

П.П. Мельников

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ

Рекомендовано УМО по образованию
в области финансов, учета и мировой экономики
в качестве **учебного пособия** для студентов,
обучающихся по специальностям «Финансы и кредит»,
«Бухгалтерский учет, анализ и аудит»,
«Налоги и налогообложение»,
«Мировая экономика»



МОСКВА
2012

УДК 004.9:33(075.8)

ББК 65с51я73

М48

Рецензенты:

С.В. Назаров, заведующий кафедрой информатики и прикладного программного обеспечения Национального исследовательского университета – Высшей школы экономики, д-р техн. наук, проф.,

В.А. Титов, доц. кафедры прикладной математики и естественно-научных дисциплин Московского института предпринимательства и права, канд. техн. наук

Мельников П.П.

М48 Компьютерные технологии в экономике : учебное пособие / П.П. Мельников. – М.: КНОРУС, 2012. – 224 с.

ISBN 978-5-406-02024-1

Рассматриваются компьютерные технологии вычислений в среде табличного процессора MS Excel, применяемые для решения систем эконометрических и нелинейных уравнений, исследования функций, решения задач условной оптимизации, дифференциального и интегрального исчисления, статистического анализа, анализа табличных данных, финансовых расчетов. Также рассматриваются технологии автоматизации при решении трудоемких и типовых задач на основе применения MS Visual Basic for Application.

Для студентов экономических вузов, обучающихся по специальностям «Финансы и кредит», «Бухгалтерский учет, анализ и аудит», «Мировая экономика», а также для студентов магистратуры, обучающихся по направлению «Экономика» и изучающих дисциплину федерального компонента ГОС ВПО «Компьютерные технологии в экономической науке и практике».

УДК 004.9:33(075.8)

ББК 65с51я73

Мельников Петр Петрович

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ

Сертификат соответствия

№ РОСС RU. АЕ51. Н 15407 от 31.05.2011 г.

Изд. № 4484. Формат 60×90/16.

Гарнитура «Petersburg». Печать офсетная. Бумага газетная.
Усл. печ. л. 14,0. Уч.-изд. л. 7,0. Тираж 500 экз. Заказ № 6068.

ООО «КноРус».

129085, Москва, проспект Мира, д. 105, стр. 1.

Тел.: (495) 741-46-28.

E-mail: office@knorus.ru <http://www.knorus.ru>

Отпечатано в полном соответствии с качеством предоставленного издательством электронного оригинал-макета в ГУП «Брянское областное полиграфическое объединение». 241019, г. Брянск, пр-т Ст. Димитрова, 40.

ISBN 978-5-406-02024-1

© Мельников П.П., 2012

© ООО «КноРус», 2012

ОГЛАВЛЕНИЕ

Введение	5
Глава 1. Технологии численного решения задач анализа и планирования	
1.1. Технологии решения систем эконометрических уравнений	8
1.2. Моделирование последовательностей и рядов	25
1.3. Моделирование и исследование функций	31
1.4. Технологии решения систем нелинейных уравнений	48
1.5. Применение технологии исследования функций для решения экономических задач	53
1.6. Технология численного вычисления определенного интеграла	60
1.7. Технологии решения задач оптимизации	63
Контрольные вопросы	69
Задачи для самостоятельного решения	70
Глава 2. Модели и технологии статистического анализа	
2.1. Генерация случайных величин	79
2.2. Вычисление числовых характеристик параметров случайных величин ...	83
2.3. Вычисление вероятности отдельных значений случайных величин	84
2.4. Технологии решения задач статистического анализа	92
2.5. Технологии вычисления основных статистических характеристик	97
2.6. Вычисление доверительного интервала для среднего значения	101
2.7. Технология проверки соответствия данных, полученных экспериментально, теоретическому распределению	103
2.8. Решение задач статистического анализа	106
Контрольные вопросы	117
Задачи для самостоятельного решения	117
Глава 3. Технологии финансовых вычислений	
3.1. Финансовые расчеты по простым процентам	120
3.2. Финансовые расчеты по сложным процентам	126
Контрольные вопросы	130
Задачи для самостоятельного решения	130
Глава 4. Технологии анализа данных, представленных таблично	
4.1. Организация модели данных в виде списков	132

4.2.	Применение формы при работе со списками	132
4.3.	Анализ данных на основе их сортировки	135
4.4.	Фильтрация списков	140
4.5.	Применение функций баз данных для анализа данных	152
4.6.	Анализ данных с применением технологии консолидации	155
4.7.	Анализ данных на основе сводных таблиц	157
	Контрольные вопросы	164
	Задача для самостоятельного решения	165
 Глава 5. Автоматизация решения трудоемких и типовых задач		
5.1.	Создание макросов	166
5.2.	Компоненты среды Visual Basic for Application	170
5.3.	Язык программирования Visual Basic	174
5.4.	Редактирование макросов	199
5.5.	Создание функций рабочего листа	200
5.6.	Создание приложений	208
	Контрольные вопросы	217
	Задачи для самостоятельного решения	217
Ответы к заданиям		219
Глоссарий		223
Использованная литература		224

ВВЕДЕНИЕ

Трудно представить область экономической деятельности, где бы не применялись компьютерные технологии анализа процессов и технологии их моделирования.

Компьютерное моделирование широко проникло практически во все сферы исследований. Моделирование позволяет оценить параметры процессов изучаемого объекта, не прибегая к натурному эксперименту, и сделать соответствующие выводы об изменении этих параметров с целью улучшения процессов функционирования. Особую роль компьютерное моделирование играет в области исследования экономических процессов. Эффективная экономическая деятельность, включающая множество различных по содержанию и сложности процессов, невозможна без их предварительного анализа и последующего планирования.

В отличие от многих процессов другого вида исследовать экономические процессы на натурном эксперименте по известным причинам не представляется возможным. Поэтому наиболее целесообразным и эффективным способом исследования практически любого экономического процесса является компьютерное моделирование.

Процесс моделирования в общем случае включает следующие этапы:

1) постановка проблемы. На этом этапе важно четко сформулировать суть проблемы, определить допущения и те вопросы, на которые нужно получить ответы. На этом этапе должны быть выделены основные черты моделируемого экономического процесса, определены основные зависимости, сформулированы гипотезы, объясняющие характер процесса;

2) разработка математической модели. Этап представляет собой математическую формализацию проблемы одним из доступных способов. Следует стремиться к тому, чтобы модель принадлежала к классу хорошо изученных задач, методы решения которых известны;

3) подготовка исходных данных;

4) формализация математической модели в терминах программного продукта, применяемого для разработки компьютерной модели. При формализации используются различные методы и инструменты обработки данных;

- 5) планирование эксперимента;
- 6) численное решение;
- 7) анализ полученных результатов и подготовка решения.

Наиболее ответственным моментом компьютерного моделирования является подготовка модели экономического объекта, с помощью которой будет проводиться исследование объекта или процесса. Особенностью экономических объектов является высокая степень сложности, поэтому разработка их моделей трудоемка и требует затрат значительных ресурсов. При разработке моделей в них не отражают те процессы, которые не являются существенными для эксперимента.

Под термином «модель» понимают прототип реального объекта либо процесса, который адекватно отражает только те свойства реального процесса или объекта, которые существенны для исследования или решения задачи. Таким образом, изучение одних сторон моделируемого объекта осуществляется ценой отказа от отражения в модели других его сторон.

Иными словами, модель — это такой объект, который в процессе исследования замещает объект-оригинал так, что его изучение дает новые знания об объекте-оригинале.

Модель любого экономического процесса представляет собой систему некоторых функций, имеющих один или несколько параметров. Задачи исследования сводятся к исследованию этих функций и к изучению влияния различных параметров этих функций на эффективность экономического процесса.

Основная задача моделирования экономических процессов с целью их исследования, прогнозирования и поиска наилучших решений сводится к расчету на модели анализируемых показателей при тех или иных значениях входных величин (параметров). Следует помнить, что достоверность результатов, получаемых при исследовании экономических процессов на моделях, в значительной мере зависит от адекватности применяемой модели реальному процессу.

Для экономического анализа и планирования находят широкое применение различные математические методы и основанные на них технологии компьютерного моделирования.

Класс экономических задач очень разнообразен, поэтому для построения моделей широко применяются различные вычислительные методы обработки данных: интерполяция, аппроксимация, решение уравнений и их систем, решение нелинейных уравнений, вычисление производных и интегралов, решение дифференциальных уравнений, методы оптимизации, вероятностные и статистические методы и др.

Исходные данные для компьютерных моделей экономических процессов в зависимости от их вида могут представляться аналитически, таблично, графически, в виде матриц и т.п.

Способ построения компьютерной модели определяется используемым для этой цели программным обеспечением. Наиболее распространенным и доступным средством является табличный процессор, входящий в пакет офисных приложений широко применяемых в настоящее время операционных систем ОС Microsoft Windows и открытой ОС Linux.

Табличный процессор представляет собой визуальную среду, позволяющую выполнять вычисления различного характера и не требующую знаний в области алгоритмизации и программирования. Большой набор библиотечных функций для рабочего листа позволяет автоматизировать процесс подготовки необходимой модели. Решая ту или иную задачу в табличном процессоре, пользователь использует различные модели, которые реализуются в разнообразных формах организации данных и способов их обработки, обеспечивают получение достоверных результатов. Возможности табличного процессора существенно расширяются за счет включенных в него надстроек, содержащих различные пакеты программ для статистического анализа, оптимизации.

В главах пособия рассматриваются технологии решения систем эконометрических и нелинейных уравнений, технологии исследования функций, технологии решения задач условной оптимизации, технологии дифференциального и интегрального исчисления, технологии статистического анализа, методы анализа табличных данных, технологии финансовых расчетов.

В заключительной главе рассматриваются технологии автоматизации на основе применения MS Visual Basic for Application.

ТЕХНОЛОГИИ ЧИСЛЕННОГО РЕШЕНИЯ ЗАДАЧ АНАЛИЗА И ПЛАНИРОВАНИЯ

1.1. ТЕХНОЛОГИИ РЕШЕНИЯ СИСТЕМ ЭКОНОМЕТРИЧЕСКИХ УРАВНЕНИЙ

1.1.1. Операции с массивами в табличном процессоре

Массив — это набор данных одного типа. Массив в MS Excel хранится в диапазоне ячеек. MS Excel позволяет создавать одномерные, двумерные и трехмерные массивы, которые хранятся соответственно в одномерных, двумерных и трехмерных диапазонах. Одномерный и двумерный диапазоны создаются на одном рабочем листе. Адресная ссылка на такой диапазон имеет формат:

<Адрес_первой_ячейки>:<Адрес_последней_ячейки>. Трехмерные диапазоны создаются в одноименных ячейках нескольких смежных рабочих листов. Адресная ссылка на такой диапазон имеет формат:

<Имя_первого_рабочего_листа>:<Имя_последнего_рабочего_листа>!<Адрес_первой_ячейки>:<Адрес_последней_ячейки>, например:

=Лист1:Лист2!\$A\$1:\$B\$4.

Если массив содержит данные арифметического типа, то с таким массивом можно выполнять арифметические операции, в которых в качестве операндов участвуют:

- массив и единственная переменная, например умножение элементов массива на число;
- двумерный массив и одномерный массив, например почленно-построчное умножение;
- массивы одинаковой размерности.

Для умножения (деления, сложения или вычисления функции) каждого элемента массива на число следует выполнить следующие действия:

- ввести значения элементов массива в ячейки рабочего листа;
- выделить область для размещения результата операции, ее размерность должна быть такой же, как размерность исходного массива;
- в строку формул ввести формулу, например: =A1:B2*3 или =sin(A1:B2) и т.п.;

– указать, что производится операция над массивом, нажав комбинацию клавиш <Ctrl> + <Shift> + <Enter>.

Пример 1.1. Требуется умножить элементы массива размерностью

2×2 на число 3. Исходный массив $\begin{pmatrix} 10 & 5 \\ 4 & 6 \end{pmatrix}$.

Решение

Введем в ячейки диапазона значения элементов массива.

Выделим диапазон ячеек такой же размерности, в который будет помещаться результат операции.

Введем в выделенный диапазон формулу в формате:

=<адрес_начальной_ячейки_диапазона>:<адрес_конечной_ячейки>*<адрес_второго_операнда>.

Нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter>.

В ячейках выделенного диапазона появится результат.

	D1	fx {=A1:B2*3}				
	A	B	C	D	E	
1	10	5		30	15	
2	4	6		12	18	
3						

1.1.2. Технологии операций с векторами

Вычисление суммы векторов

Вектора и матрицы в электронной таблице хранятся в виде массивов.

Известно, что сумма векторов – это вектор, координаты которого равны суммам соответствующих координат исходных векторов:

$$\bar{c} = \bar{a} + \bar{b} = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n).$$

Для вычисления суммы векторов нужно выполнить следующую последовательность действий:

– в диапазоны ячеек одинаковой размерности ввести значения числовых элементов каждого вектора;

– выделить диапазон ячеек для вычисляемого результата такой же размерности, что и исходные векторы;

– ввести в выделенный диапазон формулу перемножения диапазонов =Адрес_Вектора_1*Адрес_Вектора_2;

– нажать комбинацию клавиш <Ctrl> + <Shift> + <Enter>.

Пример 1.2. Даны два вектора:

$$\bar{a}_1 = (1, 2, 0), \quad \bar{a}_2 = (2, 1, 1).$$

Требуется вычислить сумму этих векторов.

Решение

В ячейки диапазона A2:A4 введем значения координат вектора \bar{a}_1 , а в ячейки диапазона C2:C4 — координаты вектора \bar{a}_2 .

Выделим ячейки диапазона, в которых будет вычисляться результирующий вектор \bar{c} (E2:E4) и введем в выделенный диапазон формулу
=A2:A4+C2:C4.

Нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter>. В ячейках диапазона E2:E4 будут вычислены соответствующие координаты результирующего вектора.

	E2		fx {=A2:A4+C2:C4}		
	A	B	C	D	E
1	a₁		a₂		c
2	1		2		3
3	2		1		3
4	0		1		1
5					

Вычисление произведения вектора на число

Произведением вектора на число является вектор, координаты которого получаются умножением соответствующих координат исходного вектора на это число:

$$\bar{c} = \lambda \bar{a} = (\lambda a_1, \lambda a_2, \dots, \lambda a_n).$$

Для вычисления произведения вектора на число нужно выполнить следующую последовательность операций:

— в диапазон ячеек рабочего листа ввести числовые значения элементов вектора;

— в ячейку ввести значение числа, на которое нужно умножить вектор λ ;

— выделить диапазон ячеек такой же размерности, что и исходный вектор для вычисляемого результата;

— ввести в выделенный диапазон формулу перемножения диапазонов

=Адрес_Вектора_1*Адрес_Числа;

— нажать комбинацию клавиш <Ctrl> + <Shift> + <Enter>.

Вычисление скалярного произведения векторов

Известно, что скалярное произведение векторов — это сумма произведений соответствующих координат этих векторов:

$$\overline{ab} = (a_1 b_1 + a_2 b_2 + \dots, a_n b_n).$$

Для вычисления скалярного произведения векторов нужно применить следующую последовательность операций:

- в диапазоны ячеек одинаковой размерности ввести значения числовых элементов каждого вектора;
- выделить диапазон ячеек для вычисляемого результата такой же размерности, что и исходные диапазоны;
- ввести в выделенный диапазон формулу перемножения диапазонов

$$=СУММ(\text{Адрес_Вектора_1}*\text{Адрес_Вектора_2}).$$

Пример 1.3. Даны два вектора:

$$\overline{a}_1 = (1, 2, 0), \quad \overline{a}_2 = (2, 1, 1).$$

Требуется вычислить скалярное произведение этих векторов.

Решение

В ячейки диапазона A2:A4 введем значения координат вектора \overline{a}_1 , а в ячейки диапазона C2:C4 — координаты вектора \overline{a}_2 .

В ячейку, в которой нужно получить результат, например E2, введем формулу =СУММ(A2:A4*C2:C4) и нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter>. В результате вычисления будет получен результат — 4.

E2		fx {=СУММ(A2:A4*C2:C4)}				
	A	B	C	D	E	
1	a₁		a₂		ab	
2	1		2		4	
3	2		1			
4	0		1			
5						

1.1.3. Технологии операций с матрицами

Умножение матрицы на число

Из теории линейной алгебры известно, что прямоугольная таблица чисел вида

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{mn} \end{pmatrix}$$

называется матрицей.

Здесь a_{ij} — действительные числа, называемые элементами матрицы, i и j — соответственно индексы строки и столбца.

Двумерные матрицы в электронной таблице могут быть представлены массивами значений их элементов, записанных в двумерных диапазонах.

Произведением матрицы A на действительное число α называется матрица, каждый элемент которой получен умножением соответствующего элемента матрицы A на число α .

Рассмотрим технологию умножения матрицы на число на примере.

Пример 1.4. Нужно умножить матрицу $A = \begin{pmatrix} 1 & -2 \\ 9 & -1 \end{pmatrix}$ на число 4,

иначе говоря, получить матрицу $C = A \times 4$.

Решение

В ячейки рабочего листа введем элементы матрицы, например в диапазон A2:B3.

Выделим диапазон, в котором будет вычисляться результирующая матрица, например D2:E3.

В выделенный диапазон введем формулу $=A2:B3*4$.

Нажмем комбинацию клавиш <Ctrl> + <Shif> + <Enter> (такая комбинация указывает, что должна быть выполнена операция с массивом ячеек). После выполнения операций в диапазоне ячеек D2:C3 будут помещены результаты вычислений.

	A	B	C	D	E
1	Матрица A			Матрица C	
2	1	-2		4	-8
3	9	-1		36	-4

Суммирование и вычитание матриц

Суммой матриц A и B одинаковой размерности называется матрица C такой же размерности, каждый элемент которой равен сумме

соответствующих элементов матриц A и B . Например, если $A = \begin{pmatrix} 1 & 2 \\ 9 & -1 \end{pmatrix}$

и $B = \begin{pmatrix} 0 & -4 \\ 5 & 16 \end{pmatrix}$, то $C = \begin{pmatrix} 1+0 & 2-4 \\ 9+5 & -1+16 \end{pmatrix} = \begin{pmatrix} 1 & -2 \\ 14 & 15 \end{pmatrix}$.

Пример 1.5. Сложить матрицы $A = \begin{pmatrix} 1 & 2 \\ 9 & -1 \end{pmatrix}$ и $B = \begin{pmatrix} 0 & -4 \\ 5 & 16 \end{pmatrix}$.

Решение

Введем значения элементов матрицы A в диапазон A2:C3, а элементы матрицы B — в диапазон D2:E3.

Выделим диапазон, где будут размещаться элементы результирующей матрицы C , например G2:H3.

В выделенный диапазон введем формулу =A2:B3 + D2:E3.

Нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter>. После выполнения операций в диапазоне ячеек G2:H3 будут помещены результаты вычисленных значений элементов результирующей матрицы C .

	G2	fx {=A2:B3+D2:E3}						
	A	B	C	D	E	F	G	H
1	Матрица A			Матрица B			Матрица C	
2	1	2		0	-4		1	-2
3	9	-1		5	16		14	15
4								

Подобным образом вычисляется разность матриц, а также их скалярное произведение (деление).

В библиотеке Excel в категории математических функций есть функции для выполнения операций над матрицами, список которых приведен в табл. 1.1.

Все перечисленные функции, кроме функции ТРАНСП, размещены в Мастере функций в группе *Математические функции*. Функция ТРАНСП находится в группе функций *Ссылки и массивы*.

Параметрами функций, приведенных в таблице, могут быть адресные ссылки на массивы, содержащие значения элементов матриц, или имена диапазонов, например МОБР(A1:B2), или МОПР(матрица_A).

Встроенные функции MS Excel для работы с матрицами

Русифицированное имя функции	Англоязычное имя функции	Выполняемое действие
МОБР (параметр)	MINVERSE (parametr)	Обращение матрицы
МОПР (параметр)	MDETERM (parametr)	Вычисление определителя матрицы
МУМНОЖ (список параметров)	MMULT (parametrlist)	Умножение матриц
ТРАНСП (параметр)		Транспонирование матриц

Вычисление произведения матриц

Произведение матриц может быть вычислено, если количество столбцов умножаемой матрицы равно количеству строк матрицы множителя.

Если $A = (a_{ij})$ имеет размерность $m \times n$, и $B = (b_{ij})$ с размерностью $n \times p$, то матрица C , полученная умножением матрицы A на матрицу B , будет иметь размер $m \times p$, а каждый ее элемент будет равен сумме произведений i -й строки матрицы A на соответствующие элементы j -го столбца матрицы B :

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{ip}b_{pj} = \sum_{k=1}^p a_{ik}b_{kj}; \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, p.$$

Вычисление произведения матриц в табличном процессоре выполняется с помощью специальной функции рабочего листа, имя которой МУМНОЖ. Она имеет синтаксис:

$$\text{МУМНОЖ}(\text{Массив1}; \text{Массив2}),$$

где *Массив1* — адрес диапазона, в котором записаны элементы первой матрицы;
Массив2 — адрес диапазона, в котором записаны элементы второй матрицы.

Пример 1.6. Умножить матрицу $A = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}$ на матрицу $B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$.

Решение в электронной таблице приведено на рисунке.

G2		fx {=МУМНОЖ(A2:B3;D2:E3)}						
	A	B	C	D	E	F	G	H
1	Матрица А			Матрица В			Результат	
2	2	3		1	2		11	16
3	3	4		3	4		15	22
4								

Решение

Присвоим диапазону A3:B4 имя (например, A) и введем в ячейки значения элементов матрицы A.

Присвоим диапазону D3:D4 имя (например, B) и введем значения элементов вектора B.

Выделим область F3:F4 для помещения результата решения системы и введем в него формулу

=МУМНОЖ(МОБР(A);B).

Нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter>, в ячейках диапазона F3:F4 будет получен результат.

	F3	fx {=МУМНОЖ(МОБР(A3:B4);D3:D4)}				
	A	B	C	D	E	F
1						
2	Матрица A			Вектор B		Результат
3	2	1		3		2,166666667
4	4	5		2		-1,333333333
5						

Чтобы выполнить проверку полученных результатов, достаточно перемножить исходную матрицу на вектор результата, итогом этой операции является массив, содержащий такие же значения, как и вектор B.

При решении некоторых задач в электронной таблице в формулах удобно использовать ссылки на имена ячеек или диапазонов. Имя — это идентификатор. Область действия имени — вся рабочая книга. Для присвоения имени диапазону можно применить два приема:

- выделить диапазон и в поле имени записать идентификатор;
- выполнить команду меню **Вставка/Имя/Присвоить**. В открывшемся окне в поле *Имя* ввести идентификатор имени, а в поле *Формула* записать адрес диапазона (лучше в виде абсолютного адреса. Для установки абсолютного адреса используйте клавишу <F4с>.

Метод наименьших квадратов

Технология решения систем линейных уравнений для случая, когда $m = n$, рассмотрена выше. Однако в общем случае m может быть не всегда равно n . Возможны три случая: $m < n$, $m = n$ и $m > n$. При $m < n$, если система является совместной, то она не определена и имеет бесконечное множество решений.

В случае если $m > n$ и система совместна, то матрица A имеет по крайней мере $m - n$ линейно независимых строк. В этом случае решение может быть получено отбором n любых линейно независимых уравнений и применением формулы $x = A^{-1} \times B$. Однако при решении задачи в электронной таблице удобнее применить более общий подход — метод наименьших квадратов. Для этого обе части уравнения нужно умножить на транспонированную матрицу системы A^T : $A^T A X = A^T B$.

Затем обе части уравнения нужно умножить на $(A^T A)^{-1}$. Если эта матрица существует, то система определена. С учетом того, что $(A^T A)^{-1} \times A^T A = E$, где E — единичная матрица, решение системы будет иметь вид $X = (A^T A)^{-1} \times A^T B$.

