяем, на каком месте в массиве она находится и функцией `index()` смотрим, какая входная плата находится на этом месте (рис. 2.26). Рядом с ячейками с результатами приведены формулы для получения этих результатов (используется функция `expression()`).

А	В	С	Д	Е	Ф	Г	Н
Входная плата, у.е.	1	1,5	2	2,5	3	3,5	5
Кол-во посетителей	200	175	160	140	124	110	70
Выручка	200	262,5	320	350	372	385	350
Макс. выручка	385	<code>max(B4:H4)</code>					
Позиция	6	<code>match(B6:B4:H4)</code>					
Оптимальная плата	3,5	<code>index(B2:H4;1;B7)</code>					

Рисунок 2.26. Решение задачи о дискотеке

В этом примере участвует функция `max()`, которая относится к категории статистических функций, к знакомству с которыми теперь и перейдем.

### ■ **Статистические функции.**

Основные статистические функции – это `min()`, `max()` и `average()`, вычисляющие, соответственно, минимальное, максимальное и среднее значение для набора аргументов. Аргументы могут быть либо перечислены через запятую (если данные находятся в несмежных ячейках), либо в качестве аргумента может быть использован диапазон ячеек. Также к статистическим функциям можно отнести функции `sumif()` и `countif()`, которые формально находятся в `Gnumeric` среди математических функций, а также функцию `rank()`, определяющее «рейтинг» какого-то значения в списке аналогичных значений.

В качестве примера рассмотрим следующую задачу.

Дан список участников соревнования среди студентов по бегу на 100 метров и метанию мяча. В таблице (рис. 2.27) указаны пол (юноша или девушка) и результаты. Определить места каждому участнику в каждом виде соревнований,

минимальный, максимальный и средний результаты в каждом виде соревнований, на сколько юноши (в среднем) метают мяч дальше, чем девушки.

Сразу под списком в соответствующих столбцах подсчитываем максимальный, минимальный и средний результаты (по столбцу С =max(C2:C16), =min(C2:C16), =average(C2:C16) и аналогично по столбцу D).

Затем подсчитываем количество юношей и количество девушек. В ячейку B22 вносим формулу

```
=countif(B2:B16, "юноша"),
```

а в ячейку B23 формулу

```
=countif(B2:B16, "девушка")
```

В ячейки C22 и C23 записываем формулы для подсчета суммы результатов по метанию для юношей и девушек соответственно

```
=sumif(B2:B16, "юноша", D2:D16)
```

```
=sumif(B2:B16, "девушка", D2:D16)
```

После чего подсчитываем среднее значение в ячейках D22 и D23, разделив сумму результатов на количество участников в каждой группе. Затем подсчитываем разницу средних значений (рис. 2.27).

А	В	С	Д
Фамилия	Пол	БЕГ	МЕТАНИЕ
Беляева	девушка	17,6	35
Васильева	девушка	18,2	55
Виноградов	юноша	14,3	38
Воробьева	девушка	16,8	46
Герасимов	юноша	15,2	55
Дмитриев	юноша	12,3	57
Егоров	юноша	14,5	56
Иванова	девушка	15,9	47
Карпов	юноша	15,3	50
Крылова	девушка	15,9	38
Матвеева	девушка	18,4	42
Новикова	девушка	16,3	53
Осипова	девушка	17,5	51
Павлова	девушка	16,0	54
Петров	юноша	13,4	57
наибольшее значение		18,4	57
наименьшее значение		12,3	35
среднее значение		15,8	48,93
	количество	сумма метров	среднее значение
юноши	6	313	52,17
девушки	9	421	46,78
	разница		5,39

Рисунок 2.27. Решение задачи о соревнованиях

Для иллюстрации использования функции `rank()` рассмотрим следующую задачу: На основе данных из задачи о соревнованиях распределить места в соревнованиях по бегу среди юношей, а также определить, сколько человек показали результаты ниже среднего (т.е. время больше среднего).