Рассмотрим технологию решения систем линейных уравнений методом наименьших квадратов на примере.

Пример 1.8. Требуется решить систему уравнений:

$$\begin{cases} 3x + 2y = 7 \\ 4x - 5y = 40 \\ 3x + 3y = 3. \end{cases}$$

Решение

Введем значения элементов матрицы A в диапазон ячеек рабочего листа (A2:B4).

Введем значения элементов вектора B в ячейки рабочего листа, например D2:D4.

Транспонируем исходную матрицу, для чего выделим диапазон ячеек размерностью 3×2 (A6:C7), введем в него формулу

$$= \text{ТРАНСП}(A2:B4).$$

Нажмем комбинацию клавиш $\langle \text{Ctrl} \rangle + \langle \text{Shift} \rangle + \langle \text{Enter} \rangle$ — в ячейках выделенного диапазона будут помещены элементы транспонированной матрицы.

Вычислим произведение $A^T A$, для чего выделим диапазон (A9:B10) и введем в него формулу

$$= \text{МУМНОЖ}(A6:C7; A2:B4).$$

Вычислим произведение $A^T B$, для чего выделим диапазон из двух ячеек (E6:E7) и введем в него формулу

$$= \text{МУМНОЖ}(A6:C7; D2:D4).$$

Выделим диапазон (D9:E10), введем в него формулу $= \text{МОБР}$ (A9:B10) для вычисления обратной матрицы $(A^T A)^{-1}$. Матрица существует, следовательно, исходная система определена.

Для вычисления итогового результата – решения системы уравнений выделим диапазон (B12:B13) и введем в него формулу умножения матриц $(A^T A)^{-1} \times A^T B$
 $=\text{МУМНОЖ}(D9:E10;A9:B10)$.

Нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter> – в ячейках диапазона B12:B13 будет получен результат решения системы.

B12		fx {=МУМНОЖ(D9:E10;E6:E7)}			
	A	B	C	D	E
1	Матрица A			Вектор B	
2	3	2		7	
3	4	-5		40	
4	3	3		3	
5	Матрица A транспонированная				A_TB
6	3	4	3		190
7	2	-5	3		-177
8	Произведение A_TA			Матрица (A_TA)⁻¹	
9	34	-5		0,029992	0,003946
10	-5	38		0,003946	0,026835
11	Результат (A_TA)⁻¹A_TB				
12	x	5,00			
13	y	4,00			
14					

При достаточно хорошем навыке работы с Мастером функций приведенную задачу можно решить без промежуточных вычислений, как это рассмотрено выше, а введя сразу все выражение для вычисления в строку формул.

Рассмотрим эту технологию более подробно на том же примере. Формула, которая дает решение системы $X = (A^T A)^{-1} \times (A^T B)$ содержит две группы (заключенные в скобки), которые должны быть перемножены с помощью функции МУМНОЖ(аргумент 1; аргумент 2). Аргумент 1 в нашем случае сам является вычисляемым выражением $(A^T A)^{-1}$, аргумент 2 также вычисляется – $(A^T B)$. При вводе формул, представляющих сложные выражения, целесообразно придерживаться технологии, которая предлагается далее.

Для решения задачи выполним действия.

1. Выделим диапазон, в котором будет вычисляться результат, и, используя мастер функций, введем в него функцию МУМНОЖ, переведем курсор в поле *Массив2* диалогового окна **Аргументы функции**, после чего щелкнем на кнопке *fx*, расположенной в левой части строки формул, — окно **Аргументы функции** закроется, а в строке формул появится выражение =МУМНОЖ(;) (символ «;» разделяет аргументы функции).

2. Первый аргумент в нашем случае является обратной матрицей результата произведения матриц. Переведем щелчком курсор в поле первого аргумента и, используя список *Функции*, включим функцию МОБР и закроем окно **Аргументы функции**, щелкнув на кнопке *fx* в левой части строки формул. При этом курсор остается в строке формул в поле аргумента функции МОБР.

3. Аргумент функции МОБР в рассматриваемой задаче является произведением матриц. Используя список *Функции* включим функцию МУМНОЖ. Переведем курсор в поле *Массив2* диалогового окна **Аргументы функции** и укажем адрес массива, содержащего элементы матрицы *A*. Затем переведем курсор в поле *Массив2* диалогового окна и, используя список *Функции*, включим функцию ТРАНСП и в качестве ее аргумента укажем адрес массива, содержащего элементы матрицы *A*. Щелкнем на кнопке *fx* — окно **Аргументы функции** закроется.

4. В строке формул установим курсор в область второго аргумента первой функции МУМНОЖ. Используя список *Функции*, включим функцию МУМНОЖ. Поскольку второй аргумент не нужно вычислять, в первую очередь укажем его: в поле *Массив2* введем адрес диапазона, в котором содержатся элементы вектора *B*.

5. Переключим курсор в поле *Массив2* диалогового окна **Аргументы функции**, используя список *Функции*, включим функцию ТРАНСП и в поле *Массив* этой функции укажем адрес диапазона, в котором содержатся значения элементов матрицы *A*. Запись формулы для решения системы уравнений методом наименьших квадратов завершена. Она имеет окончательный вид

=МУМНОЖ(МОБР(МУМНОЖ(ТРАНСП(A2:B4);A2:B4));
МУМНОЖ(ТРАНСП(A2:B4);D2:D4)).

6. Нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter> — в ячейках выделенного диапазона будет результат решения системы (рис. 1.1).

B6		fx {=МУМНОЖ(МОБР(МУМНОЖ(ТРАНСП(A2:В4);A2:В4)) МУМНОЖ(ТРАНСП(A2:В4);D2:D4)))}	
	А	В	
1	Матрица А		Вектор В
2	3	2	7
3	4	5	40
4	3	3	3
5	Решение		
6	х	5,00	
7	у	4,00	
8			

Рис. 1.1. Пример решения системы линейных уравнений

На первый взгляд приведенная процедура может показаться сложной и длительной. Однако это кажется только на первый взгляд. При достаточном ее освоении значительно сокращается время решения и уменьшается вероятность ошибки.

1.1.5. Применение технологий при решении экономических задач

Применение технологий при решении микроэкономических задач

Рассмотрим пример использования технологии выполнения **операций с массивами** на примере следующей задачи.

Пример 1.9. Предприятие ежедневно выпускает четыре вида изделий, их производственно-экономические показатели приведены в таблице.

Вид изделия. Условный номер	Количество выпускаемых изделий, шт.	Расход сырья, кг/изд.	Норма времени изготовления, ч/изд.	Стоимость изделия, ден. ед./изд.
	N	s	t	p
1	20	5	10	30
2	50	2	5	15
3	30	7	15	45
4	40	4	8	40

Требуется определить следующие ежедневные показатели: расход сырья S , затраты рабочего времени T и стоимость P выпускаемой продукции предприятия.

В таблице приведенные производственно-экономические показатели можно представить в виде следующих векторов: $\bar{n} = (20 \ 50 \ 30 \ 40)$ — вектор количества выпускаемых изделий по видам продукции; $\bar{s} = (5 \ 2 \ 7 \ 4)$ — вектор расхода сырья по видам продукции; $\bar{t} = (10 \ 5 \ 15 \ 8)$ — вектор затрат времени на изготовление продукции; $\bar{p} = (30 \ 15 \ 45 \ 20)$ — вектор стоимости. Тогда решение задачи будет представлять собой скалярные произведения вектора количества выпускаемой продукции \bar{n} на три других вектора: ежесуточный расхода сырья S будет вычисляться по формуле $S = \bar{n}\bar{s}$, затраты рабочего времени T — по формуле $T = \bar{n}\bar{t}$, стоимость выпускаемой продукции $P = \bar{p}\bar{n}$.

Решение задачи приведено на рисунке.

B8		fx {=СУММ(B3:B6*C3:C6)}			
	A	B	C	D	E
1	Вид изделия, условный номер	К-во выпускаемых изделий, шт.	Расход сырья, кг/изд.	Норма времени изготовления, ч/изд.	Стоимость изделия, ден. ед./изд.
2					
3	1	20	5	10	30
4	2	50	2	5	15
5	3	30	7	15	45
6	4	40	4	8	40
7	Результаты решения				
8	S	570			
9	T	1220			
10	P	4300			
11					

Экономические задачи, содержательный смысл которых заключается в прогнозировании количества выпускаемой продукции, оказания услуг и т.п. на основе известных запасов сырья или других ресурсов обычно приводятся к **системе линейных уравнений**, описывающих балансовые соотношения. Рассмотрим технологию решения подобных задач на примере.

Пример 1.10. Предприятие выпускает три вида продукции из сырья трех типов. Характеристики производства приведены в таблице.

Требуется определить возможный объем выпуска каждой продукции при заданных запасах сырья.

Таблица

Вид сырья	Расход сырья по видам продукции, вес. ед./изд.			Запас сырья, вес. ед.
	1	2	3	
1	6	4	5	2 400
2	4	3	1	1 450
3	5	2	3	1 550

Решение

Введем обозначения неизвестных объемов выпускаемой продукции: x_1, x_2, x_3 . Тогда при условии полного расхода запасов сырья и при условии отсутствия ограничений, которые определяются другими ресурсами, балансовые соотношения можно записать в виде следующей системы уравнений:

$$\begin{cases} 6x_1 + 4x_2 + 5x_3 = 2400 \\ 4x_1 + 3x_2 + x_3 = 1450 \\ 5x_1 + 2x_2 + 3x_3 = 1550. \end{cases}$$

Матрица системы и матрица свободных членов будут соответственно равны

$$A = \begin{pmatrix} 6 & 4 & 5 \\ 4 & 3 & 1 \\ 5 & 2 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 2400 \\ 1450 \\ 1550 \end{pmatrix}.$$

Количество уравнений в системе равно количеству неизвестных, поэтому для ее решения применим метод обратной матрицы. Решение системы в MS Excel может выглядеть так, как представлено на рисунке.

B6 ▼ fx {=МУМНОЖ(МОБР(A2:C4);E2:E4)}						
	A	B	C	D	E	F
1	A				B	
2	6	4	5		2400	
3	4	3	1		1450	
4	5	2	3		1550	
5	Решение					
6	x1 =	150				
7	x2 =	250				
8	x3 =	100				
9						

Применение технологии при решении макроэкономических задач

Известно, что рациональное функционирование многоотраслевого хозяйства предполагает соблюдение баланса между отраслями. Каждая отрасль многоотраслевого хозяйства является, с одной стороны, производителем определенной продукции, а с другой — потребителем продукции, выпускаемой другими отраслями. Макроэкономика функционирования многоотраслевого хозяйства требует, чтобы соблюдался баланс по производству и потреблению между отдельными отраслями.

Балансовый принцип связи различных отраслей состоит в том, что валовой выпуск i -й отрасли должен быть равен сумме объемов потребления. В простейшей форме балансовые соотношения имеют вид

$$x_i = x_{i1} + x_{i2} + \dots + x_{in} + y_i, \quad i = 1, 2, \dots, n,$$

где x_i — общий объем выпускаемой продукции i -й отрасли;
 x_{ij} — объем продукции i -й отрасли, потребляемый j -й отраслью при производстве объема продукции x_j ;
 y_i — объем продукции i -й отрасли конечного потребления (для реализации в непроизводственной сфере).

Для производства продукции j -й отрасли объемом x_j нужно использовать продукцию i -й отрасли объемом $a_{ij}x_j$, где a_{ij} — постоянное число, характеризующее прямые затраты. Это допущение позволяет представить модель многоотраслевой экономики (модель Леонтьева) в виде системы линейных уравнений, которая в матричной форме имеет вид

$$\bar{x} = A\bar{x} + \bar{y},$$

где \bar{x} — вектор валового выпуска;
 \bar{y} — вектор объема продукции конечного потребления;
 A — матрица коэффициентов прямых затрат.

Приведенная система уравнений может быть представлена в виде

$$(E - A)\bar{x} = \bar{y},$$

где E — единичная матрица.

Если существует обратная матрица $(E - A)^{-1}$ (матрица полных затрат), то существует единственное решение системы $\bar{x} = (E - A)^{-1}\bar{y}$.

Из экономической теории известно несколько критериев продуктивности матрицы A :

— матрица A продуктивна тогда и только тогда, когда матрица $(E - A)^{-1}$ существует и ее элементы неотрицательны;

— матрица A с неотрицательными элементами продуктивна, если сумма элементов по любому ее столбцу (строке) не больше единицы, причем хотя бы для одного столбца (строки) строго меньше единицы.

Рассмотрим пример решения макроэкономической задачи на применение модели Леонтьева.

Пример 1.11. В таблице приведены данные по балансу за некоторый период времени между пятью отраслями.

№ п/п	Отрасль	Потребление					Конечный продукт	Валовой выпуск, ден. ед.
		1	2	3	4	5		
1	Станкостроение	15	12	24	23	16	10	100
2	Энергетика	10	3	35	15	7	30	100
3	Машиностроение	10	5	10	10	10	5	50
4	Автомобильная промышленность	10	5	10	5	5	15	50
5	Добыча и переработка углеводородов	7	15	15	3	3	50	100

Требуется найти векторы конечного потребления и валового выпуска, а также матрицу коэффициентов прямых затрат и определить ее продуктивность.

В приведенной таблице в первых пяти столбцах (группа «Потребление») содержатся значения x_{ij} , в последнем столбце содержатся элементы вектора валового выпуска \bar{x} , в предпоследнем столбце — элементы вектора объема конечного потребления \bar{y} .

Решение

В диапазон ячеек рабочего листа (B4:F8) введем числа, записанные в столбцах «Потребление», исходной таблицы (рис. 1.13).

Введем в ячейки диапазона (H4:H8) значения элементов вектора валового выпуска X , который соответствует последнему столбцу исходной таблицы, а в диапазон (I4:I8) — значения элементов вектора Y — вектор конечного продукта.

Матрица коэффициентов прямых затрат A вычисляется путем деления i -го столбца матрицы «Потребление» на i -ю строку вектора X . Это вычисление можно выполнить, используя формулу $A = П : X^T$, где $П$ — матрица «Потребление».

Выделим диапазон ячеек, в котором будет размещаться матрица A и введем в него формулу деления массива «Потребление» на транспонированный вектор X : =B4:F8/ТРАНСП(H4:H8) и нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter>. После выполнения этой операции в выделенном диапазоне будут вычислены значения элементов матрицы коэффициентов прямых затрат A .

Просуммируем столбцы полученной матрицы A .

Вычислим значения элементов матрицы полных затрат.

Проанализируем полученные в результате расчетов данные.

Матрица полных затрат $(E - A)^{-1}$ существует, все ее элементы положительны. Следовательно, первое условие продуктивности матрицы A выполняется.

Все элементы матрицы A положительные, однако в третьем и четвертом столбцах их суммы превышают значение единицы, следовательно второе условие продуктивности матрицы A не выполняется. Таким образом, матрица коэффициентов прямых затрат в решаемой задаче непродуктивна.

	A	B	C	D	E	F	G	H	I	J	K	L	
1		i=1	i=2	i=3	i=4	i=5							
2		Матрица "Потребление" - П					Валовый выпуск	Конечное потребление					
3							X	Y	j=1				
4		15	12	24	23	16	100	10	j=2				
5		10	3	35	15	7	100	30	j=3				
6		10	5	10	10	10	50	5	j=4				
7	=B4:F8/ТРАНСП(Н4:Н8)	10	5	10	5	5	50	15	j=5				
8		7	15	15	3	3	100	50	j=6				
9													
10		Матрица коэффициентов прямых затрат A					Единичная матрица E						
11	=СУММ(B11:F15)	0,15	0,12	0,48	0,46	0,16	1						
12		0,1	0,03	0,7	0,3	0,07		1					
13		0,1	0,05	0,2	0,2	0,1			1				
14		0,1	0,05	0,2	0,1	0,05				1			
15		0,87	0,15	0,3	0,06	0,03					1		
16	Сумма	0,52	0,4	1,88	1,12	0,41							
17	=МОБР(Н11:L15-B11:F15)	Матрица полных затрат $(E-A)^{-1}$											
18		1,708756	0,491571	2,068446	1,537531	0,609827							
19		0,571054	1,337135	1,986722	1,209601	0,457856							
20		0,373667	0,235835	1,983855	0,731847	0,3209							
21		0,323914	0,199508	0,843403	1,546225	0,234478							
22		0,347223	0,327527	1,122227	0,619995	1,25949							

1.2. МОДЕЛИРОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ И РЯДОВ

1.2.1. Создание массива элементов числовой последовательности

При решении некоторых задач необходимо использовать последовательности. Различают два вида последовательностей — числовые и функциональные.

Числовые последовательности представляют собой множества чисел. Если каждому числу n из натурального ряда чисел $1, 2, 3, \dots, n$ поставлено в соответствие вещественное число x_n , то множество чисел $x_1, x_2, x_3, \dots, x_n$ называют числовой последовательностью. Числа $x_1, x_2, x_3, \dots, x_n$ называют членами последовательности, элемент x_n — общим элементом, а число n — его номером. Таким образом, числовая после-

довательность представляет собой множество пронумерованных элементов.

Говорят, что последовательность задана, если известен способ получения любого ее элемента.

Последовательность обозначается символом $\{xn\}$. Например, символ $\{1/n\}$ обозначает последовательность чисел $1, 1/2, 1/3, 1/4, \dots, 1/n$.

В общем случае для создания массива элементов последовательности в табличном процессоре нужно выполнить следующие действия:

- создать массив, содержащий множество чисел натурального ряда. Каждый элемент этого массива является номером элемента создаваемой числовой последовательности;

- ввести в ячейку формулу последовательности, делая в ней адресные ссылки на ячейки, содержащие номера элементов последовательности;

- скопировать введенную формулу во все другие ячейки диапазона, в котором формируется числовая последовательность.

Для иллюстрации приведенной технологии на рис. 1.2а приведен пример создания последовательности $\{1/n\}$, а на рис. 1.2б – последовательности $\{n/(n+1)\}$ для семи элементов.

ВЗ		fx = 1/A3	
	А	В	
1	Последовательность (1/n)		
2	Номер элемента	Значение элемента	
3	1	1	
4	2	1/2	
5	3	1/3	
6	4	1/4	
7	5	1/5	
8	6	1/6	
9	7	1/7	
10			

а)

ВЗ		fx = A3/(A3+1)	
	А	В	
1	Последовательность (n/(n+1))		
2	Номер элемента	Значение элемента	
3	1	1/2	
4	2	2/3	
5	3	3/4	
6	4	4/5	
7	5	5/6	
8	6	6/7	
9	7	7/8	

б)

Рис. 1.2. Примеры создания числовых последовательностей:
а) $\{1/n\}$; б) $\{n/(n+1)\}$

Для создания наиболее часто встречающихся последовательностей, таких как арифметическая или геометрическая прогрессия, табличный процессор имеет специальный инструмент **Прогрессия**, кото-

рый включается командой меню **Правка/Заполнить/Прогрессия**. Для создания последовательности с помощью этого инструмента нужно:

- ввести значение первого элемента прогрессии в ячейку рабочего листа;
- выделить диапазон ячеек для членов прогрессии;
- выполнить команду меню **Правка/Заполнить/Прогрессия**;
- в появившемся окне диалога *Прогрессия* указать тип и параметры создаваемой последовательности (рис. 1.3).

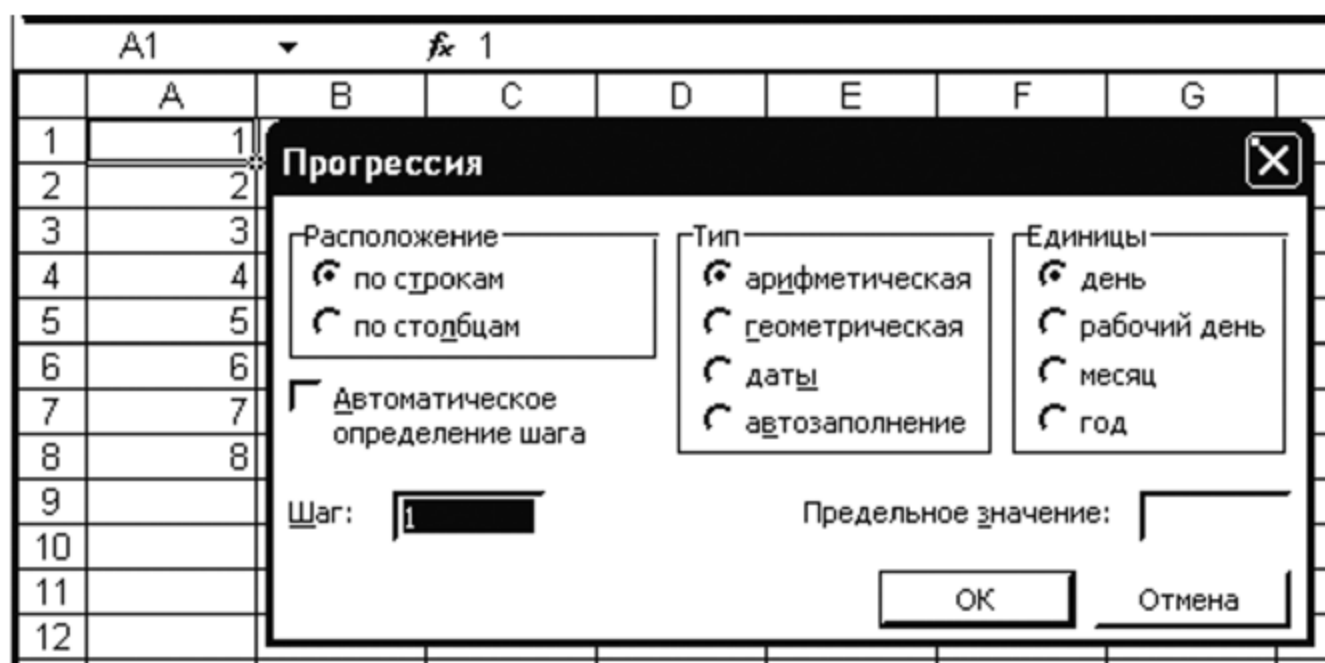


Рис. 1.3. Диалоговое окно *Прогрессия*

1.2.2. Приближенное вычисление пределов числовых последовательностей

Технологию приближенного вычисления предела числовой последовательности рассмотрим на примере.

Пример 1.12. Найти предел числовой последовательности

$$\lim_{n \rightarrow \infty} \frac{n}{n+1}$$

Решение

Полагая, что в ячейке A2 будет находиться число n , в ячейку рабочего листа B2 введем формулу $=A2/(A2+1)$.

В ячейку A2 введем большое число, примерно равное 1×10^{12} , но не более высокого порядка (в противном случае может наступить переполнение разрядной сетки процессора ПК и результат получится неправильным). После ввода числа в ячейке B2 отобразится приближенное значение предела числовой последовательности.

	B2	fx =A2/(A2+1)		
	A	B	C	D
1	Предел последовательности n/(n+1)			
2	9999999999	1		
3				
4				

1.2.3. Применение последовательностей в экономических моделях

Рассмотрим примеры практического применения пределов числовых последовательностей в экономике и финансах.

Известно, что формула сложных процентов имеет вид

$$Q_k = Q_0 \left(1 + \frac{p}{100} \right)^k,$$

- где Q_0 – первоначальная сумма вклада в банк;
 p – процент начисления за определенный период времени;
 k – количество периодов времени хранения вклада;
 Q_k – сумма вклада по истечении k периодов.

Если полагать, что проценты начисляются непрерывно, то справедлива формула

$$Q = Q_0 \left[\lim_{x \rightarrow \infty} \left(1 + \frac{p}{n} \right)^n \right]^m,$$

где $m = kp/100$ – процентная ставка, вычисленная за весь расчетный период и выраженная десятичной дробью.

Пример 1.13. Пусть начальный вклад равен 1000 денежных единиц, процентная ставка составляет 10% годовых, начисление процентов непрерывное. Требуется определить, какая сумма вклада будет по истечении двух лет при условии, что финансовый год равен 360 дням.

Решение

Ставка за весь период составит $m = 10\% \times 2 : 100 = 0,2$.

Введем исходные данные на рабочий лист, как показано на рисунке, и число n – достаточно большое.

В ячейку E2 введем формулу для вычисления числа $m = B2 / 360 * C2 * 360$.

В ячейку F2 введем формулу для вычисления суммы вклада по истечении двух лет

$$= \$D\$2 * ((1 + 1 : \$A\$2) ^ \$A\$2) ^ E2.$$

F24 ▼ f_x						
	A	B	C	D	E	F
	Число n	Ставка годовая	Срок (г)	Начальная сумма	m	Результат
1						
2	9999999999	0,1	2	1000	=B2/360*C2*360	=D\$2*((1+1/\$A\$2)^\$A\$2)
3						

Результат вычисления приведен на рисунке – $Q_k = 1221,40278$ руб.

F2 ▼ $f_x = \$D\$2*((1+1/\$A\$2)^{\$A\$2})^E2$						
	A	B	C	D	E	F
	Число n	Ставка годовая	Срок (г)	Начальная сумма	m	Результат
1						
2	9999999999	0,1	2	1000	0,2	1221,40278
3						

1.2.2. Применение рядов в экономических моделях

Числовые ряды

Напомним, что числовым рядом называется бесконечная последовательность чисел $u_1, u_2, \dots, u_n \dots$, соединенных знаком сложения:

$$u_1 + u_2 + \dots + u_n + \dots = \sum_{n=1}^{\infty} u_n.$$

Ряд считается заданным, если известен его общий член $u_n = f(n)$. Сумма n первых членов ряда называется частичной суммой ряда. Для вычисления частичной суммы ряда в табличном процессоре нужно выполнить следующие шаги:

- вычислить n первых членов числовой последовательности;
- вычислить частичную сумму членов числовой последовательности.

Вычисление функциональных рядов

В отличие от числовых рядов членами функционального ряда являются функции. Ряд, составленный из функций одной и той же переменной x

$$f_1(x) + f_2(x) + \dots + f_n(x) + \dots = \sum_{n=1}^{\infty} f_n(x),$$

называется функциональным.

Функциональные ряды находят практическое применение в финансовых вычислениях. Например, в задаче о сложных процентах при вкладе в банк Q_0 денежных единиц с ежегодной выплатой p процентов годовых функциональный ряд годовых приростов будет иметь вид

$$Q_0 + Q_0p + Q_0(1+p)p + \dots + Q_0(1+p)^n - 1p + \dots = Q_0(1+p)^n.$$

Для вычисления частичных сумм этого ряда в библиотеке табличного процессора есть специальная функция БС. Кроме того, есть несколько функций, предназначенных для вычисления различных параметров такого ряда. Так, функция КПЕР позволяет вычислить число членов ряда n по его частичной сумме, функция ПС вычисляет начальное значение Q_0 при заданном числе членов ряда n , частичной сумме ряда и величине процентной ставки.

Пример 1.14. В начале года цена товара составляла 1000 ден. ед. Инфляционные процессы в течение года по кварталам представлены в таблице.

Квартал	Инфляция по отношению к предыдущему периоду (%)
I	3
II	2
III	3
IV	3

Требуется:

- 1) определить, какова должна быть цена товара в конце года, чтобы компенсировать влияние инфляции;
- 2) определить, какова будет реальная цена товара в ценах на начало года, если фактическая ее цена изменяться с целью компенсации влияния инфляции не будет.

Решение

Для решения задачи нужно определить итоговый процент инфляции на конец года. Он определится из следующей зависимости

$$I = \sum_{k=1}^4 I_k + I_k \times I_{k+1},$$

где I_1 — начальная цена товара, равная 100%;
 I_k — процент инфляции за k -й период.

Нетрудно заметить, что приведенное выражение вычисляет частичную сумму ряда.

Решение и полученный результат приведены на рисунке.

	А	В	С		А	В	С
1	Модель вычисления			1	Результат вычисления		
2	Квартал	Инфляция (%)	Изменение цены товара с учетом инфляции	2	Квартал	Инфляция (%)	Изменение цены товара с учетом инфляции
3	1	0,03	=1+1*В3	3	1	3,00%	103,00%
4	2	0,02	=С3+С3*В4	4	2	2,00%	105,06%
5	3	0,03	=С4+С4*В5	5	3	3,00%	108,21%
6	4	0,03	=С5+С5*В6	6	4	3,00%	111,46%
7		Инфляция	=С6-1	7		Инфляция	11,46%
8		Реальная цена товара без компенсации (в процентах к первоначальной)	=1-С7	8		Реальная цена товара без компенсации (в процентах к первоначальной)	88,54%

1.3. МОДЕЛИРОВАНИЕ И ИССЛЕДОВАНИЕ ФУНКЦИЙ

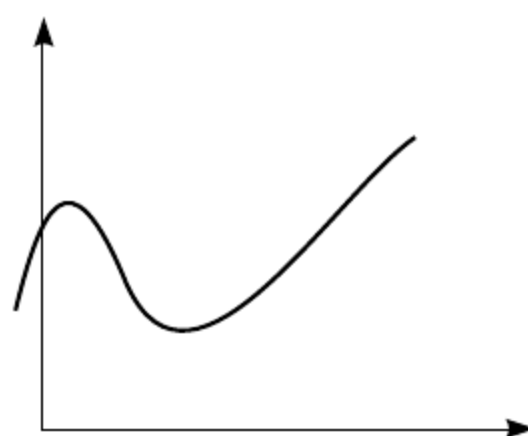
1.3.1. Способы задания функций

Функция — это модель, устанавливающая характер зависимости какой-либо одной величины (зависимой переменной) от другой величины или нескольких других величин (аргументов). Функция может быть задана таблично, в виде графика или аналитически (в виде формулы).

Табличный способ задания функций имеет широкое распространение в различных областях знаний и приложениях — экспериментальных измерениях, таблицах бухгалтерской отчетности и банковской деятельности, статистических данных и т.п. В табличном представлении одна из переменных представляется как независимая, другие величины будут являться функциями этого аргумента. Каждому значению независимой переменной X соответствует значение функции Y , записанное в той же строке таблицы (рис. 1.4а).

Независимая переменная	Функция
X	Y
-1	1
-0,8	0,64
-0,6	0,36
-0,4	0,16
-0,2	0,04
0	0
0,2	0,04
0,4	0,16
0,6	0,36
0,8	0,64
1	1

а)



б)

Рис. 1.4. Способы задания функций: а) табличный; б) графический

Графическое представление функции, позволяет наглядно представить характер функции (рис. 1.4б).

Аналитический способ задания функции заключается в задании связи между аргументом в виде формулы или системы формул, например $Y = x^2$.

1.3.2. Технология построения графической модели функции

Если функциональная зависимость задана таблично или аналитически, то в ряде случаев бывает целесообразно для исследования функциональной зависимости представить ее графически.

График — это графическая модель, отображающая характер зависимости значения функции от значения ее аргумента.

Графики функций и диаграммы в табличном процессоре создаются с помощью мастера диаграмм, который включается командой меню **Вставка/Диаграмма** или щелчком на соответствующей кнопке пиктографического меню.

График (диаграмма) табличного процессора имеет несколько взаимосвязанных элементов. С точки зрения объектного представления он является составным и включает несколько объектов (рис. 1.5). В их число входят:

- область диаграммы — объект, в котором могут размещаться все другие объекты диаграммы;
- область построения диаграммы — объект, в котором размещаются ряды и линии сетки;
- ось категорий (аргумента);
- ось значений;
- область названия оси категорий;
- область названия оси значений;
- область заголовка диаграммы;
- область легенды.

Все перечисленные элементы являются объектами, поэтому к ним применимы операции над объектами — их можно маркировать (выделять), перемещать, удалять, изменять свойства. Некоторые объекты, такие как области заголовков, являются объектами типа *Надпись*. В них можно помещать текстовые данные, редактировать или удалять их.

Исходными данными для построения графика в табличном процессоре является табличная модель функции. Если функция задана таблично, то можно сразу приступить к построению графика. Если функция задана аналитически, то, прежде чем строить график, необходимо рассчитать значения функции для некоторого множества значений ее аргумента, принадлежащих области определения функции.

Подобный расчет для значений аргумента, изменяющихся с одинаковым шагом, часто называют табулированием функции.



Рис. 1.5. Элементы графика табличного процессора.

Целью табулирования является получение координат узловых точек. Принцип построения графика заключается в размещении узловых точек функции $Y = f(x)$ в декартовой системе координат и соединения их отрезками прямых, т.е. при построении графика осуществляется линейная интерполяция для промежуточных точек.

Рассмотрим последовательность построения графической модели функции на примере.

Пример 1.15. Требуется построить графическую модель функции одной переменной в диапазоне изменения значений ее аргумента $[-1; +1]$. Математическая модель функции задана выражением

$$Y = x^3 - 0,01x^2 - 0,7044x + 0,139104.$$

Решение

Табулируем функцию, используя заданную математическую модель, для чего:

– в ячейку A2 рабочего листа введем текст «Значение аргумента», а в ячейку B2 – «Значение функции»;

— в ячейку В3 введем формулу, реализующую математическую модель заданной функции,

$$=A3^3 - 0,01*A32^2 - 0,7044*A3 + 0,13910$$

(обратите внимание, что в формуле есть адресная ссылка на ячейку, которая содержит начальное значение аргумента X);

— в ячейку А3 запишем число -1 (левая граница диапазона значений аргумента);

— применяя операцию создания последовательности, заполним диапазон ячеек А3:А13 значениями арифметической прогрессии с шагом 0,2 (конечное значение $+1$);

— скопируем, используя маркер буксировки, формулу, записанную в ячейку В3, до ячейки В13 включительно — в ячейках диапазона В3:В13 появятся вычисленные значения функции для каждого значения аргумента.

Построим график функции, для чего:

— выделим область А2:В13, содержащую табличные значения функции и надписи столбцов;

— выполним команду меню **Вставка/Диаграмма** (или щелкнем на кнопке Мастер диаграмм пиктографического меню) — откроется диалоговое окно **Мастер диаграмм**. Построение диаграммы Мастер диаграмм выполняет за четыре шага.

Шаг 1

— в окне Мастер диаграмм выберем закладку *Стандартные*, в списке *Тип* выберем *Точечная*, в окне *Вид* выберем вид графика;

— после выполнения этих операций щелкнем на кнопке *Далее*;

Шаг 2

— в окне Мастер диаграмм выберем закладку *Диапазон данных* и в поле *Диапазон* проверим корректность указанного исходного диапазона. Если он задан неверно, то установим курсор в поле *Диапазон* и, удерживая левую клавишу мыши, выделим область рабочего листа, содержащую исходную таблицу вместе с заголовками;

— выберем закладку *Ряд*. В списке *Ряд* выберем *Значение аргумента* и щелкнем на кнопке *Удалить*;

— в поле *Имя* установим адрес ячейки, содержащей текст названия таблицы (А1);

— в поле *Значения X* укажем диапазон, содержащий значения аргумента (А3:А13), а в поле *Значения y* — диапазон значений функции. Щелкнем на кнопке *Далее*.

Шаг 3

— в окне Мастер диаграмм выберем закладку *Заголовки*. В поле *Ось X* (категорий) укажем название оси категорий (значение аргумен-

та), в поле *Ось y* (значений) — название оси значений (значение функции), в поле *Название диаграммы* укажем название диаграммы;

— выберем закладку *Оси* и установим флажки для отображения основных осей;

— выберем закладку *Подписи данных* и, если необходимо, установим переключатель *Значение* (включает подписи значений функции) или *Категория* (включает подписи значений аргумента). Если необходимо, установим флажок *Ключ легенды* (включает разметку точных значений функции или аргумента);

— выберем закладку *Таблица данных* и, если необходимо, установим флажок *Таблица данных* — в области построения диаграммы отобразится исходная таблица;

— выберем закладку *Линии сетки* и установим соответствующие флажки;

— выберем закладку *Легенда* и для определения положения легенды в области построения диаграммы установим соответствующий переключатель;

— щелкнем на кнопке *Далее*.

Шаг 4

— укажем, где будет помещаться диаграмма — на отдельном листе или на имеющемся, и щелкнем на кнопке *Готово*. Построение диаграммы завершено.

При необходимости отредактируем диаграмму.



Необходимость редактирования диаграммы возникает в следующих случаях:

- размер диаграммы (составляющих ее элементов) или ее пропорции, положение на рабочем листе или положение элементов диаграммы внутри области построения не удовлетворяют каким-либо требованиям;

- была неверно задана исходная таблица;

- на диаграмме присутствуют (отсутствуют) лишние (нужные) элементы и в других случаях.

Для изменения размера диаграммы (или элемента) или изменения ее положения выполним действия:

- маркируем (выделяем) соответствующий объект;

- для изменения размера объекта поместим указатель мыши на маркер (небольшой прямоугольник темного цвета) и отбуксируем его в нужном направлении;

- для изменения положения объекта установим указатель мыши в область соответствующего объекта и, удерживая левую клавишу мыши, переместим указатель в нужном направлении.

Чтобы исправить ошибку в определении исходной таблицы, выполним действия:

- поместим указатель мыши в область построения диаграммы и выполним команду **Исходные данные** контекстного меню;

- в открывшемся диалоговом окне укажем диапазон, содержащий исходную таблицу.

Чтобы удалить или добавить элементы в область построения диаграммы или изменить их формат, маркируем соответствующий объект и нажмем клавишу $\langle \text{Del} \rangle$, объект будет удален.

Для изменения формата объекта выполним соответствующую команду контекстного меню.

Чтобы добавить элемент в область построения диаграммы, выполним команду контекстного меню **Параметры диаграммы**.

1.3.3. Вычисление предела функции

Напомним, что функция $f(x)$ имеет предел в точке a тогда и только тогда, когда в этой точке существуют левый и правый пределы, причем они равны. В математике для нахождения пределов функций применяются специальные приемы, в частности такой, как разложение числителя и знаменателя на сомножители, и некоторые другие.

В табличном процессоре для нахождения предела функции можно применить следующую технологию:

- в ячейку рабочего листа ввести формулу, соответствующую выражению функциональной зависимости, в которой значение ар-

гумента указывается адресной ссылкой на ячейку, которая содержит аргумент;

– в ячейку, предназначенную для записи аргумента функции, ввести число, максимально близкое к точке, в которой вычисляется предел функции.

Технологию вычисления пределов функции рассмотрим на примере.

Пример 1.16. Найти пределы: $\lim_{x \rightarrow 2} \frac{x^2 - 5x + 6}{x^2 - 3x + 2}$, $\lim_{x \rightarrow \infty} \frac{4x + 2}{x + 1}$.

Решение

Введем в ячейку рабочего листа A4 значение достаточно близкое к значению 2, например слева – 1,99999999999.

В ячейку B4 введем формулу, реализующую аналитическое выражение функции

$= (A4^2 - 5 * A4 + 6) / (A4^2 - 3 * A4 + 2)$. После вычисления в ячейке B4 будет отображено приближенное значение предела функции.

B4					f = (A4^2 - 5 * A4 + 6) / (A4^2 - 3 * A4 + 2)				
	A	B	C	D					
1	Предел функции								
2	x	Предел				Функция			
3	Значение слева								
4	1,9999999999	-1,0000000000				$\lim_{x \rightarrow 2} \frac{x^2 - 5x + 6}{x^2 - 3x + 2}$			
5									
6	Значение справа								
7	2,00000000010	-1,0000000000							
8									
9	9999999999999	4,0000000000				$\lim_{x \rightarrow \infty} \frac{4x + 2}{x + 1}$			
10									

Аналогичным образом происходит вычисление предела $\lim_{x \rightarrow \infty} \frac{4x + 2}{x + 1}$.

Обратите внимание на то, что значение x в ячейке A9 задано достаточно большое.

1.3.4. Вычисление корней функции одной переменной

Корнями функции $Y = f(x)$ называют такие значения x , при которых функция принимает значение ноль.

Процесс нахождения корней функции, как правило, осуществляется в два этапа. На первом этапе отделяются корни, т.е. находят такие отрезки, внутри которых находится строго один корень. На втором

этапе производится уточнение корней, т.е. находят их значение с заданной точностью (ϵ). В практических задачах решением является значение x , отличающееся по модулю от точного значения не более чем на величину ϵ .

При решении практических задач величина x являются каким-либо ресурсом, величина которого ограничена и лежит в области допустимого диапазона значений. Поэтому при решении задачи интерес представляют только те корни, которые находятся в области возможных значений x .

Отделение корней функции в ограниченной области определения переменной x в табличном процессоре можно выполнить, используя ее табличную или графическую модель.

Для отделения корней функции нужно выполнить следующие операции:

- табулировать функцию, задавая значения аргумента в диапазоне допустимых значений аргумента;
- построить график функции и определить, где находятся точки пересечения графика функции с осью x ;
- в полученной табличной модели найти ближайшие приближения к значениям корней. Ближайшими приближениями являются те значения аргумента, в промежутке между которыми значение функции изменяет знак.

Уточнение значений корней можно выполнить с помощью одного из двух инструментов – **Подбор параметра** или **Поиск решения**. Оба эти инструмента используют итерационные методы и позволяют получить результат с заданной точностью. Для уточнения корней с помощью инструмента **Подбор параметра** нужно выполнить следующие операции.

1. Выполнить настройку табличного процессора, для этого:
 - выполнить команду меню **Сервис/Параметры**;
 - в открывшемся диалоговом окне **Параметры** выбрать закладку *Вычисления*;
 - в открывшемся диалоговом окне **Вычисления** установить флажок *Итерации*, в поле *Предельное число итераций* установить нужное число итераций, в поле *Относительная погрешность* ввести величину относительной погрешности вычислений;
 - щелкнуть на кнопке ОК.
2. Используя инструмент табличного процессора **Подбор параметра**, вычислить корни уравнения с заданной точностью.

Рассмотрим технологию вычисления корней функции на примере.

Пример 1.17. Найти все корни функции

$$y = x^3 - 0,01x^2 - 0,7044x + 0,139104$$

в диапазоне значений аргумента $[-1; 1]$.

Решение

Заданная функция представлена полиномом третьей степени, следовательно, она может иметь не более трех корней.

Для локализации начальных приближений определим интервалы значений X , внутри которых значение функции пересекает ось абсцисс, т.е. функция меняет знак. С этой целью табулируем функцию на отрезке $[-1; +1]$ с шагом 0,2, получим табличные значения функции. Просмотрев полученную таблицу, находим, что график функции трижды пересекает ось X , следовательно, исходное уравнение имеет на заданном отрезке все три корня.

Анализ таблицы показывает, что функция меняет знак в следующих интервалах значений аргумента X : $(-1; -0,8)$, $(0,2; 0,4)$ и $(0,6; 0,8)$, следовательно, корни функции лежат внутри этих интервалов. Поэтому в качестве начальных приближений возьмем значения X : $-0,8$; $0,2$ и $0,6$.

На свободном участке рабочего листа в диапазон ячеек (A16:A18) введем начальные приближения, а в соответствующие ячейки столбца B введем формулу, реализующую функциональную зависимость.

Выполним команду меню **Сервис/Параметры**, во вкладке *Вычисления* открывшегося диалогового окна установим относительную погрешность вычислений $\varepsilon = 0,000001$, а число итераций $N = 1000$, установим флажок *Итерации*.

Выполним команду меню **Сервис/Подбор параметра**. В диалоговом окне заполним следующие поля:

– *Установить в ячейке*: в поле укажем адрес ячейки, в которой записана формула правой части функции (B16);

– *Значение*: в поле укажем значение, которому должно удовлетворять значение функции, т.е. правая часть уравнения (в нашем случае 0);

– *Изменяя значение*: в поле укажем адрес ячейки (где записано начальное приближение A16), в которой будет вычисляться корень уравнения и на которую ссылается формула. После щелчка на кнопке ОК в ячейке A16 получим значение первого корня: $-0,92$.

5. Выполняя последовательно операции аналогичные предыдущим, вычислим значения остальных корней: $-0,209991$ и $0,720002$.

B18		fx = A18^3-0,01*A18^2-0,7044*A18+0,139104				
	A	B	C	D	E	
1	Вычисление корней функции					
2	x	Y				
3	-1	-0,1665				
4	-0,8	0,184224	первое приближение			
5	-0,6	0,342144				
6	-0,4	0,355264				
7	-0,2	0,271584				
8	0	0,139104				
9	0,2	0,005824	второе приближение			
10	0,4	-0,08026				
11	0,6	-0,07114	третье приближение			
12	0,8	0,081184				
13	1	0,424704				
14						
15	Точное решение					
16	-0,92	-2,6E-09				
17	0,209999	3,98E-07				
18	0,72	-9,7E-10				
19						
20						

Подбор параметра ✕

Установить в ячейке: fx

Значение:

Изменяя значение ячейки: fx

1.3.5. Решение уравнений

В предыдущем параграфе рассмотрена технология вычисления корней функции одной переменной. Предположим, что требуется решить уравнение

$$x^2 - 4 = 0,$$

т.е. найти такие значения x , при которых левая часть выражения, представленная полиномом второй степени, обращается в ноль. Представим уравнение в виде функциональной зависимости $y = x^2 - 4$.

Нетрудно догадаться, что решениями уравнения будут корни полученной функции.

1.3.6. Численное вычисление производной функции одного переменного

Известно, что численными приближенными методами производная функции в заданной точке может быть вычислена с использованием формулы конечных разностей. Выражение для вычисления произ-

водной функции одной переменной в точке x_k , записанное в конечных разностях, имеет вид

$$F'(x_k) = \frac{\Delta F}{\Delta x} = \frac{F(x_{k+1}) - F(x_k)}{x_{k+1} - x_k},$$

где Δx — очень малая конечная величина.

При достаточно малых значениях Δx , можно с приемлемой точностью получить величину производной функции в точке. Для вычисления производной в MS Excel будем использовать приведенную выше формулу. Рассмотрим технологию вычисления производной на примере.

Пример 1.18. Найти производную функции $y = 2x^3 + x^2$ в точке $x = 3$. Заметим, что производная приведенной функции в точке $x = 3$, вычисленная аналитическим методом, равна 60 — это значение нам понадобится для проверки результата, полученного путем вычисления численным методом.

Задачу вычисления производной в табличном процессоре можно решать двумя способами.

Решение первым способом

Введем в ячейку рабочего листа формулу правой части заданной функциональной зависимости, например в ячейку B2, как показано на рисунке, делая ссылку на ячейку, где будет находиться значение x , например A2,

$$=2*A2^3+A2^2.$$

Зададим окрестность точки $x = 3$ достаточно малого размера, например значение слева $x_k = 2,99999999$, а значение справа $x_{k+1} = 3,00000001$, и введем эти значения в ячейку A2 и A3 соответственно.

В ячейку C2 введем формулу вычисления производной

$$=(B3-B2)/(A3-A2).$$

В результате вычисления в ячейке C2 будет выведено приближенное значение производной заданной функции в точке $x = 3$, величина которой равна 60, что соответствует результату, полученному аналитически.

C2 ▼ fx =(B3-B2)/(A3-A2)			
	A	B	C
1	X	Y	Производная
2	2,999999999999	62,99999999940	60,000
3	3,0000000000010	63,00000000006	

Решение вторым способом

Введем в ячейку рабочего листа A2 заданное значение аргумента, равное 3, в ячейке B2 укажем достаточно малое приращение аргумента – (1E – 9), в ячейку C2 введем формулу для вычисления производной

$$=(2*(A2+B2)^3+(A2+B2)^2-(2*A2^3+A2^2))/B2.$$

После нажатия клавиши <Enter> получим результат вычисления 60,0000.

Как видим, результат получен такой же, как и при первом способе. Приведенный второй способ является более предпочтительным в случаях, когда нужно построить таблицу значений производной функции для заданных значений аргумента.