Для распределения участников по местам как раз и потребуется функция `rank()`, а кроме того потребуется `countif()` для вычисления с условием. Однако условие для `countif()` обязательно должно быть текстом, поэтому при формировании условия с вычисляемыми данными целесообразно использовать текстовую функцию `concatenate()`. Результат использования этих функций показан на рис. 2.28.

А	В	С
Фамилия	БЕГ	Места
Виноградов	14,3	3
Герасимов	15,2	5
Дмитриев	12,3	1
Егоров	14,5	4
Карпов	15,3	6
Петров	13,4	2
Среднее время	14,1666666666667	
Условие	>14,1666666666667	
Хуже среднего	4	

Рисунок 2.28. Задача о распределении мест

Формула для определения места участника соревнований (ячейка С2) будет выглядеть следующим образом:

`=rank(B2, $B$2:$B$7, 1)`

Абсолютные адреса диапазона использованы для обеспечения возможности копирования формулы. Третий аргумент установлен в «1», что обеспечивает обратный порядок распределения мест, т.е. чем меньше значение, тем лучше место (первое место – минимальный результат). Для обеспечения прямого порядка распределения мест (первое место – максимальный результат) третий аргумент нужно установить в «0» или не указывать.

Среднее время определяется с помощью функции `average()`, а условие для подсчета аутсайдеров формируется с помощью текстовой функции `concatenate()`, которая «сцепляет» строки для формирования одного значения строкового типа. В данном случае в ячейке В10 записана формула

`=concatenate(' '>' ', B9).`

Подсчет аутсайдеров выполняется по формуле (ячейка В11)

`=countif(B2:B7, B10).`

К статистическим функциям относятся также функции `count()` (подсчет количества числовых значений в диапазоне ячеек) и `counta()` (подсчет количества не пустых ячеек в диапазоне). Кроме того, большое число статистических функций вычисляют статистические параметры различных распределений и обеспечивают генерацию случайных чисел с заданными параметрами распределений.

### ◦ **Текстовые (строковые) функции**

Одна из типовых задач в офисной работе – изменения формы представления списков людей или организаций. Избежать трудоёмкой работы по переписыванию текста из одного вида в другой помогают функции работы с текстом (строковые функции).

Почему-то подобные задачи кажутся очень сложными, поэтому покажем пример решения и рассмотрим минимальный набор функций, необходимых для получения результата.

Пусть имеется список лиц с полными фамилиями, именами и отчествами (например, список группы слушателей каких-нибудь курсов. Для составления журнала полные имена и отчества не требуются, а требуются только инициалы. Задача заключается в преобразовании имён и отчеств в буквы инициалов.

На рис. 2.29 показан пример решения такой задачи. Следует обратить внимание, что имеются скрытые столбцы, в которых содержатся результаты промежуточных вычислений.

	А	В	Н	Ж	Л	
1						
2	Фамилия, имя, отчество	Фамилия	Имя	Отчество	Результат	
3	Виноградов Василий Семёнович	Виноградов	Василий	Семёнович	Виноградов В.С.	
4	Герасимов Иван Сергеевич	Герасимов	Иван	Сергеевич	Герасимов И.С.	
5	Дмитриев Сергей Никитович	Дмитриев	Сергей	Никитович	Дмитриев С.Н.	
6	Егоров Василий Иванович	Егоров	Василий	Иванович	Егоров В.И.	
7	Карпов Владимир Яковлевич	Карпов	Владимир	Яковлевич	Карпов В.Я.	
8	Петров Борис Андреевич	Петров	Борис	Андреевич	Петров Б.А.	