1.3.7. Вычисление локальных экстремумов функции

Напомним, что функция $Y=f(x)$ имеет экстремум при значении $x = x_k$, если производная функции в этой точке равна нулю.

Если функция $f(x)$ непрерывна на отрезке $[a, b]$ и имеет внутри этого отрезка локальный экстремум, то его можно найти, используя надстройку Excel **Поиск решения**.

Рассмотрим последовательность нахождения экстремума функции на примере.

Пример 1.19. Задана неразрывная функция $y = x^2 + x + 2$. Требуется найти ее экстремум (минимальное значение) на отрезке $[-2; 2]$.

Решение

В ячейку A3 рабочего листа введем любое число, принадлежащее заданному отрезку, в этой ячейке будет находиться значение x .

В ячейку B3 введем формулу, определяющую заданную функциональную зависимость. Вместо переменной x в этой формуле должна быть ссылка на ячейку A3

$$=A3^2+A3+2.$$

Выполним команду меню **Сервис/Поиск решения**.

В открывшемся окне диалога **Поиск решения** в поле *Установить целевую ячейку* укажем адрес ячейки, содержащей формулу (B3), установим переключатель *Минимальному значению*, в поле *Изменяя ячейки* укажем адрес ячейки, в которой содержится переменная x – A3.

Добавим два ограничения в соответствующее поле: $A3 \geq -2$ и $A3 \leq 2$.

	B3		$f_x = A3^2 + A3 + 2$					
	A	B	C	D	E	F	G	H
1	Поиск экстремума функции							
2	X	Y						
3	-0,5	1,75						
4	<div style="border: 1px solid black; padding: 5px;"> <p>Поиск решения ✕</p> <p>Установить целевую ячейку: <input type="text" value="\$B\$3"/> Выполнить</p> <p>Равной: <input type="radio"/> максимальному значению <input checked="" type="radio"/> значению: <input type="text" value="0"/> Закреть</p> <p><input checked="" type="radio"/> минимальному значению</p> <p>Изменяя ячейки: <input type="text" value="\$A\$3"/> Предположить</p> <p>Ограничения: <input type="text" value="\$A\$3 <= 2"/> Добавить</p> <p><input type="text" value="\$A\$3 >= -2"/> Изменить</p> <p style="text-align: right;">Удалить</p> <div style="text-align: right; margin-top: 10px;"> Параметры Восстановить Справка </div> </div>							
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								

	C2		$f_x = (B3 - B2) / (A3 - A2)$
	A	B	C
1	x	y	y'
2	-0,4999999999	1,7500000000	0,0000000000
3	-0,5000000001	1,7500000000	
4			

Щелчком на кнопке *Параметры* и в открывшемся диалоговом окне **Параметры поиска** решения установим относительную погрешность вычислений и предельное число итераций.

Щелчком на кнопке *Выполнить*. В ячейке A3 будет вычислено значение аргумента x функции, при котором она принимает минимальное значение, а в ячейке B3 — минимальное значение функции.

В результате выполнения вычислений в ячейке A3 будет получено значение независимой переменной, при котором функция принимает наименьшее значение $-0,5$, а в ячейке B3 — минимальное значение функции, равное $1,75$.

Построим график заданной функции и убедимся, что решение уравнения найдено верно.

Примечание. В частном случае при нахождении локального экстремума с использованием рассмотренной технологии, можно получить значение, которое не является экстремумом, а просто являет-

ся минимумом или максимумом функции в заданном диапазоне изменения аргумента.

Поэтому необходима дополнительная проверка, т.е. вычисление производной функции в найденной точке.

Используя приведенную технологию численного вычисления производной функции в заданной точке, проверим, является ли найденная точка $x = -0,5$ точкой экстремума функции $y = x^2 + x + 2$. Решение приведено на рисунке.

Как видно, производная в найденной точке равна нулю, следовательно, найденное значение функции является ее экстремальным значением.

Пример 1.20. Требуется найти значения аргумента в диапазоне $[-1; 1]$, при которых функция $y = x^2 + x + 2$ имеет экстремумы.

Решение

Табулируем заданную функцию с шагом 0,2.

Применяя второй из приведенных способов вычисления производной, вычислим значения функции $y = f(x + dx)$.

Вычислим значения производной при каждом табличном значении аргумента.

Анализируя полученные значения производных функции в точках, находим, что производная меняет знак в интервале значений аргумента $(-0,6; -0,4)$, следовательно, на этом интервале есть точка экстремума. Кроме того, заметим, что знак производной меняется с минуса на плюс, следовательно, точка экстремума является минимумом функции.

Применяя инструмент **Подбор параметра** или **Поиск решения** для решения уравнения $Y'(x) = 0$ относительно x , вычислим точное значение аргумента, при котором исходная функция принимает экстремальное значение $(-0,5)$.



Полученное значение производной исследуемой функции в точке $x = -0,5$ равно нулю, следовательно, в этой точке функция имеет экстремум.

1.3.8. Технология получения математической модели функции по ее табличному представлению

На практике часто бывает необходимо получить аналитическую формулу для функциональной зависимости, полученной экспериментально, модель которой представлена в виде таблицы. Это бывает необходимо в тех случаях, когда нужно найти значение функции в тех точках внутри данного интервала, где она таблично не задана — задача интерполяции, либо вычислить значение функции в точках за пределами заданного интервала — задача экстраполяции.

Решение задач интерполяции и экстраполяции обеспечивается построением интерполяционной или аппроксимирующей функции $L(x)$, приближенно заменяющей исходную $f(x)$, заданную таблично. Подбор аналитической формулы сводится к вычислению входящих в нее параметров таким образом, чтобы из всех функций такого вида выбрать ту, которая наилучшим образом описывает зависимость между изучаемыми величинами.

При построении аппроксимирующей функции должны быть решены следующие вопросы:

- выбор типа аппроксимирующей функции $L(x)$;
- оценка погрешности аппроксимации.

Подбираемая эмпирическая функция в зависимости от характера экспериментальных данных может быть следующих видов:

— линейная ($y = ax + b$) обычно применяется в тех случаях, когда экспериментальные данные изменяются относительно постоянно;

— полиномиальная ($y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$) — используется для описания экспериментальных данных, попеременно возрастающих и убывающих;

— логарифмическая ($y = a \ln x + b$), где a и b — константы, — применяется для описания экспериментальных данных, которые первоначально быстро возрастают или убывают, а затем постепенно стабилизируются;

— степенная ($y = bx^a$), где a и b — константы, — используется для аппроксимации экспериментальных данных, скорость изменения которых постоянно увеличивается или уменьшается;

— экспоненциальная ($y = be^{ax}$), где a и b — константы, — применяется для описания экспериментальных данных, которые быстро возрастают или убывают, а затем стабилизируются.

Близость исходной и аппроксимирующей функций определяется критерием аппроксимации. Наиболее распространен квадратичный критерий (R^2), именно такой критерий оценки применяется в табличном процессоре. Не вдаваясь в подробности вычисления этого критерия, скажем: чем ближе значение этого критерия к единице, тем ближе аппроксимирующая функция к исходной, тем точнее полученная аналитическая модель отражает функциональную зависимость.

Чтобы получить аппроксимирующую формулу в табличном процессоре, нужно выполнить следующую последовательность действий:

- построить по имеющейся таблице экспериментальных данных графическую модель функции;

- установить указатель мыши на линию графика (ряд) и вызвать контекстное меню;

- в контекстном меню выбрать *Добавить линию тренда* — откроется диалоговое окно **Линия тренда**;

- в диалоговом окне **Линия тренда** на вкладке *Тип*, учитывая характер изменения функции на графике, выбрать вид аппроксимирующей функции, а на вкладке *Параметры* задать дополнительные параметры, в том числе установить флажок *Показывать уравнение на диаграмме*.

После выполнения приведенных операций на диаграмме будет отображена линия тренда (графическое отображение аппроксимирующей функции), а также аппроксимирующая функция в аналитическом виде.

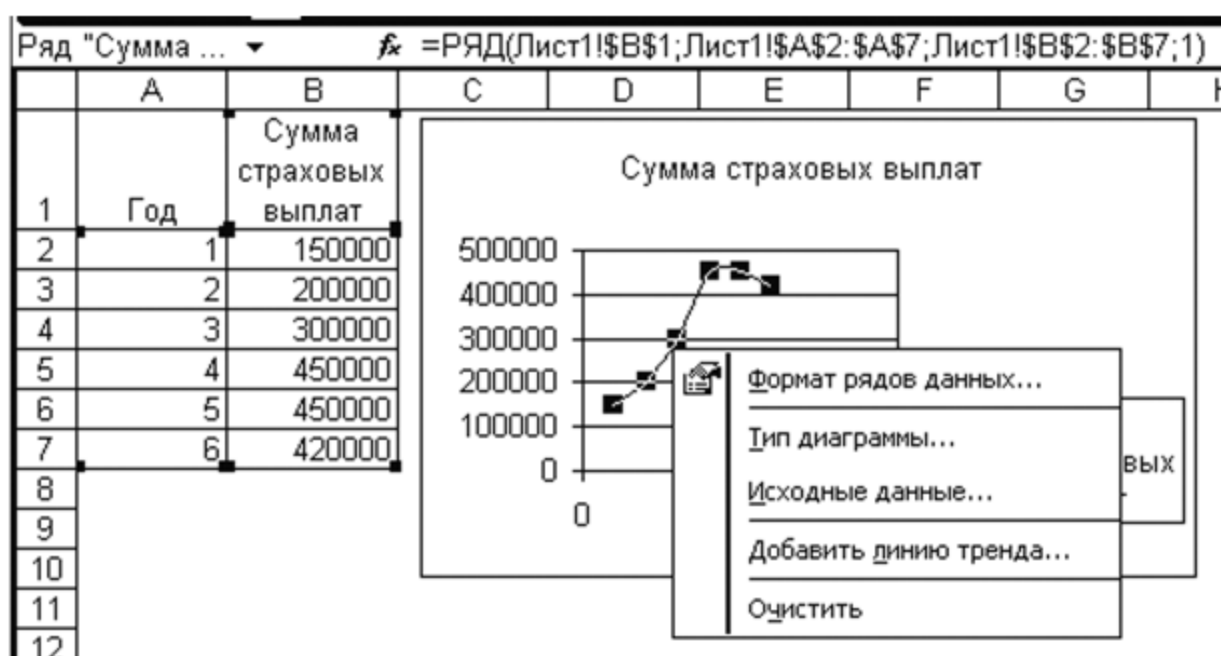
Рассмотрим процесс получения аппроксимирующей функции на примере.

Пример 1.21. Имеются сведения о величинах страховых выплат по годам, представленные в таблице. Требуется исследовать характер изменения величины страховых выплат и подобрать аппроксимирующую функцию.

Год	Сумма страховых выплат (руб.)
1	150 000
2	200 000
3	300 000
4	450 000
5	450 000
6	420 000

Решение

В ячейки рабочего листа введем исходные данные в виде таблицы.

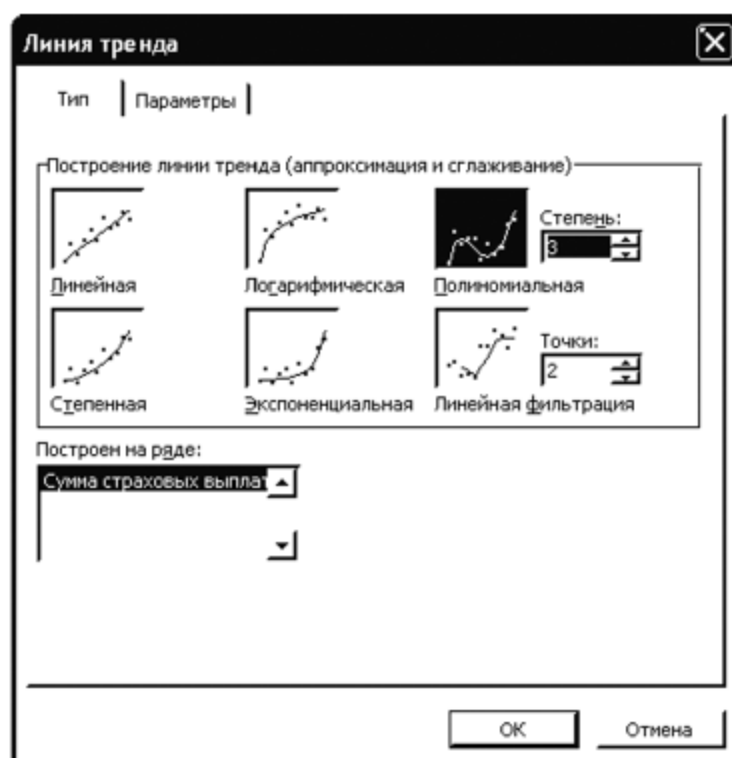


По данным таблицы, полученным экспериментально, построим график. На графике видно, что экспериментальные данные вначале растут, а затем убывают и есть только один экстремум. Следовательно, в качестве аппроксимирующей функции следует выбрать полиномиальную функцию третьей степени.

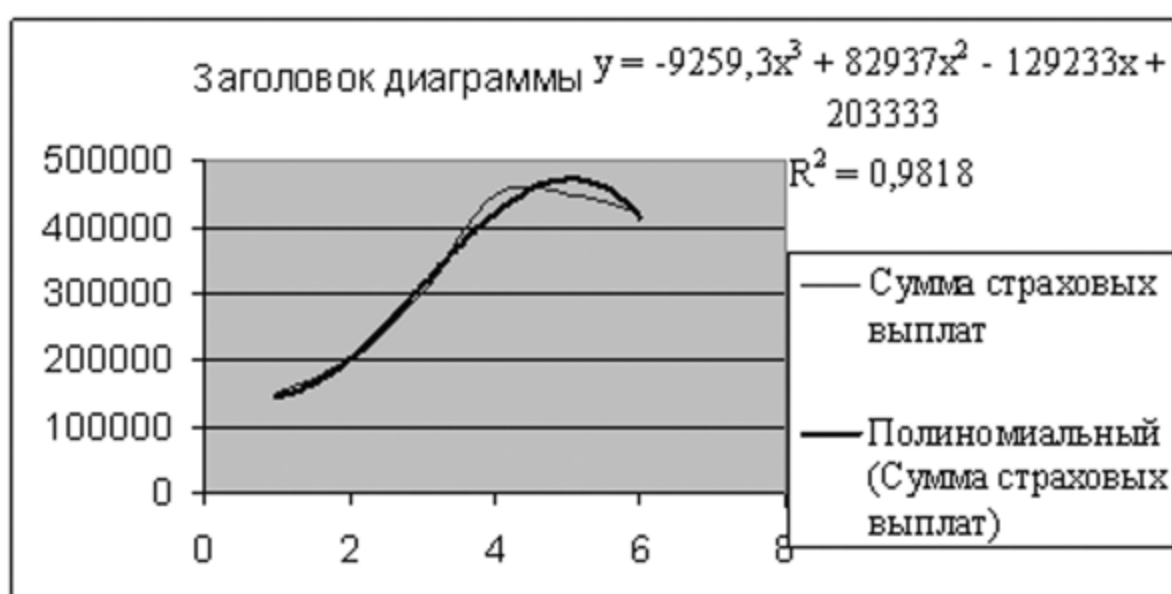
Откроем контекстное меню и выберем пункт *Добавить линию тренда*.

В диалоговом окне **Линия тренда** на вкладке *Тип* выберем *Полиномиальная* и установим *Степень*, равную трем. На вкладке *Параметры* установим флажки *Показывать уравнение на диаграмме* и *Поместить на диаграмму величину достоверности аппроксимации R^2* .

После щелчка на кнопке ОК на диаграмме отобразится линия тренда и аналитическая запись аппроксимирующей функции



$$y = -9259,3x^3 + 6E+07x^2 - 1E + 11x + 7E + 13, R^2 = 0,9818.$$



Величина R^2 близка к единице, следовательно, степень достоверности аппроксимации является высокой.

1.4. ТЕХНОЛОГИИ РЕШЕНИЯ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

1.4.1. Приближенное решение систем нелинейных уравнений

Для того чтобы графически решить систему из двух уравнений с двумя неизвестными, представленными в виде

$$\begin{cases} f_1(x, y) = 0 \\ f_2(x, y) = 0, \end{cases}$$

нужно выполнить следующие действия.

Привести уравнения системы к виду $y = \varphi(x)$:

$$\begin{cases} y = f_1(x) \\ y = f_2(x). \end{cases}$$

Создать последовательность значений аргумента x в заданном диапазоне.

Рассчитать значения функций для каждого значения аргумента (табулировать функции).

По полученным табличным значениям построить графики функций $f_1(x)$ и $f_2(x)$.

Найти точку пересечения построенных графиков, навести указатель мыши на точку пересечения — появится всплывающая надпись

с указанием искоемых (приближенных) значений координат точки пересечения.

Заметим, что точность вычисления корней при графическом методе решения определяется величиной шага последовательности x . Если, например, шаг равен 0,2, то абсолютная погрешность вычисления равна $\pm 0,2$.

Технологию графического решения систем уравнений рассмотрим на примере.

Пример 1.22. Найти графически приближенное решение системы

$$\begin{cases} \ln x_1 - x_2 = 0 \\ 1 - 2x_1 - x_2 = 0 \end{cases} \text{ в диапазоне } x \in [0, 2; 3] \text{ с шагом } 0,2.$$

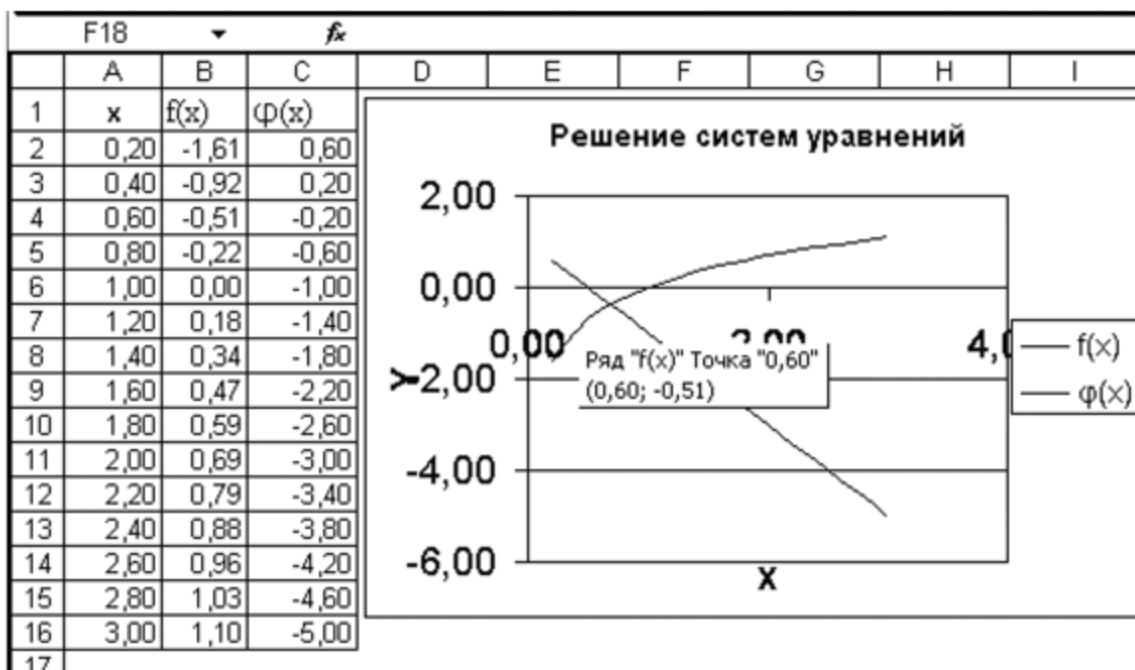
Решение

Представим приведенные уравнения в виде системы функций

$$\begin{cases} y = f_1(x) = 1 - 2x \\ y = f_2(x) = \ln x. \end{cases}$$

Табулируем функции $f_1(x)$ и $f_2(x)$ в заданном диапазоне x_1 и с указанным шагом.

По данным полученной таблицы построим график.



Подведем указатель мыши к узловой точке, которая расположена ближе других к точке пересечения графиков функций — отобразятся координаты точки пересечения с точностью, определяемой шагом табуляции (0,6; -0,51). Таким образом, приближенное решение системы получено.

1.4.2. Уточнение решения систем нелинейных уравнений

Уточнение решения систем нелинейных уравнений выполняется после их приближенного решения, которое выполняется, как правило, графическим способом. Полученные приближенные решения уточняются путем последовательных приближений с использованием итерационных методов. При этом в качестве критерия точности может быть использована величина $\varepsilon = f(x, y) - 0$, т.е. величина отклонения от нуля каждого уравнения системы, после подстановки в него искомым значений, полученных на очередном шаге итерации.

Для уточнения решения систем нелинейных уравнений в табличном процессоре можно применить одну из технологий, рассматриваемых в примерах 1.23 и 1.24.

Пример 1.23. Требуется решить систему нелинейных уравнений

$$\begin{cases} \ln x - y = 0 \\ 1 - 2x - y = 0 \end{cases} \text{ с точностью } |\varepsilon| < 0,001.$$

Решение

Графическим методом найдем приближенное решение приведенной системы. В результате приближенного решения получим $x = 0,6$; $y = -0,51$.

Подставим полученные значения неизвестных в исходные уравнения, получим: $\varepsilon_1 = \ln(0,6) + 0,51 = -0,00083$, $\varepsilon_2 = 1 - 1,2 + 0,51 = 0,31$. Таким образом, найденные решения удовлетворяют требуемой точности только для первого уравнения системы, а для второго нет. Следовательно, необходимо уточнить полученное решение.

Полученные значения x и y будем использовать в качестве начальных приближений для последующего более точного решения. В ячейки рабочего листа, например А3 и В3, введем полученные значения x и y .

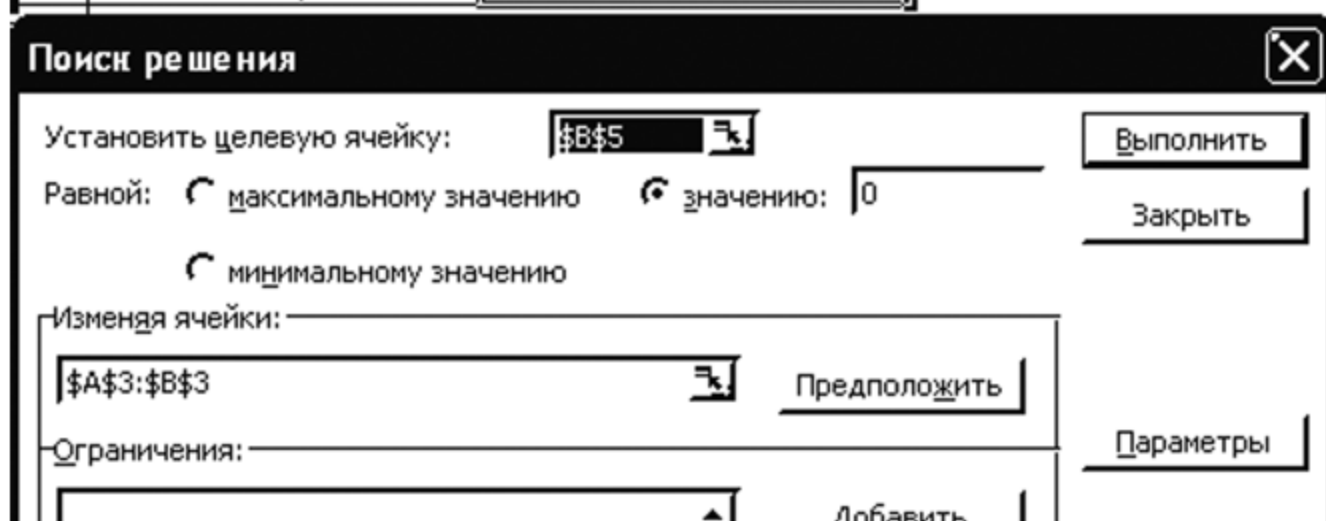
В ячейки А5 и В5 соответственно введем формулы, вычисляющие значения функций «=LN(A3) - B3» и «=1 - 2*A3 - B3».

	A5	▼	f _x = LN(A3)-B3
	A		B
1	Решение системы нелинейных уравнений		
2	x		y
3	0,687516997		-0,375033993
4	f₁(x)=ln(x)-y		f₂(x)=1-2x-y
5	0,000365266		0
6			

Используя надстройку табличного процессора **Поиск решения**, выполним несколько последовательных циклов итерационного решения каждого из уравнений относительно x и y , решая поочередно каждое из уравнений до тех пор, пока не будет обеспечена заданная точность.

После четырех циклов последовательного приближения получим решение системы, удовлетворяющее заданным требованиям точности. Точное решение исходной системы: $x = 0,687516997$, $y = -0,375033993$.

B5		$f_x = 1-2*A3-B3$
	A	B
1	Решение системы нелинейных уравнений	
2	x	y
3	0,687436003	-0,374872007
4	$f_1(x)=\ln(x)-y$	$f_2(x)=1-2x-y$
5	8,54665E-05	0



Пример 1.24. Дана система двух уравнений $\begin{cases} x^2 + y^2 = 3 \\ 2x + 3y = 1. \end{cases}$

Требуется найти решение приведенной системы в диапазоне значений x и y $[-3; 3]$.

Решение

Приведем систему к одному уравнению. Пара (x, y) является решением системы тогда и только тогда, когда она является решением следующего уравнения с двумя неизвестными:

$$(x^2 + y^2 - 3)^2 + (2x + 3y - 1)^2 = 0.$$

Для решения последнего уравнения необходимо найти начальные приближения, для этого табулируем выражение, стоящее в левой

части, как функцию двух переменных x и y . Для табуляции функции выполним следующие действия:

- в столбец А введем последовательность значений X с шагом 0,5, а строку 3 – последовательность значений Y также с шагом 0,5;

- присвоим диапазонам значений X и Y имена X и Y соответственно;

- выделим диапазон ячеек, в котором будут вычисляться значения функции (B4:N16);

- в выделенный диапазон введем формулу

$$=(X^2+Y^2-3)^2+(2*X+3*Y-1)^2;$$

- нажав комбинацию клавиш <Ctrl> + <Shift> + <Enter> выполним операцию над выделенным массивом. В выделенном диапазоне появятся вычисленные значения функции.

Найдем начальные приближения. Поскольку табулируемая функция задает поверхность, то начальные приближения следует искать во впадинах, т.е. в точках, где функция принимает наименьшие значения. На рисунке эти точки затемнены. Начальными приближениями являются пары $(-1; 1)$ и $(1,5; -0,5)$.

Введем значения найденных приближений в смежные ячейки рабочего листа. Над столбцами сделаем надписи XX и YY , которые будут выполнять в формулах роль меток. Обратите внимание, что мы уже использовали имена X и Y , поэтому имена новых меток должны отличаться.

В ячейку строки, в которой записана первая пара X и Y введем формулу, вычисляющую значение функции,

$$=(XX^2+YY^2-3)^2+(2*XX+3*YY-1)^2$$

и скопируем ее в следующую строку.

Установим курсор на ячейку, в которой записана формула и выполним команду меню **Сервис/Поиск решения**. Выполним настройку параметров инструмента **Поиск решения**: *Предельное число итераций* 1000, *относительная погрешность* 0,000001.

В окне **Поиск решения** в качестве целевой ячейки установим адрес ячейки, содержащей формулу, установим переключатель *Минимальному значению*, в поле *Изменяя ячейки* укажем адрес диапазона, содержащего начальные приближения. Щелкнем на ОК. В ячейках, где хранились начальные приближения, будет получена первая пара корней.

Повторим эти операции для получения второй пары приближений.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Решение системы нелинейных уравнений													
2		Y												
3	X	-3	-2,5	-2	-1,5	-1	-0,5	0	0,5	1	1,5	2	2,5	3
4	-3	481,00	360,31	269,00	200,31	149,00	111,31	85,00	69,31	65,00	74,31	101,00	150,31	229,00
5	-2,5	375,06	272,50	196,56	140,50	99,06	68,50	46,56	32,50	27,06	32,50	52,56	92,50	159,06
6	-2	296,00	208,81	146,00	100,81	68,00	43,81	26,00	13,81	8,00	10,81	26,00	58,81	116,00
7	-1,5	237,06	162,50	110,56	74,50	49,06	30,50	16,56	6,50	1,06	2,50	14,56	42,50	93,06
8	-1	193,00	128,31	85,00	56,31	37,00	23,31	13,00	5,31	1,00	2,31	13,00	38,31	85,00
9	-0,5	160,06	102,50	65,56	42,50	28,06	18,50	11,56	6,50	4,06	6,50	17,56	42,50	88,06
10	0	136,00	82,81	50,00	30,81	20,00	13,81	10,00	7,81	8,00	12,81	26,00	52,81	100,00
11	0,5	120,06	68,50	37,56	20,50	12,06	8,50	7,56	8,50	12,06	20,50	37,56	68,50	120,06
12	1	113,00	60,31	29,00	12,31	5,00	3,31	5,00	9,31	17,00	30,31	53,00	90,31	149,00
13	1,5	117,06	60,50	26,56	8,50	1,06	0,50	4,56	12,50	25,06	44,50	74,56	120,50	189,06
14	2	136,00	72,81	34,00	12,81	4,00	3,81	10,00	21,81	40,00	66,81	106,00	162,81	244,00
15	2,5	175,06	102,50	56,56	30,50	19,06	18,50	26,56	42,50	67,06	102,50	152,56	222,50	319,06
16	3	241,00	156,31	101,00	68,31	53,00	51,31	61,00	81,31	113,00	158,31	221,00	306,31	421,00
17														
18	Исходные приближения													
19	Первая пара	XX	YY	Формула										
20		-1	1	1,00										
21	Вторая пара	1,5	-0,5	0,50										
22														

1.5. ПРИМЕНЕНИЕ ТЕХНОЛОГИИ ИССЛЕДОВАНИЯ ФУНКЦИЙ ДЛЯ РЕШЕНИЯ ЭКОНОМИЧЕСКИХ ЗАДАЧ

1.5.1. Кривые спроса и предложения, точка равновесия

Рассмотренные технологии исследования функций полезны для решения некоторых экономических задач. Одной из таких задач является определение точки равновесия спроса и предложения.

Известно, что чем ниже цена (p), тем больше спрос (D) при постоянной покупательной способности населения. Обычно зависимость спроса от цены в графическом представлении имеет вид ниспадающей линии, чаще всего приближающейся к прямой. В свою очередь предложение растет с увеличением цены на товар и в графическом представлении имеет вид восходящей линии.

Если зависимость спроса от цены определяется функцией $D = f(p)$, а зависимость предложения от цены $S = \varphi(p)$, то условие равновесия определяется уравнением

$$f(p) = \varphi(p)$$

и соответствует точке пересечения кривых D и S . Цена P_0 , при которой выполняется это условие, называется равновесной.

Рассмотрим технологию решения задачи по определению точки равновесия на примере.

Пример 1.25. Зависимость спроса y на некоторый товар от цены x выражается уравнением $f_1(x) = 2/x - y + 2$, а зависимость предложе-

ния от цены — уравнением $f_2(x) = x^2 - y + 1$. Требуется, решив систему уравнений, найти точку равновесия в диапазоне $x \in [0,2; 3]$ с точностью 0,001.

Решение

Приведем исходные уравнения к системе следующего вида

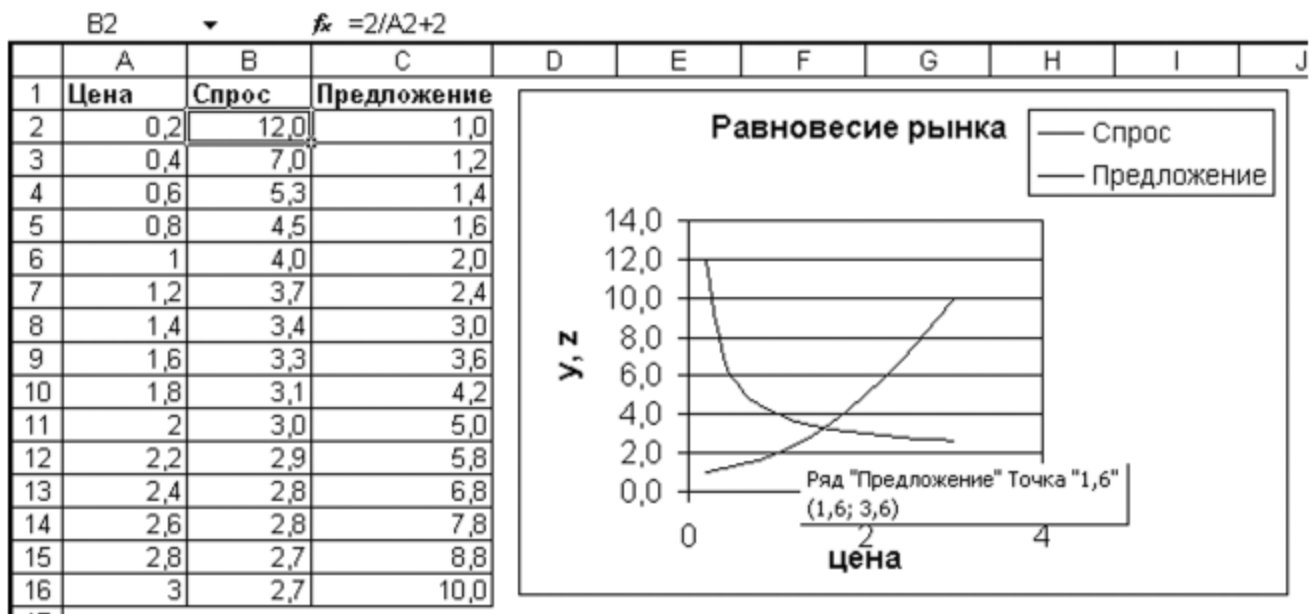
$$\begin{cases} y = f(x) = \frac{2}{x} + 2 \\ y = \phi(x) = x^2 + 1. \end{cases}$$

Создадим последовательность значений x с шагом 0,2.

Рассчитаем значения функций спроса $f(x)$ и предложения $\phi(x)$ для сформированной последовательности значений x .

По данным полученной таблицы построим графики функций.

5. Подведем указатель мыши к точке пересечения кривых — отобразятся приближенные координаты точки равновесия. В данном случае цена x в точке равновесия равна 1,6, предложение и спрос характеризуются величиной $f(x) = \phi(x) = 3,6$.



Применяя приведенную раньше технологию, уточним решение. Результат уточнения приведен на рисунке.

	A	B
1	цена - x	y
2	1,521683393	3,315520142
3	$f_1(x)$	$f_2(x)$
4	-0,00118629	2,06678E-07

Таким образом, равновесное значение цены составляет 1,521, а спрос и предложение находятся в равновесии и выражаются величиной $f(x) = \varphi(x) = 3,315$.

1.5.2. Технология построения и исследования паутиной модели рынка

Известно, что процесс достижения равновесной цены является динамическим и представляет собой форму торга, при котором продавец назначает свою цену P_1 . Покупатель оценивает спрос D_1 при этой цене и определяет свою цену P_2 , при которой этот спрос D_1 равен предложению. Цена P_2 ниже равновесной, так как покупатель стремится купить дешевле. В свою очередь продавец оценивает спрос D_2 , соответствующий цене P_2 , и назначает свою цену P_3 , эта цена выше равновесной — продавец стремится дороже продать. Этот процесс торга продолжается и при определенных условиях приближается к устойчивому приближению к равновесной цене. Этот процесс графически можно изобразить в виде спирали, которая, в зависимости от условий, может быть скручивающейся, и тогда компромиссная цена приближается к равновесной, раскручивающейся — в этом случае компромисс не будет найден, или спираль имеет замкнутую форму — в этом случае продавец и покупатель остаются каждый со своей первоначальной ценой. Приведенный процесс иллюстрируется на рис. 1.6.

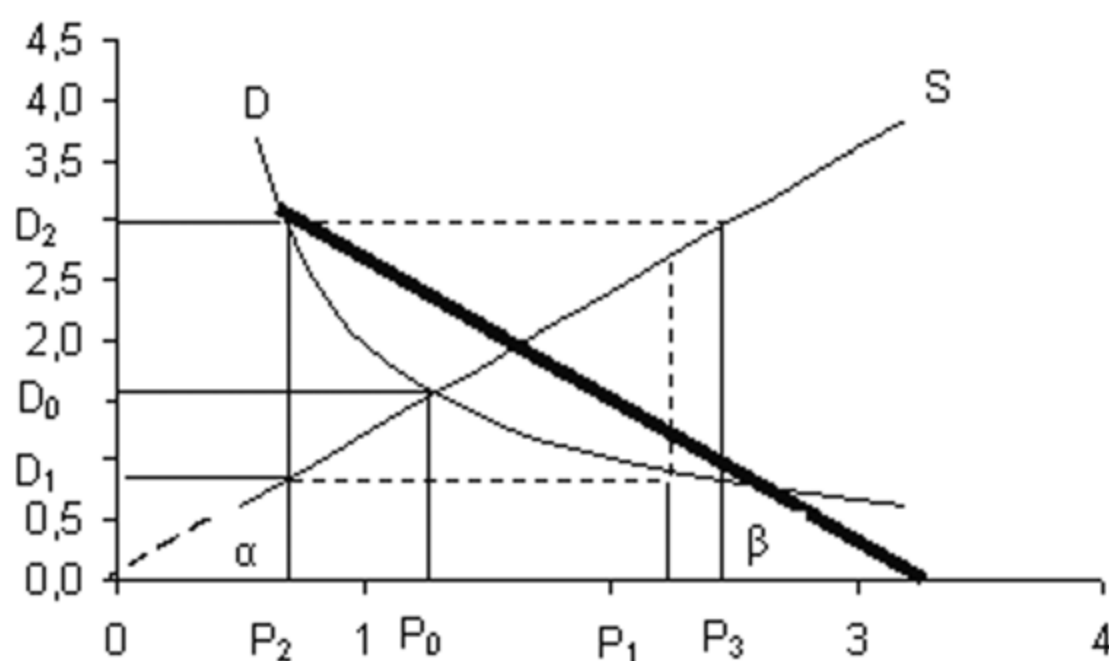


Рис. 1.6. Графическое изображение паутиной модели рынка

На приведенном рисунке процесс имеет вид раскручивающейся спирали. Из рисунка нетрудно заметить, что вид спирали зависит от соотношения величин углов наклона α и β сглаживающих прямых в пределах границ разумно допустимых цен. При $\alpha = \beta$ спираль имеет замкнутую форму, при $\alpha > \beta$ – спираль скручивается, в противном случае – спираль раскручивается. Величины углов наклона линий тренда определяются величинами коэффициентов α и β . Таким образом, задача определения вида процесса состоит в вычислении величин этих коэффициентов и последующего сравнения.

Пример 1.26. Торговая фирма по результатам реализации одного из видов товаров в течение нескольких периодов времени получила следующие экспериментальные данные:

№ периода	Цена (усл. ед.)	Спрос	Предложение
1	0,1	3,0	2,0
2	0,2	2,8	2,5
3	0,3	2,4	3,5
4	0,4	2,1	3,9
5	0,5	2,0	4,7
6	0,6	1,7	5,2

Требуется:

1. Определить является ли паутиная модель рынка скручивающейся.
2. Найти равновесную цену.

Решение

Введем данные таблицы в ячейки рабочего листа.

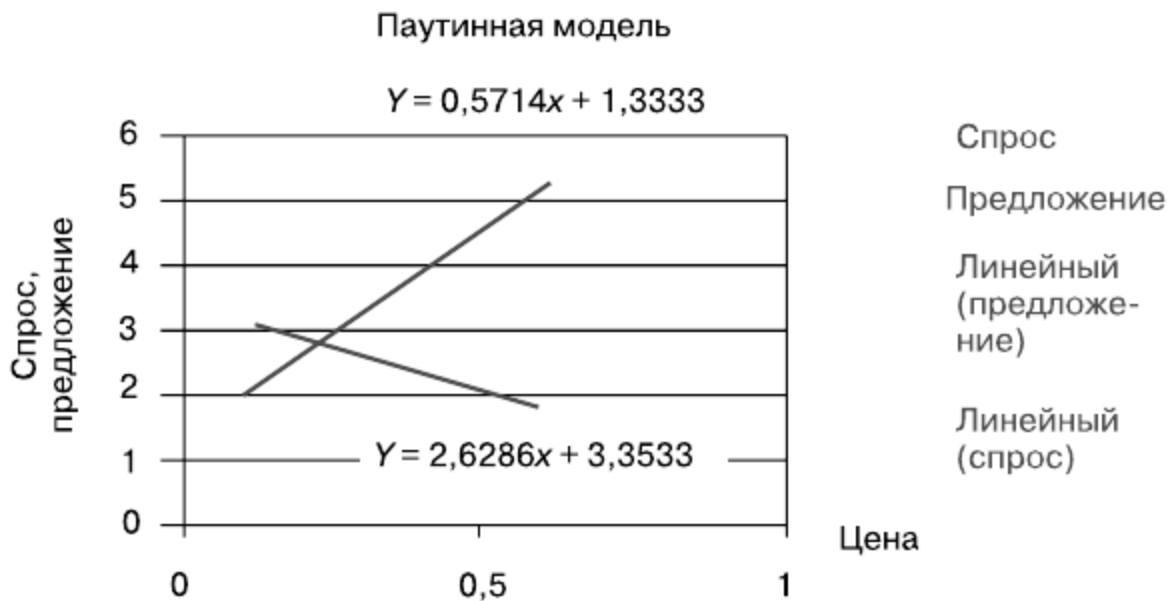
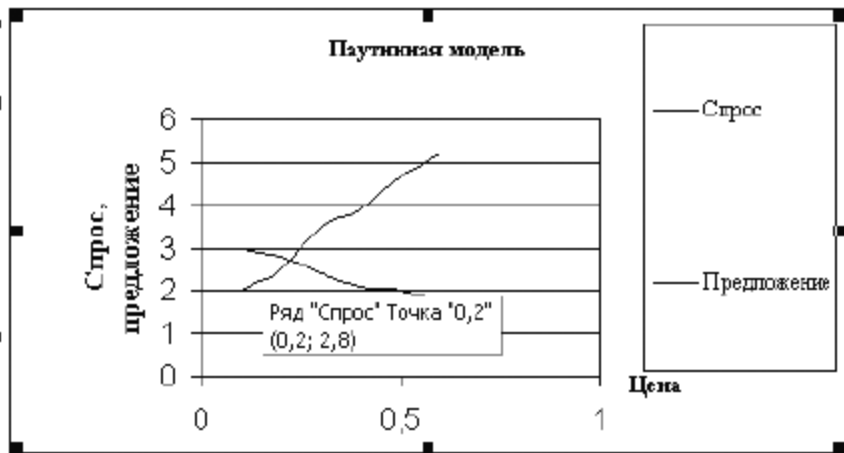
По табличным данным построим графики спроса и предложения.

Подведем указатель мыши к точке пересечения кривых спроса и предложения – отобразятся координаты точки пересечения (0,2; 2,8). Таким образом, равновесная цена равна 0,2 усл. ед.

Построим линии тренда для обеих кривых с линейной аппроксимацией и отображением уравнения на диаграмме. Полученное уравнение предложения имеет вид $y = 6,5714x + 1,3333$, а уравнение спроса: $y = -2,6286x + 3,2533$.

Величина коэффициента $\alpha = 2,6286$, а коэффициента $\beta = 6,5714$, т.е. $\beta > \alpha$. Следовательно, модель паутинового рынка раскручивается.

№ периода	Цена (усл. ед.)	Спрос	Предложение
1	0,1	3	2
2	0,2	2,8	2,5
3	0,3	2,4	3,5
4	0,4	2,1	3,9
5	0,5	2	4,7
6	0,6	1,7	5,2



1.5.3. Вычисление предельных экономических показателей

В экономических задачах одни экономические показатели являются функциями каких-либо других показателей или величин. Иначе говоря, существует зависимость одних показателей от других: $y = f(x)$.

Так, например, себестоимость продукции зависит от производимого объема $C = f(Q)$, издержки производства зависят от количества выпускаемой продукции, величина созданного общественного продукта зависит от совокупных затрат живого труда и суммарного объема применяемых производственных фондов и т.п.

Предельные экономические показатели характеризуют величину прироста величины функции Δy от прироста ее аргумента Δx :

$$PY = \Delta y : \Delta x.$$

Так, например, предельная себестоимость характеризует себестоимость ΔC прироста продукции ΔQ :

$$PC = \Delta C : \Delta Q.$$

Если зависимость Δy от Δx непрерывна, то приведенное разностное уравнение можно заменить производной $ПУ = f'(x)$.

Пример 1.27. Зависимость издержек производства от объема выпускаемой продукции в денежных единицах выражается формулой $C = 20Q - 0,05Q^3$.

Требуется определить предельные издержки производства при объеме выпускаемой продукции 10 ден. ед.

Решение

Предполагая, что в ячейке А2 рабочего листа будет записано значение Q_k — левая граница окрестности точки $Q = 10$, в ячейку В2 введем формулу

$$=20*A2-0,05*A2^3.$$

Скопируем введенную формулу в ячейку В3.

В ячейку С3 введем формулу вычисления производной

$$=(B3-B2)/(A3-A2).$$

В ячейки А2 и А3 введем значения Q для левой и правой окрестности точки $Q = 10$.

	А	В	С
	Q	С	Предельные издержки
1			
2	9,999999	=20*A2-0,05*A2^3	=(B3-B2)/(A3-A2)
3	10,000001	=20*A3-0,05*A3^3	
4			

После выполнения приведенных выше операций в ячейке С2 будет получен результат, отображенный на рисунке.

	А	В	С
	Q	С	Предельные издержки
1			
2	9,999999	149,999995	4,999999991
3	10,000001	150,000005	
4			

Таким образом, предельные издержки производства при объеме выпускаемой продукции 10 ден. ед. составляют примерно 4,99999 ден. ед.

1.5.4. Вычисление эластичности экономических показателей

Под эластичностью экономического показателя понимается процентное изменение величины значения величины функции, определяющей зависимость одного показателя от другого, при изменении ее

аргумента на 1%. Так, например, при анализе и прогнозах ценовой политики применяется понятие эластичности спроса.

Пусть $y = f(x)$ – функциональная зависимость показателя y от другого – x . Тогда эластичность y от x определится выражением

$$E_y = \frac{\Delta y : y \times 100\%}{\Delta x : x \times 100\%}.$$

При достаточно малых значениях $\Delta x \Rightarrow 0$ эластичность можно определить как

$$E_y = x \frac{Y'(x)}{Y(x)}.$$

Рассмотрим технологию вычисления эластичности экономических показателей на примере.

Пример 1.28. Спрос на товар определяется функциональной зависимостью $D(P) = 100 - 3P$.

Требуется определить эластичность спроса при цене на товар $P = 20$ ден. ед.

Решение

Конечно, в приведенной задаче функция спроса от цены задана простым выражением и эластичность можно вычислить аналитически, но мы, чтобы продемонстрировать технологию, решим задачу в электронной таблице. Выполним следующие действия.

В рабочие ячейки A3 и A4 введем границы окрестности точки $P = 20$.

В ячейки B3 и B4 введем формулы, вычисляющие значения функции $D(P)$.

В ячейку C3 введем формулу для вычисления производной функции.

В ячейку D3 введем формулу, вычисляющую эластичность.