Рисунок 2.29. Пример преобразования списка

Алгоритм решения задачи может быть таким:

1. Определяем длину строки «Фамилия, имя, отчество»
2. Определяем длину фамилии (количество букв до первого пробела)
3. Делим строку «Фамилия, имя, отчество» на фамилию и всё что осталось (получаются строки «Фамилия» и «Имя, отчество»)
4. Определяем длину строки «Имя, отчество»
5. Определяем длину имени (количество букв до первого пробела в строке «Имя, отчество»)
6. Делим строку «Имя, отчество» на имя и всё что осталось (получаются строки «Имя» и «Отчество»)
7. Выделяем первую букву имени
8. Выделяем первую букву отчества
9. Создаём итоговую строку из строки «Фамилия» и первых букв имени и отчества

Строковые функции, которые понадобятся для решения данной задачи, описаны в таблице ниже.

Таблица 5. Некоторые строковые функции

Название, аргументы	Назначение
<code>len(str)</code>	Вычисляет длину (количество символов) для строки <code>str</code> .
<code>find(str1;str2;start)</code>	Определяет позицию (номер символа), с которой начинается подстрока <code>str1</code> в строке <code>str2</code> , начиная с позиции <code>start</code> . Если аргумент <code>start</code> не указан, поиск идёт с начала строки.
<code>left(str;n)</code>	Выделяет <code>n</code> символов с начала строки <code>str</code> . Если аргумент <code>n</code> не указан, функция возвращает первый символ строки.
<code>right(str;n)</code>	Выделяет <code>n</code> символов с конца строки <code>str</code> . Если аргумент <code>n</code> не указан, функция возвращает последний символ строки.
<code>concatenate(str1;str2;...;strN)</code>	Формирует одну строку из «фрагментов» – строк <code>str1</code> , <code>str2</code> , ..., <code>strN</code> .

В следующей таблице приведены формулы, использованные при решении задачи.



Таблица 6. Формулы для решения задачи со списком (для первой строки данных)

Адрес ячейки, назначение	Формула
В3: Длина исходной строки	=len(A3)
С3: Позиция первого пробела (количество букв в фамилии)	=find(" ";A3)
D3: Строка «Фамилия»	=left(A3;C3)
E3: Строка «Имя, отчество»	=right(A3;B3-C3)
F3: Длина строки«Имя, отчество»	=len(E3)
G3: Позиция первого пробела в строке «Имя, отчество» (количество букв в имени)	=find(" ";E3)
H3: Строка «Имя»	=left(E3;G3)
I3: Первая буква имени	=left(H3)
J3: Строка «Отчество»	=right(E3;F3-G3)
K3: Первая буква отчества	=left(J3)
L3: Результат	=concatenate(D3;" ";I3;".";K3;".")

Символьные (строковые) значения в формулах (в аргументах функций) следует указывать в кавычках.

В качестве других полезных строковых функций (по мнению автора) нужно отметить функции преобразования регистров `lower()` (переводит все символы строки в нижний регистр, т. е. в строчные буквы) и `upper()` (переводит все символы строки в верхний регистр, т. е. в прописные буквы), функцию `mid()`, которая позволяет вывести заданное количество символов строки, начиная с заданного символа, а также функцию `value()`, превращающую строку из символов-цифр в число.

### ◦ **Обработка матриц**

Гnumeric обеспечивает базовые возможности по работе с матрицами. Список функций приведён в таблице ниже. Все эти функции в качестве аргументов используют диапазоны ячеек, в которые записаны элементы матриц.

Таблица 7. Функции для обработки матриц

Название, аргументы	Назначение
transpose(matrix)	Выполняет транспонирование матрицы matrix (строки становятся столбцами и наоборот). Функция может находиться в категории «Поиск».
mdeterm(matrix)	Вычисляет определитель квадратной матрицы.
minverse(matrix)	Вычисляет матрицу, обратную по отношению к исходной, при условии, что определитель не равен нулю.
mmult(matrix1;matrix2)	Вычисляет матричное произведение. Результирующая матрица имеет строк как в matrix1 и столбцов как в matrix2.