D12 ▼ f*				
	A	B	C	D
1	Модель вычисления эластичности спроса			
2	P	F(P)	F'(P)	E
3	19,99999999	=10-3*A3	=(B4-B3)/(A4-A3)	=20*C3/(100-3*20)
4	20,00000001	=10-3*A4		
5				
D3 ▼ f* =20*C3/(100-3*20)				
	A	B	C	D
1	Модель вычисления эластичности спроса			
2	P	F(P)	F'(P)	E
3	19,99999999	-49,999999970	-3	-1,5
4	20,00000010	-50,00000030		
5				

В результате вычисления в ячейке D3 будет получен результат. Таким образом, эластичность спроса для заданной функции при цене на товар 20 ден. ед. равна 1,5.

1.6. ТЕХНОЛОГИЯ ЧИСЛЕННОГО ВЫЧИСЛЕНИЯ ОПРЕДЕЛЕННОГО ИНТЕГРАЛА

Интегральное исчисление в экономике применяется для решения таких задач, как, например, вычисление издержек по заданной функции предельных издержек, для нахождения дисконтированной стоимости денежного потока и др.

Для численного вычисления определенного интеграла с использованием конечных разностей существует несколько методов. Наиболее простым является метод трапеций, для которого справедлива формула

$$\int_k^n F(x) = \sum_{i=k}^n F(x_i)(x_{i+1} - x_i) + \frac{(Fx_{i+1} - F(x_i))(x_{i+1} - x_i)}{2}.$$

Как можно заметить, абсолютная погрешность вычисленного значения зависит от величины задаваемого конечного приращения аргумента Δx . Чем меньше это значение, тем точнее результат.

При вычислении величины определенного интеграла в табличном процессоре можно использовать две технологии: приближенного вычисления и точного вычисления.

1.6.1. Технология приближенного вычисления

Технология приближенного вычисления определенного интеграла в табличном процессоре основана на построении табличных значений подынтегрального выражения для каждого из значений аргумента функции, задаваемых на отрезке, определяемом пределами интегрирования с достаточно малым шагом. Используя его, можно получить лишь приближенное значение интеграла.

Технологию приближенного вычисления определенного интеграла в табличном процессоре с использованием формулы трапеций рассмотрим на примере.

Пример 1.29. Требуется вычислить определенный интеграл $\int_k^n 2x dx$.

Величина интеграла, вычисленная аналитически, равна 9.

Решение

Табулируем подинтегральную функцию в диапазоне изменения значений аргумента 0–3 с шагом 0,2.

В ячейку С2 введем формулу $= (A3-A2)*B2 + (A3-A2)*(B3-B2)/2$, которая реализует часть размещенной правее знака суммы приведенной выше формулы трапеций, т.е. вычисляет величину элементарной площадки (криволинейной трапеции).

Скопируем буксировкой формулу, записанную в ячейке С2, до значения аргумента $x = 2,8$.

В ячейке С17 просуммируем с помощью автосуммирования полученные результаты. Вычисленное значение в ячейке С17 и будет величиной интеграла – 9.

C2		fx = B2*(A3-A2)+(B3-B2)*(A3-A2)/2				
	A	B	C	D	E	F
	x	y	частные суммы			
1						
2	0	0	0,04			
3	0,2	0,4	0,12			
4	0,4	0,8	0,2			
5	0,6	1,2	0,28			
6	0,8	1,6	0,36			
7	1	2	0,44			
8	1,2	2,4	0,52			
9	1,4	2,8	0,6			
10	1,6	3,2	0,68			
11	1,8	3,6	0,76			
12	2	4	0,84			
13	2,2	4,4	0,92			
14	2,4	4,8	1			
15	2,6	5,2	1,08			
16	2,8	5,6	1,16			
17	3	6	9			

Результат
интегрирования

1.6.2. Технология точного вычисления

Технология точного вычисления основана на использовании аппарата циклических ссылок и итераций табличного процессора. Применение этой технологии позволяет задавать достаточно малый шаг интегрирования, что увеличивает точность вычислений. Для точного вычисления выполним следующие операции:

– определим, на сколько интервалов нужно разбить диапазон интегрирования, чтобы получить требуемую точность, и задать их коли-

чество в виде числа итераций. Положим, для решения нашей задачи достаточно 10 000 интервалов;

— выполним команду меню **Сервис/Параметры**, откроем закладку *Вычисления* в диалоговом окне **Параметры** и в поле *Предельное число итераций* введем число 10 000. Если установлен флажок *Итерации*, то выключим его. Закроем диалоговое окно **Параметры**;

— в ячейки рабочего листа введем исходные данные и формулы для вычислений (рис. 1.7).

	А	В	С	Д
1	Точное вычисление определенного интеграла			
2			х	Значение интеграла
3	Нижний предел	0	= 0+С3+В6	=D3+2*С3*В6+(2*(С3+В6)-2*С3)*В6/2
4	Верхний предел	3		
5	Число итераций	10000		
6	Шаг интегрирования	=(В4-В2)/В5		
7	Результаты решения			

	А	В	С	Д
1	Точное вычисление определенного интеграла			
2			х	Значение интеграла
3	Нижний предел	0	3	9,0018
4	Верхний предел	3		
5	Число итераций	10000		
6	Шаг интегрирования	0,0003		

Рис. 1.7. Пример реализации технологии точного вычисления определенного интеграла

В ячейке В6 формула $=(B4-B2)/B5$ вычисляет шаг интегрирования. В ячейке С3 формула $=0+C3+B6$ — вычисляет текущее значение аргумента x . Значение 0 в формуле устанавливает нижний предел интегрирования. В формуле есть циклическая ссылка на эту же ячейку $C3+B6$, она реализует накопление величины x относительно нижнего предела интегрирования.

В ячейке D3 записана формула, реализующая метод трапеций и накопление суммы площадей элементарных трапеций.

После ввода исходных данных и формул вновь выполним команду меню **Сервис/Параметры**, откроем закладку *Вычисления* в диалоговом окне **Параметры** и установим флажок *Итерации*. Щелкнем на кнопке ОК. Потребуется некоторое время для того, чтобы табличный процессор выполнил заданное количество циклов итераций и вычислил результат.

После завершения вычислений вновь вызовем диалоговое окно **Параметры** и выключим флажок *Предельное число итераций*.

1.7. ТЕХНОЛОГИИ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ

1.7.1. Постановка задачи

Основной целью экономики является рациональное функционирование хозяйствующих субъектов или, иначе говоря, оптимальная деятельность при ограниченных ресурсах. Поэтому в экономической области существует широкий класс задач оптимизации, или, как их еще называют, экстремальных задач.

В задачах оптимизации вычисляются значения параметров некоторой функции $y = f(x_1, x_2, \dots, x_n)$, при которых она принимает наилучшее значение (максимальное или минимальное). При этом предполагается, что на значения аргументов функции (x) наложены ограничения. Эту функцию называют целевой, а набор количественных значений между переменными, выражающих определенные требования к параметрам экономической задачи, в виде уравнений или неравенств называют системой ограничений. Совокупность соотношений, содержащих целевую функцию и ограничения на ее аргументы, называют математической моделью экономической задачи оптимизации.

Если целевая функция линейна и на ее аргументы наложены линейные ограничения, то такую задачу оптимизации называют задачей линейного программирования.

В общем виде математическая модель задачи может быть представлена в виде

$$f(x) = a_1x_1 + a_2x_2 + \dots + a_nx_n \rightarrow \max (\min)$$

при условии, что $Ax \leq b$, $lb \leq x \leq ub$,

где $f(x)$ — целевая функция;

x — аргументы функции;

a — коэффициенты при аргументах;

A — матрица коэффициентов;

b — вектор, содержащий значения ограничений;

lb и ub — вектора ограничений на значения аргументов целевой функции.

1.7.2. Технология оптимизации

Существуют различные методы решения задач линейного программирования. В табличном процессоре для этой цели предназначен инструмент **Поиск решения**, о котором речь шла ранее. В этом инструменте применен итерационный способ подбора параметров целевой функции. Применение этого инструмента позволяет решать задачи оптимизации с высокой точностью.

Технологическая последовательность решения задачи в табличном процессоре включает следующие шаги:

– на основе постановки задачи и уяснения ее экономической сути разрабатывается математическая модель, аналитически представляющая целевую функцию и функции ограничений;

– в электронную таблицу вводятся исходные данные и формулы, реализующие разработанную математическую модель;

– настраиваются параметры инструмента **Поиск решения**, после чего он применяется для решения задачи.

Рассмотрим технологию решения задачи линейного программирования на примере.

Пример 1.30. Оптимальный план выпуска продукции.

Фирма производит два вида мороженого: сливочное и шоколадное. Для изготовления мороженого используются два исходных продукта: молоко и наполнители, расходы которых на 1 кг готового продукта и их суточные запасы приведены в таблице.

Исходный продукт	Расход исходных продуктов на 1 кг мороженого (кг)		Запас на складе (кг)
	Сливочное	Шоколадное	
Молоко	0,8		0,5 400
Наполнители	0,4		0,8 365

Суточный спрос на сливочное мороженое превышает спрос на шоколадное не более чем на 100 кг. Кроме того, известно, что спрос на шоколадное мороженое не превышает 350 кг в сутки. Отпускная цена 1 кг сливочного мороженого — 16 ден. ед., шоколадного — 14 ден. ед. Требуется определить, в каком количестве мороженого каждого вида должна производить фирма, чтобы доход от реализации продукции был максимальным.

Решение

1. Разработка математической модели.

Введем обозначения: x_1 — суточный объем производства сливочного мороженого, x_2 — суточный объем производства шоколадного мороженого. Исходя из условия задачи, целевая функция будет иметь вид

$$L(\bar{x}) = 16x_1 + 14x_2 \rightarrow \max$$

при ограничениях

$$\begin{cases} 0,8x_1 + 0,5x_2 \leq 400 - \text{ограничение по запасу молока} \\ 0,4x_1 + 0,8x_2 \leq 365 - \text{ограничение по запасу наполнителей} \\ x_1 - x_2 \leq 100 - \text{ограничение по спросу на сливочное мороженое} \\ x_2 \leq 350 - \text{ограничение по спросу на шоколадное мороженое} \\ x_1 \geq 0 \\ x_2 \geq 0. \end{cases}$$

2. Формализация модели в электронной таблице.

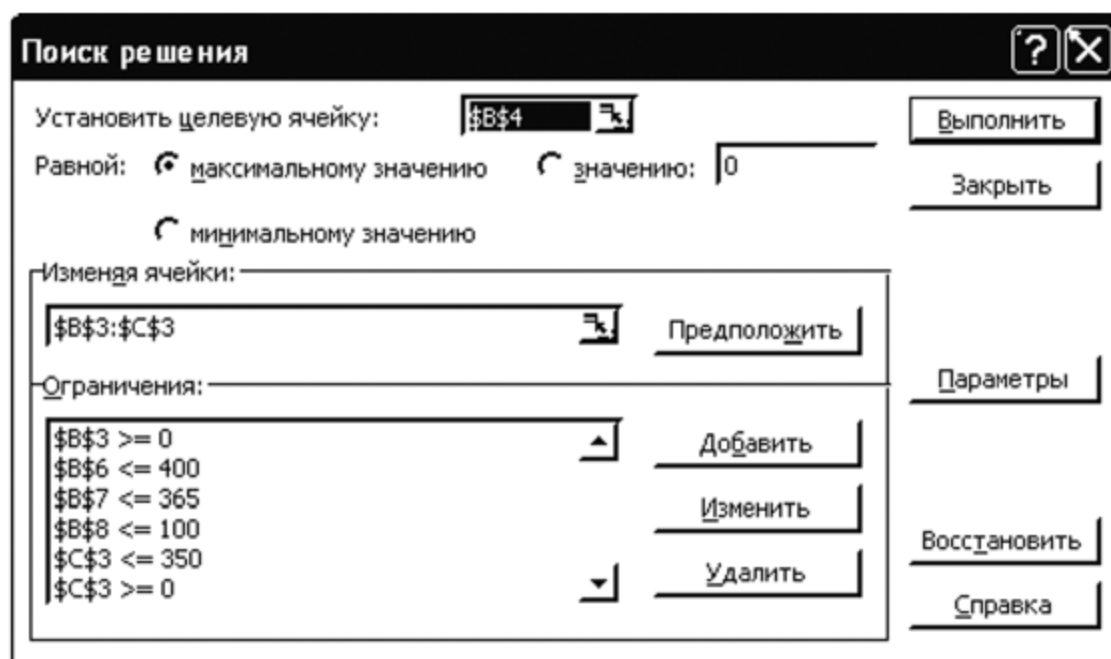
На рабочем листе сформируем табличку, как показано на рисунке.

C12		fx	
	A	B	C
1	Задача оптимизации плана выпуска продукции		
2	Параметры	X ₁	X ₂
3	Решение		
4	Целевая функция	=16*B3+14*C3	
5	Ограничения		
6	По молоку	=0,8*B3+0,5*C3	
7	По наполнителям	=0,4*B3+0,8*C3	
8	По спросу на сливочное	=B3-C3	
9			

Полагая, что аргументы целевой функции (x_1 и x_2) будут размещаться в ячейках B3 и C3, в ячейку B4 введем формулу целевой функции: =16*B3+14*C3, а в ячейки B6, B7 и B8 – формулы ограничений.

3. Настройка инструмента Поиск решения.

Выполним команду меню **Сервис/Поиск решения** – откроется диалоговое окно **Поиск решения**.



Укажем в диалоговом окне ссылки на целевую ячейку, на ячейки, в которых размещены параметры целевой функции, а также введем все ограничения.

После щелчка на кнопке ОК в ячейках В3 и С3 будет получено решение — значения x_1 и x_2 , при которых целевая функция имеет максимальное значение при заданных ограничениях.

A14 ▾ f_x			
	А	В	С
1	Задача оптимизации плана выпуска продукции		
2	Параметры	x_1	x_2
3	Решение	312,50000002	300
4	Целевая функция	9200,0000002	
5	Ограничения		
6	По молоку	400,00000001	
7	По наполнителям	365	
8	По спросу на сливочное	12,500000035	
9			
10			

Таким образом, при выпуске 312,5 кг сливочного и 300 кг шоколадного мороженого максимальный доход от реализации составит 9200 ден. ед.

1.7.3. Технология решения транспортной задачи линейного программирования

Целью транспортной задачи является планирование наиболее рациональных путей и способов транспортировки товаров. В сущности, технология решения транспортной задачи линейного программирования практически ничем не отличается от технологии решения других оптимизационных задач. Покажем это на примере.

Пример 1.31. На складах А1, А2, А3 имеются запасы товаров в количествах 90, 400 и 110 т соответственно. Грузополучатели В1, В2, В3 должны получить эти товары в количествах 130, 300, 160 т соответственно. Требуется найти такой вариант перевозки грузов, при котором сумма затрат на перевозки будет минимальной. Расходы по перевозке 1 т грузов в условных единицах приведены в таблице.

Грузополучатели	Склад А1	Склад А2	Склад А3
В1	2	5	2
В2	4	1	5
В3	3	6	8

Решение

Математическая модель

Введем следующие обозначения:

- x_{11} — количество товара, перевозимое грузополучателю В1 со склада А1;
 x_{12} — количество товара, перевозимое грузополучателю В1 со склада А2;
 x_{13} — количество товара, перевозимое грузополучателю В1 со склада А3;
 x_{21} — количество товара, перевозимое грузополучателю В2 со склада А1;
 x_{22} — количество товара, перевозимое грузополучателю В2 со склада А2;
 x_{23} — количество товара, перевозимое грузополучателю В2 со склада А3;
 x_{31} — количество товара, перевозимое грузополучателю В3 со склада А1;
 x_{32} — количество товара, перевозимое грузополучателю В3 со склада А2;
 x_{33} — количество товара, перевозимое грузополучателю В3 со склада А3.

Тогда целевая функция будет иметь вид

$$L = 2x_{11} + 5x_{12} + 2x_{13} + 4x_{21} + x_{22} + 5x_{23} + 3x_{31} + 6x_{32} + 8x_{33} \rightarrow \min$$

При ограничениях:

$$\left\{ \begin{array}{l} x_{11} + x_{12} + x_{13} = 140 - \text{груз, привезенный грузополучателю В1} \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{со всех складов} \\ x_{21} + x_{22} + x_{23} = 300 - \text{груз, привезенный грузополучателю В2} \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{со всех складов} \\ x_{31} + x_{32} + x_{33} = 160 - \text{груз, привезенный грузополучателю В3} \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{со всех складов} \\ x_{11} + x_{21} + x_{31} = 90 - \text{груз, отгруженный со склада А1} \\ x_{12} + x_{22} + x_{32} = 400 - \text{груз, отгруженный со склада А2} \\ x_{13} + x_{23} + x_{33} = 90 - \text{груз, отгруженный со склада А3} \\ x_{11} \geq 0; x_{12} \geq 0; x_{13} \geq 0 \\ x_{21} \geq 0; x_{22} \geq 0; x_{23} \geq 0 \\ x_{31} \geq 0; x_{32} \geq 0; x_{33} \geq 0. \end{array} \right.$$

Представим математическую модель на рабочем листе в виде, изображенном на рис. П1.31а, записывая в ячейку В9 формулу функции цели, а в ячейки В11 — В16 формулы ограничений.

В диалоговом окне **Поиск решения** в соответствующих полях установим адрес целевой ячейки, адрес диапазона, содержащего подбираемые параметры, введем все ограничения, установим переключатель *Минимальному значению* (рис. П1.31б).

После выполнения вычислений будет получен результат, представленный на рис. П1.31в.

Минимальные затраты на перевозку грузов со складов потребителям при полученном плане перевозок составят 1360 у.е.

	A	B	C	D
1	Транспортная задача линейного программирования			
2				
3	Склады			
4	Получатели	A1	A2	A3
5	B1	14,9999993824129	15,0000016175871	109,999999
6	B2	0	299,999999	0
7	B3	75,0000006175871	84,9999993824129	0
8	Функция цели	$=2*B5+5*C5+2*D5+4*B6+C6+5*D6+3*B7$		
9				
10	Ограничения			
11	Груз для B1	$=B5+C5+D5$	140	
12	Груз для B2	$=B6+C6+D6$	300	
13	Груз для B3	$=B7+C7+D7$	160	
14	Груз со склада A1	$=B5+B6+B7$	90	
15	Груз со склада A2	$=C5+C6+C7$	400	
16	Груз со склада A3	$=D5+D6+D7$	110	
17				

Рис. П1.31а

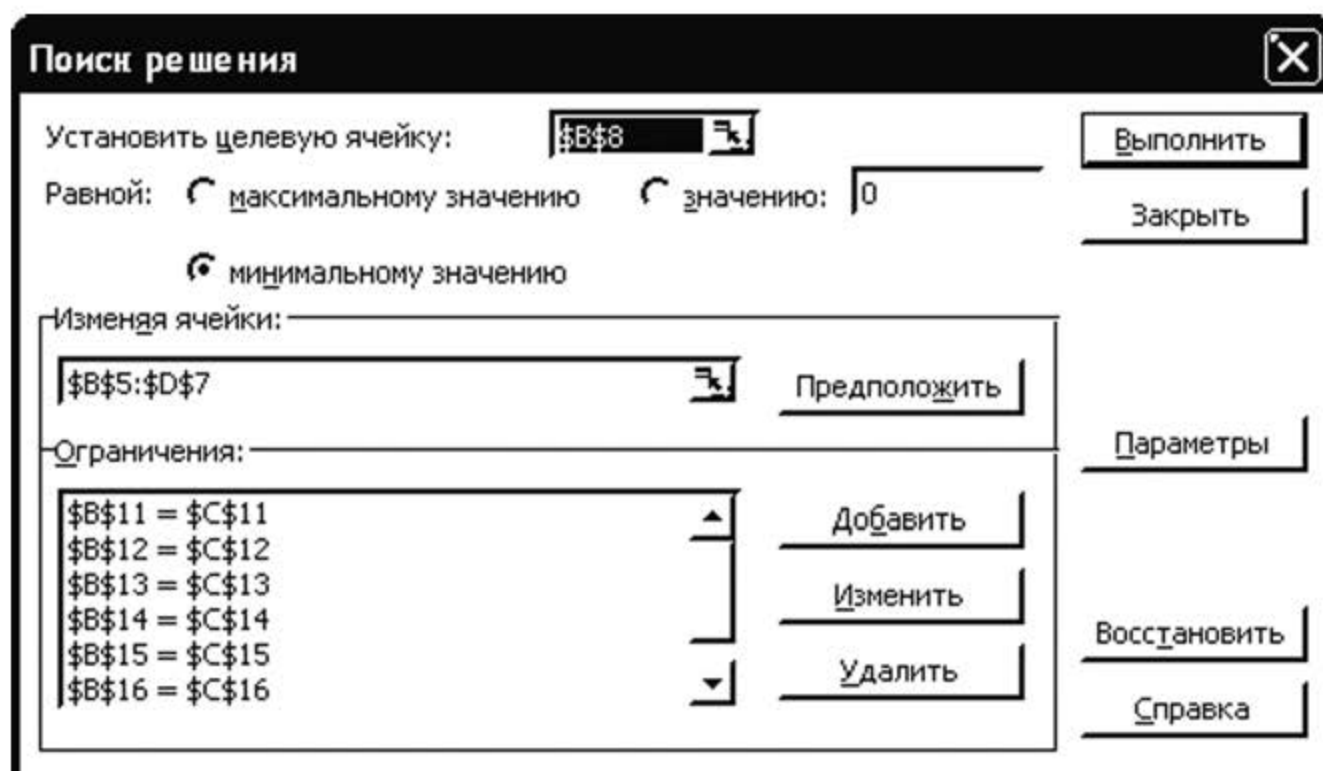


Рис. П1.31б

	A	B	C	D	E
1	Транспортная задача линейного программирования				
2					
3	Склады				
4	Получатели	A1	A2	A3	
5	B1	15,00	15,00	110,00	
6	B2	0,00	300,00	0,00	
7	B3	75,00	85,00	0,00	
8	Функция цели	1360,000002			
9					
10	Ограничения				
11	Груз для B1	140	140		
12	Груз для B2	300	300		
13	Груз для B3	160	160		
14	Груз со склада A1	90	90		
15	Груз со склада A2	400	400		
16	Груз со склада A3	110	110		
17					

Рис. П1.31в

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как могут быть представлены матрицы в табличном процессоре?
2. В чем заключается метод обратной матрицы для решения систем линейных уравнений?
3. Поясните технологию вычисления произведения матриц.
4. В каких случаях для решения систем линейных уравнений применяется метод наименьших квадратов, какова технология его применения?
5. Экономические задачи какого типа решаются с применением технологии решения систем линейных уравнений?
6. В чем заключается суть метода для вычисления предела числовой последовательности?
7. Какова технология построения графической модели функции, заданной табличной моделью?
8. Поясните метод конечных разностей для вычисления производной функции в точке.
9. Какими способами можно приближенно вычислить корни функции одного переменного?

10. Какими способами можно получить аналитическую модель функции, заданной таблично?
11. Как можно приближенно решить систему нелинейных уравнений?
12. Какие способы можно применить для уточнения решения системы нелинейных уравнений?
13. Как можно вычислить определенный интеграл численным методом?
14. Какие этапы содержит технология решения задач условной оптимизации?
15. Что представляет собой математическая модель задачи линейного программирования?
16. С использованием какого инструмента решаются задачи условной оптимизации в табличном процессоре?

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задание 1.1. Даны два массива, каждый элемент которых содержит числовые данные:

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix}; \quad B = \begin{pmatrix} 4 \\ 5 \end{pmatrix}.$$

Требуется перемножить значения элементов двумерного массива на значения элементов одномерного массива.

Задание 1.2. Вычислите в электронной таблице суммы векторов:

$$1) \bar{a} = \begin{pmatrix} 20 \\ 50 \\ 30 \\ 40 \end{pmatrix}; \quad \bar{b} = \begin{pmatrix} 5 \\ 2 \\ 7 \\ 4 \end{pmatrix}; \quad 2) \bar{a} = \begin{pmatrix} 10 \\ 15 \\ 30 \\ 45 \end{pmatrix}; \quad \bar{b} = \begin{pmatrix} 10 \\ 4 \\ 7 \\ 4 \end{pmatrix}.$$

Задание 1.3. Вычислить в электронной таблице произведение вектора на число:

$$1) \bar{a} = \begin{pmatrix} 20 \\ 50 \\ 30 \\ 40 \end{pmatrix}; \quad \lambda = 3; \quad 2) \bar{a} = \begin{pmatrix} 10 \\ 15 \\ 30 \\ 45 \end{pmatrix}; \quad \lambda = 12.$$

Задание 1.4. Вычислите в табличном процессоре скалярные произведения векторов:

$$1) \bar{a} = \begin{pmatrix} 20 \\ 50 \\ 30 \\ 40 \end{pmatrix}; \bar{b} = \begin{pmatrix} 5 \\ 2 \\ 7 \\ 4 \end{pmatrix}; \quad 2) \bar{a} = \begin{pmatrix} 10 \\ 15 \\ 30 \\ 45 \end{pmatrix}; \bar{b} = \begin{pmatrix} 10 \\ 4 \\ 7 \\ 4 \end{pmatrix}.$$

Задание 1.5.

1. Вычислите произведение матриц в в табличном процессоре:

$$а) A = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}; B = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}; \quad б) A = \begin{pmatrix} 2 & 3 & 5 \\ 4 & 1 & 3 \\ 7 & 2 & 0 \end{pmatrix}; B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

2. Найдите матрицу, обратную заданной:

$$A = \begin{pmatrix} 1 & -2 & 3 \\ 2 & 3 & -4 \\ 3 & -2 & -5 \end{pmatrix}.$$

3. Покажите вычислением, что для заданных матриц верно утверждение $(A + B)C = AC + BC$:

$$\text{для } A = \begin{pmatrix} 1 & 2 & 0 \\ 3 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix}; B = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 2 \\ 2 & -1 & 1 \end{pmatrix}; C = \begin{pmatrix} 1 & 0 & 2 \\ 2 & 1 & 1 \\ 1 & 3 & 1 \end{pmatrix}.$$

Задание 1.6. Решите системы линейных уравнений:

$$1. A = \begin{pmatrix} 9 & 5 & 4 & 7 \\ 4 & 6 & 8 & 7 \\ 5 & 8 & 7 & 6 \\ 5 & 6 & 8 & 7 \end{pmatrix}; B = \begin{pmatrix} 0 \\ 6 \\ 3 \\ 7 \end{pmatrix}.$$

$$2. A = \begin{pmatrix} 5 & 6 & 3 & 8 \\ 4 & 6 & 7 & 4 \\ 2 & 3 & 5 & 3 \\ 4 & 8 & 3 & 7 \end{pmatrix}; B = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix}.$$

При решении в электронной таблице используйте имена диапазонов. Выполните проверку решения.

3. Найдите решения систем уравнений:

$$\begin{cases} x + 2y + z = 1 \\ 2x + 3y + 2z = 2 \\ x - y + 3z = 0 \end{cases} \quad \begin{cases} 2x - y - z = 1 \\ 3x + 4y - 2z = 11 \\ 3x - 2y + 4z = 11. \end{cases}$$

Задание 1.7. Решите системы:

$$\begin{cases} 8x + 5y = 10 \\ 5x + 2y = 4 \\ 7x + 4y = 8 \end{cases} \quad \begin{cases} 2x - 4y + 3z = 1 \\ 3x - 4y + 5z = 2 \\ x - 2y + 4z = 3 \\ 6x - 2y + 10z = 4. \end{cases}$$

Задание 1.8. В таблице приведены данные о дневной производительности пяти предприятий холдинга, выпускающих четыре вида продукции с использованием трех видов сырья, а также продолжительность работы каждого предприятия в году и цена каждого вида сырья.

Вид изделия (№)	Производительность предприятия (изд./день)					Затраты видов сырья (ед. веса/изд.)		
	1	2	3	4	5	1	2	3
1	4	5	3	6	7	2	3	4
2	0	2	4	3	0	3	5	6
3	8	15	0	4	6	4	4	5
4	3	10	7	5	4	5	8	6
Количество рабочих дней в году						Цена видов сырья (ден. ед./ед. веса)		
	1	2	3	4	5	1	2	3
	200	150	170	120	140	40	50	60

Требуется определить:

- годовую производительность каждого предприятия по каждому виду изделий;
- годовую потребность каждого предприятия в каждом виде сырья;
- годовую сумму финансирования каждого предприятия для закупки сырья, необходимого для выпуска продукции указанных видов.

Задание 1.9

1. В таблице приведены данные баланса трех отраслей промышленности за некоторый период.

№ п/п	Отрасль	Потребление			Конечный продукт (ден. ед)	Валовой выпуск (ден. ед)
		1	2	3		
1	Добыча и переработка углеводородов	5	35	20	40	100
2	Энергетика	10	10	20	60	100
3	Машиностроение	20	1	10	10	50

Требуется найти объем валового выпуска каждого вида продукции, если конечное потребление по отраслям увеличить соответственно до 60, 70 и 30 условных денежных единиц.

2. Отрасль включает четыре предприятия. Вектор выпуска продукции и матрица коэффициентов прямых затрат имеют вид

$$\bar{x} = \begin{pmatrix} 400 \\ 300 \\ 250 \\ 300 \end{pmatrix}; A = \begin{pmatrix} 0,25 & 0,10 & 0,24 & 0,25 \\ 0,20 & 0,15 & 0,36 & 0,17 \\ 0,15 & 0,20 & 0,20 & 0,15 \\ 0,30 & 0,15 & 0,20 & 0,15 \end{pmatrix}.$$

Требуется найти вектор объемов конечного продукта, предназначенного для реализации вне отрасли.

Задание 1.10. Создайте в табличном процессоре числовые последовательности из 12 элементов: $\{n^*n^2\}$, $\{1/n\}$, $\{n/(n+1)\}$.

Задание 1.11.

1. Известно, что число e является пределом последовательности

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n.$$

Вычислите приближенно значение числа e .

2. Найдите предел $\lim_{n \rightarrow \infty} \left(\frac{2n^n + 2n + 4}{4n^2 + n - 3} \right)$.

3. Найдите предел последовательности $\{x_n\} = \sqrt{n+1} - \sqrt{n}$.

4. Цена товара равна 2000 руб. Темп инфляции равен 0,03% в день. Какова должна быть цена товара спустя 60 дней, чтобы компенсировать влияние инфляции?

5. Коммерческий банк аккумулирует средства предприятий в среднем на 6 месяцев. За это время он успевает три раза прокрутить эти

деньги в виде краткосрочных кредитов, выдаваемых частным предпринимателям на 3 месяца под 4% в месяц. Сколько процентов прибыли получает банк на этих операциях?

Задание 1.12.

1. Вычислите частичную сумму первых 7 членов ряда:

$$\frac{1}{1 \times 2} + \frac{1}{2 \times 3} + \dots + \frac{1}{n(n+1)} + \dots$$

2. Найдите частичную сумму первых 10 членов ряда:

$$\sum_{n=1}^{10} \frac{n+1}{\sqrt{n^3}}$$

Задание 1.13.

1. Сумма вклада в банк 2000 ед. Банк начисляет проценты по сложной ставке 6% годовых. Вычислите накопленную сумму на вкладе через 7 лет. Решите задачу двумя способами: с использованием функции БС и вычисляя частичную сумму ряда с использованием приведенной выше формулы.

2. Клиент взял кредит в сумме 100 000 руб. под 15% годовых на срок 6 месяцев. Проценты по кредиту начисляются ежемесячно. Какую сумму должен будет вернуть клиент кредитору по окончании срока кредита?

Задание 1.14. В таблице, приведенной ниже, заданы аналитические модели функций. Постройте в табличном процессоре табличную и графическую модель функции для выбранного варианта.

Вариант	1	2	3	4	5
Функция	$\sin(x)$	$\cos(x)$	$\lg(x)$	e^x	$\ln(x)$
Диапазон изменения аргумента	$(-p : +p)$	$(-p : +p)$	$(0 : 80)$	$0 : 3$	$(1 : 6)$
Вариант	6	7	8	9	10
Функция	$\operatorname{tg}(x)$	$2\sin(x) \cdot \cos(x)$	$\operatorname{ctg}(x)$	$2x - 4$	$\sin(x^2)$
Диапазон изменения аргумента	$(0 : 1,5)$	$(-p : +p)$	$(-p : +p)$	$(-1 : 2)$	$(0 : p)$

Задание 1.15. Вычислите:

1. Предел функции $\lim_{n \rightarrow \infty} \frac{\sin(x)}{x}$.

2. Предел функции $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x$.

Задание 1.16.

Найдите решения уравнений, приведенных в таблице.

Вариант	1	2	3	4
Уравнение	$2x - 4 = 0$	$2\sin(x) \cdot \cos(x) = 0$ $x \in [-p; p]$	$\lg(x + 2) - x = 0$	$\lg(x^2 + 2) - x = 0$
Вариант	5	6	7	8
Уравнение	$e^{(x+5)} - x = 0$	$x^3 + 2,84x^2 - 14,7 = 0$	$x^3 + 2,84x - 14,7 = 0$	$\cos(x) + \sin(x) = 0$ $x \in [-p; p]$
Вариант	9			
Уравнение	$e^{2x} - 3 = 0$			

Задание 1.17.

1. Вычислите все корни уравнения $X^3 + 2,84X^2 - 5,6064X - 14,766336 = 0$ на отрезке $\langle 3; -3 \rangle$ с относительной погрешностью 0,00001.

2. Решите уравнение $\sqrt{x^3 + 2x^2 - 5} = 0$.

Задание 1.18. Решите графически системы:

1. $\begin{cases} y = 2/x \\ y = 2x \end{cases}$ в диапазоне $x \in [0,2; 3]$ с точностью $\pm 0,2$.

2. $\begin{cases} y^2 + x^2 = 4 \\ y = 2\sin x \end{cases}$ в диапазоне $x \in [0,2; 3]$ с точностью $\pm 0,2$.

Задание 1.19. Зависимость спроса на товар от его цены выражается уравнением $3/x^2 + 15 = 0$, а зависимость предложения z от цены товара — уравнением $e^x - 1 = 0$. Требуется найти точку равновесия в диапазоне $x \in [0,2; 4]$ с точностью до третьего знака после запятой, а также эластичности спроса и предложения в точке равновесия.

Задание 1.20. Зависимость затрат от объема производства задана приведенной ниже таблицей. Требуется найти предельные издержки производства при объеме выпуска $x = 1,9$.

Указания: как видим, в задаче функция зависимости затрат от объема производства задана не аналитически, а таблично. Поэтому прежде всего нужно получить аналитическую зависимость в виде интерполяционной функции и, используя ее, решить задачу в соответствии с приведенной в предыдущем примере технологией.

Объем производства	Затраты
1	2,7
1,2	3,2
1,4	3,7
1,6	4,1
1,8	4,7
2	5,1
2,2	5,6
2,4	5,8

Задание 1.21. Торговая фирма по результатам реализации одного из видов товаров в течение нескольких периодов времени получила экспериментальные данные, приведенные в таблице. Требуется определить эластичность спроса и эластичность предложения при равновесной цене.

№ периода	Цена (усл. ед.)	Спрос	Предложение
1	0,1	3,0	2,0
2	0,2	2,8	2,5
3	0,3	2,4	3,5
4	0,4	2,1	3,9
5	0,5	2,0	4,7
6	0,6	1,7	5,2

Задание 1.22.

1. Вычислите двумя способами определенный интеграл $\int_0^{\pi/2} \sin(x) dx$.

2. Определите стоимость перевозки 20 т груза по железной дороге на расстояние 20 км при условии, что тариф перевозки одной тонны груза изменяется в соответствии с зависимостью $T = 150/s$, где s — расстояние в километрах.

3. Мощность потребляемой городом электроэнергии выражается формулой

$$y = \begin{cases} a, & t < 6 \\ a + b \sin \frac{\pi}{18}(t - 6), & t \geq 6, \end{cases}$$

где t — текущее время суток.

Вычислите суточное потребление электроэнергии при $a = 15\,000$ кВт, $b = 12\,000$ кВт.

4. Вычислите значение определенного интеграла $\int_0^5 (3x^2 + 1,5x + 5) dx$

и постройте график зависимости $y = f(x)$.

Вычисление интеграла выполните двумя способами.

5. Под строительство промышленного объекта задан непрерывный денежный поток со скоростью $I(t) = -t^2 + 20t + 5$ (млрд руб./год) в течение 20 лет с годовой процентной ставкой $p = 5\%$.

Требуется найти дисконтированную стоимость денежного потока.

Задание 1.23. Решите задачи:

1. $L(\bar{x}) = 3x_1 - 6x_2 \rightarrow \max(\min)$ при ограничениях

$$\begin{cases} x_1 - x_2 \geq 0 \\ -2x_1 + x_2 \leq 6 \\ 4x_1 \leq 7 \\ x_{1,2} \geq 0. \end{cases}$$

2. $L(\bar{x}) = 2x_1 + 3x_2 \rightarrow \max(\min)$ при ограничениях

$$\begin{cases} x_1 + 4x_2 \geq 8 \\ x_2 \leq 5 \\ x_1 \leq 4 \\ x_{1,2} \geq 0. \end{cases}$$

3. Фирма выпускает изделия двух видов — А и В. Для выпуска этих изделий используется сырье четырех типов. Расход сырья каждого типа на изготовление единицы продукции и запасы сырья заданы таблицей.

Расход на единицу продукции

Показатель	Тип сырья			
	1	2	3	4
Изделие				
А	2	1	0	2
В	3	0	1	1
Запас сырья	21	4	6	10

Выпуск одного изделия типа А приносит доход 300 ден. ед., а одного изделия типа В — 200 ден. ед.

Требуется составить план производства, обеспечивающей фирме наибольший доход.

4. Требуется спланировать перевозку строительных материалов с трех заводов к четырем строительным площадкам по железной дороге. В течение каждого квартала на четырех площадках требуется соответственно 5, 10, 20, 15 вагонов строительных материалов. Возможности заводов соответственно равны 10, 15 и 25 вагонов в квартал. В таблице приведены стоимости перевозки одного вагона (в ден. ед.) от различных заводов к различным строительным площадкам.

Стоимость перевозки одного вагона, ден. ед.

Номер завода	Номер строительной площадки			
	1	2	3	4
1	8	3	5	2
2	4	1	6	4
3	1	9	4	3

МОДЕЛИ И ТЕХНОЛОГИИ СТАТИСТИЧЕСКОГО АНАЛИЗА

2.1. ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ВЕЛИЧИН

Случайной называется переменная величина, принимающая различные числовые значения в зависимости от случая. В различных практических задачах используются два вида случайных величин — дискретные и непрерывные.

Случайная величина называется дискретной, если значения, которые она может принять, можно пронумеровать, или, иначе говоря, которая может принимать значения, образующие счетные множества.

Законом распределения случайной величины X называется соответствие между значениями случайной величины и вероятностями их реализации. Закон распределения может быть задан таблицей, формулой или графиком.

2.1.1. Генерация случайной величины, распределенной по равномерному закону

Дискретное равномерное распределение — это такое распределение, для которого вероятность каждого из значений случайной величины одна и та же, т.е.

$$P(x) = 1/N,$$

где N — количество возможных значений случайной величины.

Для получения случайной величины, распределенной по равномерному закону, в библиотеке **Мастера функций** табличного процессора в категории *Математические* есть специальная функция СЛЧИС(), которая генерирует случайные вещественные числа в диапазоне 0—1. Функция не имеет параметров. Если необходимо сгенерировать случайные числа в другом диапазоне, то для этого следует использовать формулу

$$=СЛЧИС()*(b-a) + a,$$

где a — число, устанавливающее нижнюю границу диапазона;
 b — число, устанавливающее верхнюю границу диапазона.

Например, для генерации чисел, распределенных по равномерному закону в диапазоне 10–20, нужно в ячейку рабочего листа ввести формулу

$$=СЛЧИС()*(20-10) + 10.$$

Для генерации целых случайных чисел, равномерно распределенных в диапазоне между двумя заданными числами, в библиотеке табличного процессора есть специальная функция СЛУЧМЕЖДУ. Функция имеет параметры

$$\text{СЛУЧМЕЖДУ(Нижн_гран; Верхн_гран)},$$

где Нижн_гран – число, устанавливающее нижнюю границу диапазона;

Верхн_гран – число, устанавливающее верхнюю границу диапазона.

Пример 2.1. Требуется создать массив из 10 чисел, распределенных равномерно в диапазоне 50–100.

Решение

1. Выделим диапазон, включающий 10 ячеек рабочего листа, например В2:В12.

2. Выполним команду меню **Вставка/Функция**.

3. В открывшемся окне диалога выберем категорию *Математические*, в списке функций – СЛЧИС, появится окно диалога **СЛЧИС**.

4. Нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter> – в выделенном диапазоне будут помещены числа, распределенные по равномерному закону в диапазоне 0–1.

5. Щелкнем указателем мыши в строке формул и изменим имеющуюся там формулу, приведя ее к виду: =СЛЧИС()*(100-50)+50.

6. Нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter> – в выделенный диапазон будут помещены числа, распределенные по равномерному закону в диапазоне 50–100.

		B3 ▾ fx {=СЛЧИС()*(100-50)+50}				
		A	B	C	D	E
1	Генерация случайных чисел					
2	№	X				
3	1	64,8723				
4	2	82,60913				
5	3	68,20304				
6	4	80,28283				
7	5	87,6051				
8	6	77,63741				
9	7	57,93881				
10	8	58,47647				
11	9	53,78759				
12	10	91,77204				

2.1.2. Генерация случайных чисел в табличном процессоре

В табличном процессоре для генерации случайных чисел есть специальный инструмент **Генерация случайных чисел**. Этот инструмент позволяет генерировать числа, распределенные по различным законам. В их число входят:

- равномерное распределение — инструмент позволяет генерировать заданное количество случайных чисел (по умолчанию в интервале 0–1);

- нормальное распределение — характеризуется средним значением и стандартным отклонением. Инструмент позволяет генерировать заданное количество случайных чисел, по умолчанию используется среднее значение 0 и стандартное отклонение 1;

- биномиальное распределение — характеризуется вероятностью успеха для некоторого числа испытаний, например можно сгенерировать случайные двухальтернативные переменные по числу испытаний, сумма которых будет биномиальной случайной величиной;

- дискретное распределение — характеризуется значением и соответствующим ему интервалом вероятности. Величины значений предварительно формируются в диапазоне ячеек в виде столбца, в смежном столбце, правее первого, указываются и соответствующие вероятности. Сумма вероятностей должна быть равна единице;

- распределения Бернулли, Пуассона и Модельное.

Для включения инструмента **Генерация случайных чисел** нужно выполнить команду меню **Сервис/Анализ данных**, в открывшемся окне диалога **Анализ данных** выбрать в списке *Генерация случайных чисел* (рис. 2.1) — откроется диалоговое окно **Генерация случайных чисел**.

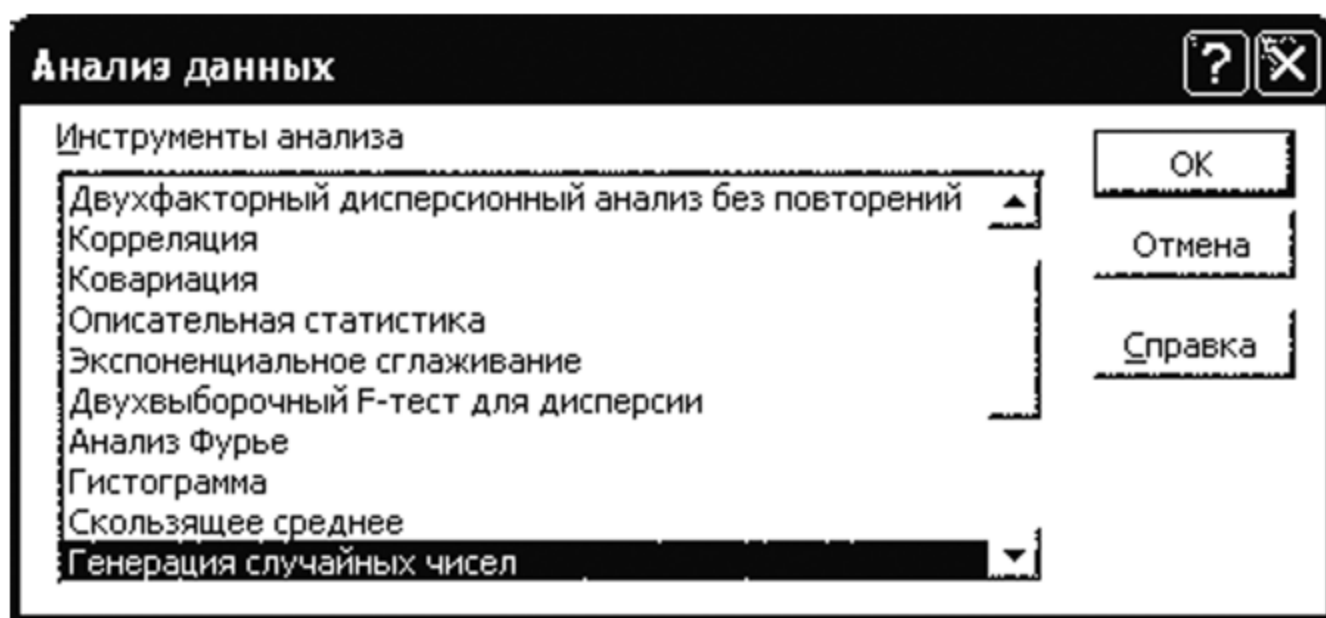


Рис. 2.1. Диалоговое окно Анализ данных

В рабочие поля группы *Параметры* вводятся параметры выбранного распределения. В поле *Случайное рассеивание* вводится произвольное значение, для которого нужно генерировать случайные числа. В поле *Выходной диапазон* вводится ссылка на верхнюю левую ячейку выходного диапазона.

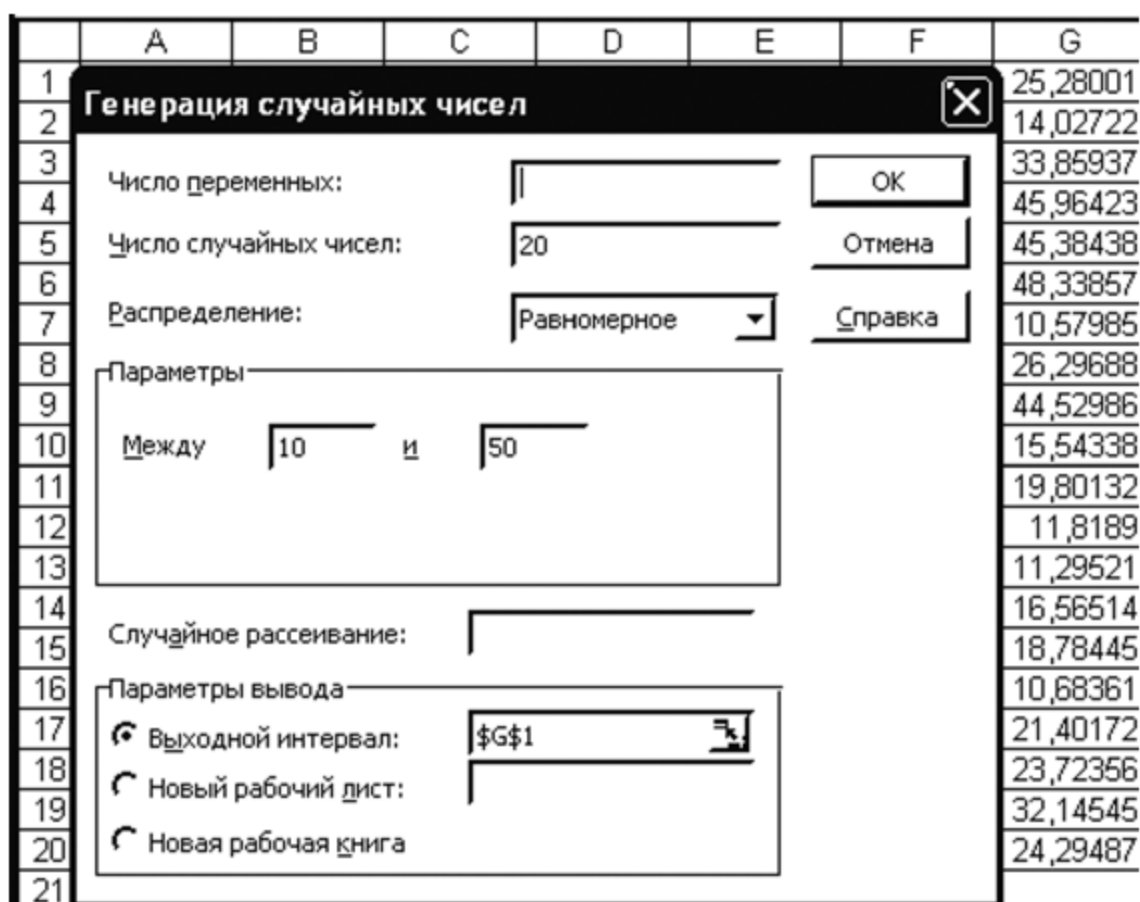
Рассмотрим на примерах технологию применения инструмента **Генерация случайных чисел** для генерации случайных чисел с различными законами распределения.