Чтобы в результате операций с матрицами получить тоже матрицу (если это нужно), в «Помощнике по формулам» нужно установить режим «Ввести как функцию массива» (рис. 2.30).

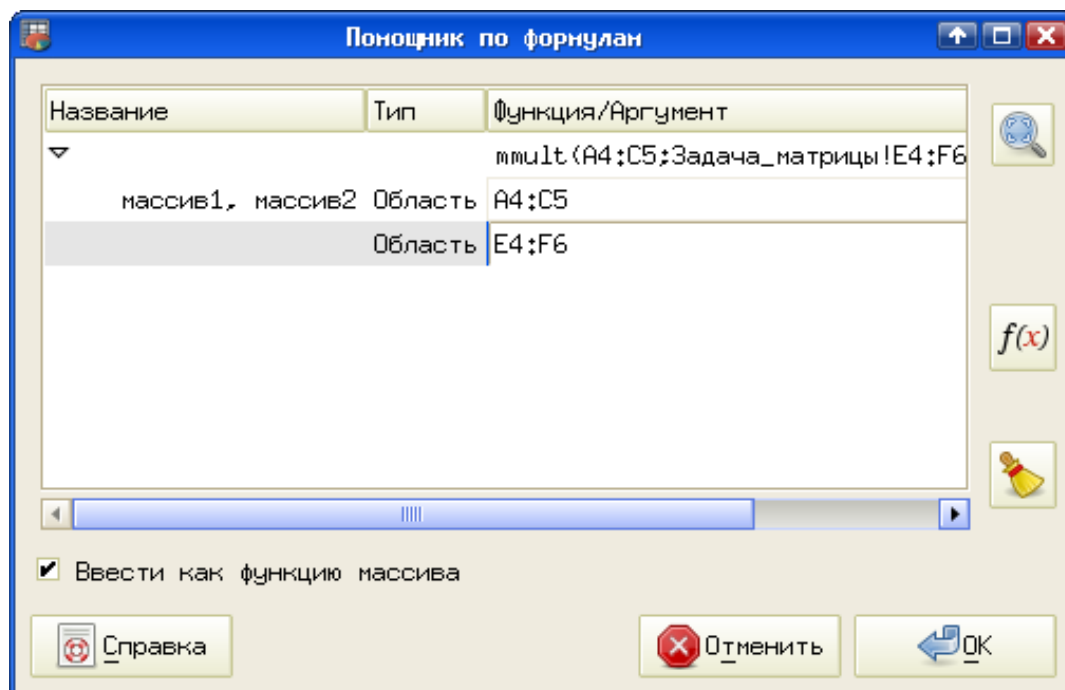


Рисунок 2.30. Создание формулы в режиме работы с матрицами

Пример вычислений с матрицами показан на рис. 2.31.

	A	B	C	D	E	F	G	H	I	J	
1											
2	Матрица1				Матрица2				Матрица3		
3											
4		1	2	3		7	8		3	4	5
5		4	5	6		9	1		7	8	9
6						2	3				
7											
8	Трансп. M1				M1xM2				M2xM3		
9		1	4			31	19		77	92	107
10		2	5			85	55		34	44	54
11		3	6						27	32	37
12											
13	Матрица4				Обр. M4				Единичная		
14		3	5	6		-2,5	2	1	1	-0	-0
15		1	3	4		9,5	-8	-3	0	1	0
16		7	8	9		-6,5	6	2	0	-0	1
17											
18	Det. M4	2									

Рисунок 2.31. Пример матриц и результатов операций с ними

В Gnumeric отсутствует функция для создания единичной матрицы, но такая матрица легко может быть получена умножением произвольной матрицы на обратную ей, что и проиллюстрировано на рис. 2.31 (здесь -0 указывает на то, что в результате погрешностей вычислений в связи с ограниченной точностью представления получилось вещественное отрицательное число с очень маленьким модулем).