Пример 2.2. Требуется создать массив из 20 чисел, распределенных по равномерному закону в диапазоне 10–50.

Решение

1. Включим инструмент **Генерация случайных чисел**.
2. В поле *Число случайных чисел* введем число 20.
3. Раскроем список *Распределение* и выберем из него *Равномерное*.
4. В поля *Между* группы *Параметры* введем значения 10 и 50.
5. Установим курсор в поле *Выходной интервал* и щелкнем на ячейке рабочего листа (на рисунке G1), которая будет являться самой верхней левой ячейкой диапазона для вывода результата генерации.

Щелкнем на кнопке ОК, в диапазоне G1:G20 будут выведены числа, распределенные по равномерному закону на интервале 10–50.



2.2. ВЫЧИСЛЕНИЕ ЧИСЛОВЫХ ХАРАКТЕРИСТИК ПАРАМЕТРОВ СЛУЧАЙНЫХ ВЕЛИЧИН

2.2.1. Вычисление числовых характеристик распределений вероятностей

Числовыми характеристиками распределения вероятностей случайных величин, позволяющими наглядно получить представление о том или ином распределении, являются моменты и квантили.

Первым моментом случайной величины является математическое ожидание или среднее значение, которое характеризует центр распределения вероятностей.

Вторым моментом, характеризующим разброс случайной величины относительно математического ожидания, является центральный момент случайной величины, который называют дисперсией. Величина, равная корню квадратному из дисперсии, называется среднеквадратическим отклонением.

Для случайных величин, принимающих вещественные значения, используются такие характеристики, как квантили.

Квантилью X_p случайной величины, имеющей функцию распределения $F(x)$, называется решение X_p уравнения $F(x) = p$, где p — заданная вероятность.

Среди квантилей чаще всего используют медиану и квартили распределения.

Медианой называется квантиль, соответствующий значению $p = 0,5$. Верхним квартилем называется квантиль, соответствующий значению $p = 0,75$, а нижним — квантиль, соответствующий значению $p = 0,25$.

В табличном процессоре для вычисления некоторых числовых характеристик дискретных распределений вероятностей используются функции СРЗНАЧ, ДИСПР, СТАНДОТКЛОНП, КВАРТИЛЬ и ПЕРСЕНТИЛЬ:

СРЗНАЧ(*число_1; число_2;...*) — предназначена для вычисления математического ожидания всей совокупности значений дискретной случайной величины;

ДИСПР(*число_1; число_2;...*) — вычисляет дисперсию дискретного распределения;

СТАНДОТКЛОНП(*число_1; число_2; ...*) — позволяет оценить стандартное отклонение дискретного распределения;

КВАРТИЛЬ(*массив; значение*) — позволяет определить квартили распределения. Параметр массив представляет собой диапазон ячеек, содержащий значения дискретной случайной величины или сами значения, а параметр значение определяет, какой квартиль должна быть

вычислен (0 – минимальное значение распределения, 1 – нижний квартиль, 2 – медиана, 3 – верхний квартиль, 4 – максимальное значение распределения).

ПЕРСЕНТИЛЬ (*массив; K*) – позволяет вычислить p – квантили распределения. Параметр массив представляет собой диапазон ячеек, содержащий значения дискретной случайной величины или сами значения, параметр K – значение персентиля (от 0 до 1).

Пример 2.3. Дан набор случайных значений случайной величины 10, 14, 5, 6, 10, 12, 13. Требуется вычислить математическое ожидание, дисперсию, стандартное отклонение, верхний квартиль и квантиль со значением 0,1.

Решение

1. В диапазон ячеек рабочего листа (B2:B7) введем заданные числа.
2. Оформим таблицу, как показано на рисунке.
3. Используя приведенные выше статистические функции, вычислим требуемые числовые характеристики.

Вычисление числовых характеристик распределения случайной величины					Вычисление числовых характеристик распределения случайной величины				
1	№	X	Характеристики	Значение	1	№	X	Характеристики	Значение
2	1	10	Матем. Ожид.	10	2	1	10	Матем. Ожид.	=СРЗНАЧ(В3:В9)
3	2	14	Дисперсия	11,6666667	3	2	14	Дисперсия	=ДИСП(В3:В9)
4	3	5	Ст. отклон	3,41565026	4	3	5	Ст. отклон	=СТАНДОТКЛОН(В3:В9)
5	4	6	Верхн. Кварт.	12,5	5	4	6	Верхн. Кварт.	=КВАРТИЛЬ(В3:В9;3)
6	5	10	Персентиль	5,6	6	5	10	Персентиль	=ПЕРСЕНТИЛЬ(В3:В9;0,1)
7	6	12			7	6	12		
8	7	13			8	7	13		

2.3. ВЫЧИСЛЕНИЕ ВЕРОЯТНОСТИ ОТДЕЛЬНЫХ ЗНАЧЕНИЙ СЛУЧАЙНЫХ ВЕЛИЧИН

2.3.1. Табличный закон распределения

Напомним, что соответствие между отдельными возможными значениями случайной величины и их вероятностями называется законом распределения дискретной случайной величины. Закон распределения дискретной случайной величины может быть задан таблицей

X	x_1	x_2	...	x_n
P	p_1	p_2	...	p_n

При распределении, заданном таблично, математическое ожидание вычисляется по формуле

$$M(x) = x_1 p_1 + x_2 p_2 + \dots + x_n p_n.$$

Дисперсия дискретной случайной величины вычисляется по формуле

$$D(X) = M(X^2) - [M(X)]^2.$$

Среднеквадратичное отклонение дискретной случайной величины вычисляется по формуле $\sigma(X) = \sqrt{D(X)}$.

Пример 2.4. Вероятностный прогноз для величины X — процентного изменения стоимости акций по отношению к текущему курсу в течение 6 месяцев дан в виде закона распределения

№	1	2	3	4	5	6
X	5	10	15	20	25	30
P	0,1	0,1	0,2	0,3	0,2	0,1

Требуется вычислить вероятность того, что покупка акций будет более выгодна, чем помещение денег на банковский депозит под 3% за месяц сроком на 6 месяцев.

Решение

Вычислим в MS Excel прирост суммы по депозиту за 6 месяцев.

C17		fx
	A	B
1	Прирост по депозиту	
2	1	=1+0,03
3	2	=B2*0,03+B2
4	3	=B3*0,03+B3
5	4	=B4*0,03+B4
6	5	=B5*0,03+B5
7	6	=B6*0,03+B6
8	прирост	=B7-1

C17		fx
	A	B
1	Прирост по депозиту	
2	1	103,00%
3	2	106,09%
4	3	109,27%
5	4	112,55%
6	5	115,93%
7	6	119,41%
8	прирост	19,41%

Прирост суммы за 6 месяцев составит 19,41%.

Вероятность того, что покупка акций выгоднее депозита, определяется суммой вероятностей, соответствующих курсу акций более 19,41% и равна

$$P(X > 19,41) = P_4 + P_5 + P_6 = 0,3 + 0,2 + 0,1 = 0,6.$$

2.3.2. Биномиальное распределение

Биномиальное распределение — одно из самых распространенных дискретных распределений, которое служит моделью для многих процессов.

Для вычисления вероятности отдельного значения биномиального распределения или значения случайной величины по заданной вероятности в табличном процессоре есть функции БИНОМРАСП и КРИТБИНОМ.

Функция БИНОМРАСП применяется для вычисления вероятности в задачах с фиксированным числом испытаний или тестов, когда результатом любого испытания может быть только успех или неудача.

Функция имеет параметры:

БИНОМРАСП(число_успехов; число_испытаний;
вероятность_успеха; интегральная),

где *число_успехов* — количество успешных испытаний;

число_испытаний — количество независимых испытаний;

вероятность_успеха — вероятность успеха каждого испытания;

интегральная — логическое значение, определяющее форму функции. Если аргумент имеет значение ИСТИНА, то функция возвращает интегральную функцию распределения, т.е. вероятность того, что число успешных испытаний не менее значения аргумента число_успехов. Если этот аргумент имеет значение ЛОЖЬ, то вычисляется значение функции плотности распределения, т.е. вероятность того, что число успешных испытаний равно значению аргумента число_успехов.

Функция КРИТБИНОМ служит для вычисления наименьшего числа успешных исходов случайной величины, для которого интегральное биномиальное распределение больше или равно заданной величине (критерию).

Функция имеет параметры:

КРИТБИНОМ(число_испытаний; вероятность_успеха; альфа),

где *число_испытаний* — количество независимых двухальтернативных испытаний;

вероятность_успеха — вероятность успеха каждого испытания;

альфа — значение критерия, которое является уровнем значимости.

Рассмотрим на примерах технологию применения приведенных функций.

Пример 2.5. Банк выдает пять кредитов. Вероятность невозврата кредита равна 0,2 для каждого из заемщиков. Составить таблицу зако-

на распределения количества заемщиков, не вернувших кредит по окончании срока кредитования.

Решение

1. На рабочем листе подготовим исходные данные для расчета:
 - в ячейку A2 введем текст «Число испытаний», а в ячейку B2 – количество выданных кредитов – число 5 (рис. П2.5а);
 - в ячейку A3 введем текст «Вероятность невозврата», а в ячейку B3 – значение вероятности невозврата кредита – число 0,2;
 - в ячейку A4 введем текст «X» – обозначающий случайную величину;
 - в диапазоне B4:G4 построим последовательность членов арифметической прогрессии от 0 до 5;
 - в ячейку A5 введем текст «P», обозначающий вероятность невозврата кредита.

2. Установим курсор в ячейку B5. Выполним команду меню **Вставка/Функция**. В открывшемся окне диалога выберем категорию *Статистические* и функцию **БИНОМРАСП**.

3. В соответствующие поля диалогового окна **Аргументы функции** (рис. П2.5б) введем значения аргументов функции, делая ссылки на соответствующие ячейки, содержащие исходные данные. Аргументу *Интегральная* установим значение 0, что соответствует тому, что вероятность числа успешных испытаний будет равна значению аргумента *Число_успехов*.

4. Установим для соответствующих аргументов абсолютные адреса, используя клавишу <F4>.

5. Скопируем полученную формулу в диапазон C5:G5. В ячейках диапазона B5:C5 будет вычислен результат.

6. По данным полученной таблицы построим график (рис. П2.5в).

Из таблицы и графика следует, что вероятность невозврата четырех и пяти кредитов очень мала, а вероятность невозврата одного кредита является самой большой и составляет величину 0,4096.

G5		fx (=БИНОМРАСП(B4:G4;B2;B3;0))					
	A	B	C	D	E	F	G
1	Распределение количества заемщиков, не вернувших кредит						
2	Число испытаний	5					
3	Вероятность невозврата	0,2					
4	X	0	1	2	3	4	5
5	P	0,32768	0,4096	0,2048	0,0512	0,0064	0,00032

Рис. П2.5а

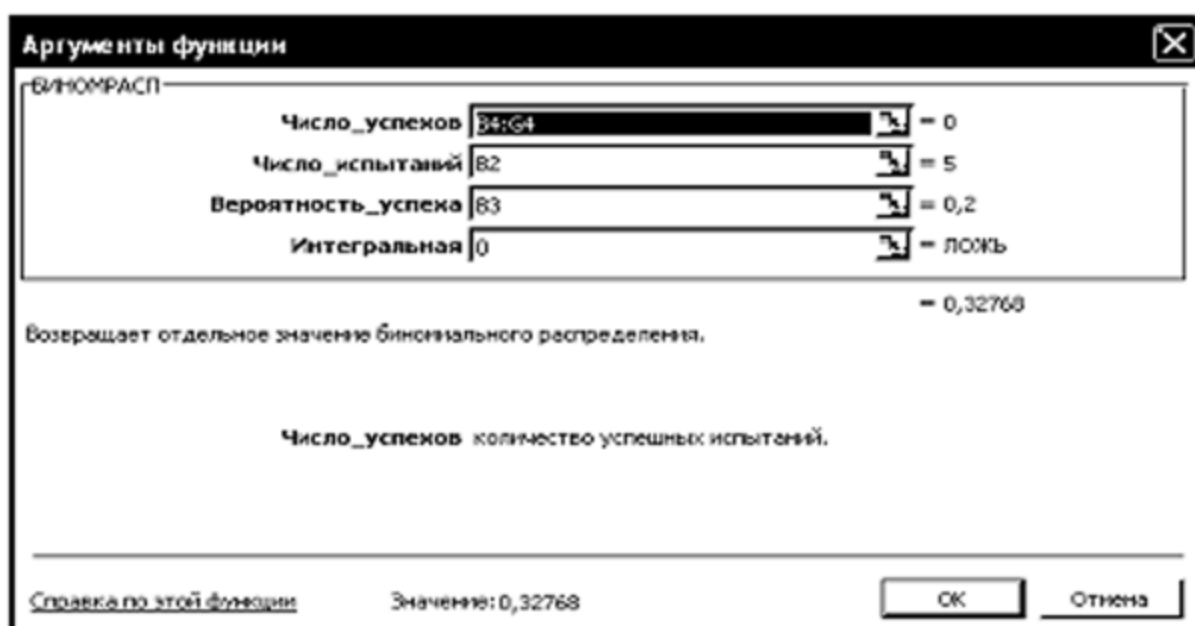


Рис. П2.5б



Рис. П2.5в

Пример 2.6. Для условий задачи предыдущего примера найти значение числа невозвращенных кредитов, для которого вероятность интегрального распределения $P \geq 0,4$.

Решение

1. Установим курсор в свободную ячейку рабочего листа. Выполним команду меню **Вставка/Функция**. В открывшемся окне диалога выберем категорию *Статистические* и функцию КРИТБИНОМ.

2. В окне Параметры функции введем значения параметров: *число_испытаний* – 5, *вероятность_успеха* – 0,2, *альфа* – 0,4.

После выполнения вычислений в ячейке будет получен результат равный единице. Таким образом, при вероятности интегрального распределения $P \geq 0,4$ будет не менее одного успешного события, т.е.

невозвращенных кредитов будет не менее одного. Это также можно увидеть из графика примера 2.5.

Пример 2.7. Построить диаграмму биномиальной функции плотности вероятности $P(A = m)$ при $n = 10$ и $p = 0,2$.

Решение

1. Диапазон рабочего листа (A3:A13) заполним возможными значениями исходов испытаний: 0, 1, 2, ..., 10.

2. Установим курсор в ячейку B3 и выполним команду меню **Вставка/Функция**. В открывшемся окне диалога выберем категорию *Статистические* и в списке функций – **БИНОМРАСП**.

3. В диалоговом окне **Параметры функции БИНОМРАСП** заполним поля параметров.

Число_успехов – введем количество успешных испытаний m , для чего щелкнем на ячейке A3;

Число_испытаний – введем общее количество проведенных испытаний – 10;

Вероятность_успеха – введем величину вероятности успеха в каждом испытании 0,2;

Интегральная – введем вид функции распределения 0 – *весовая*.

После щелчка на кнопке ОК в ячейке B3 будет вычислена вероятность $p = 0,1073774$ того, что из всех десяти испытаний будут все неудачные.

4. Скопируем формулу из ячейки B3 во все ячейки диапазона B4:B13.

Полученную таблицу оформим так, как изображено на рисунке, и построим гистограмму плотности вероятности.



2.3.3. Нормальный закон распределения

В табличном процессоре для вычисления значений нормального распределения есть специальные функции: НОРМРАСП, НОРМСТРАСП, НОРМОБР, НОРМСТОБР и НОРМАЛИЗАЦИЯ.

Функция НОРМРАСП вычисляет значения вероятности нормальной функции распределения для заданного среднего и стандартного отклонения. Она имеет параметры:

НОРМРАСП(x ; среднее; стандартное_откл; интегральная),

где x — значение, для которого строится распределение;

среднее — среднее арифметическое распределения;

стандартное_откл — стандартное отклонение распределения;

интегральная — логическое значение, определяющее форму функции.

Если параметр *интегральная* имеет значение ИСТИНА (1), то функция возвращает интегральную функцию распределения, в противном случае возвращает значение функции плотности распределения. Если параметры *среднее* = 0 и *стандартное_откл* = 1, то функция вычисляет стандартное нормальное распределение.

Для вычисления стандартного нормального интегрального распределения в библиотеке табличного процессора есть специальная функция НОРМСТРАСП. Она имеет параметры

НОРМСТРАСП(z),

где z — значение случайной величины, для которого вычисляется распределение.

Функция НОРМОБР служит для вычисления квантилей для указанного среднего и стандартного отклонения (решается уравнение $F(x) = p$). Функция имеет параметры

НОРМОБР(вероятность; среднее; стандартное_откл),

где *вероятность* — вероятность, соответствующая нормальному распределению;

среднее — среднее арифметическое распределения;

стандартное_откл — стандартное отклонение распределения.

Функция НОРМСТОБР предназначена для вычисления квантилей стандартного нормального распределения, единственным ее параметром является величина вероятности.

Функция НОРМАЛИЗАЦИЯ по заданному значению x и параметрам распределения вычисляет нормализованное значение, соответствующее x .

Рассмотрим технологию применения приведенных функций на примерах.

Пример 2.8. Требуется построить диаграмму нормальной функции плотности вероятности $f(x)$ при $M = 24,3$ и $\sigma = 1,5$.

Решение

1. В ячейку A3 введем символ x , а в ячейку B3 — символ функции плотности вероятности $f(x)$.

2. Вычислим нижнюю $M - 3\sigma$ границу диапазона значений x , для чего установим курсор в ячейку C2 и введем формулу $=24,3-3*1,5$, а также верхнюю границу — в ячейку E2 введем формулу $=24,3+3*1,5$.

3. Скопируем формулу из ячейки C2 в ячейку A4, полученное в ячейке A4 значение нижней границы будет началом последовательности арифметической прогрессии.

4. Создадим последовательность значений x в требуемом диапазоне, для чего установим курсор в ячейку A4 и выполним команду меню **Правка/Заполнить/Прогрессия**.

5. В открывшемся окне диалога **Прогрессия** установим переключатели *арифметическая*, по столбцам, в поле *Шаг* введем значение 0,5, а в поле *Предельное значение* — число, равное верхней границе диапазона.

6. Щелкнем на кнопке ОК. В диапазоне A4:A22 будет сформирована последовательность значений x .

7. Установим курсор в ячейку B4 и выполним команду меню **Вставка/Функция**. В открывшемся окне **Мастер функций** выберем категорию *Статистические*, а в списке функций — НОРМРАСП.

8. Установим значения параметров функции НОРМРАСП: для параметра x установим ссылку на ячейку A4, для параметра *Среднее* — введем число 24,3, для параметра *Стандартное откл* — число 1,5, для параметра *Интегральное* — число 0 (весовая).



9. Используя маркер буksировки, скопируем полученную формулу в диапазон ячеек B5:B22.

10. Выделим диапазон полученных табличных значений функции $f(x)$ (B3:B22) и выполним команду меню **Вставка/Диаграмма**. В окне **Мастер диаграмм** во вкладке *Стандартные* выберем *График*, а в поле *Вид* — вид графика, щелкнем на кнопке *Далее*.

11. В окне **Мастер диаграмм** (шаг 2) выберем закладку *Ряд*. В поле *Подписи оси x* укажем ссылку на диапазон, содержащий значения x (A4:A22). Щелкнем на кнопке *Далее*.

12. В окне **Мастер диаграмм** (шаг 3) введем подписи: *Название диаграммы*, *Ось x*, *Ось y*. Щелкнем на кнопке *Готово*. На рабочий лист будет выведена диаграмма плотности вероятности.



2.4. ТЕХНОЛОГИИ РЕШЕНИЯ ЗАДАЧ СТАТИСТИЧЕСКОГО АНАЛИЗА

2.4.1. Выборочный метод и выборочная функция распределения

На практике часто бывают ситуации, когда полное исследование каждого объекта из интересующей совокупности по различным причинам невозможно. В этих случаях из всей совокупности объектов случайным образом отбирают ограниченное число объектов и подвергают их исследованию. Вся совокупность объектов, из которых производится выборка, называется генеральной совокупностью. Совокупность

случайно отобранных из генеральной совокупности объектов называется выборочной совокупностью. Число объектов в совокупности называется ее объемом.

На практике сведения о законе распределения случайной величины получают независимыми многократными повторениями опыта. На основе информации из полученной выборки можно вычислить приблизительные значения для функции распределения и другие характеристики случайной величины.

Выборочной или эмпирической функцией распределения случайной величины, построенной по выборке, называют функцию, равную относительной частоте появления событий $F(x) = n_x/n$, где n_x — количество наблюдений, при которых значение признака x было меньше X , n — объем выборки (общее количество наблюдений).

Существует несколько видов отбора из генеральной совокупности. Рассмотрим один из них — с расчленением генеральной совокупности. Для построения выборочной функции распределения весь диапазон изменения случайной величины X разбивают на ряд интервалов (от 5 до 15) одинаковой ширины и затем вычисляют количество значений случайной величины X , попавших в каждый интервал.

2.4.2. Построение выборочной функции распределения

В табличном процессоре для построения выборочной функции распределения используются специальная функция ЧАСТОТА и инструмент пакета анализа **Гистограмма**.

Функция ЧАСТОТА вычисляет частоты появления случайных величин в интервалах значений и выводит их как массив чисел. Функция имеет параметры

ЧАСТОТА(массив_данных; массив_интервалов),

где *массив_данных* — массив или ссылка на диапазон данных, для которых вычисляются частоты;

массив_интервалов — массив или ссылка на множество интервалов, в которые группируются значения аргумента массив_данных.

Количество элементов в возвращаемом массиве на единицу больше, чем задано в параметре массив_интервалов. Дополнительный элемент содержит количество значений больших, чем максимальное значение в интервалах.

Инструмент **Гистограмма** служит для вычисления выборочных и интегральных частот попадания данных в указанные интервалы значений. Выходным результатом являются таблица и гистограмма.

Чтобы включить инструмент **Гистограмма**, следует выполнить команду меню **Сервис/Анализ данных**, далее в раскрывшемся диало-

говом окне **Анализ данных** из списка выбрать *Гистограмма* — откроется диалоговое окно **Гистограмма**. Вид диалогового окна **Гистограмма** приведен на рис. 2.2. Диалоговое окно имеет следующие поля:

— *Входной интервал* — поле, предназначенное для указания адресной ссылки на диапазон, содержащий исследуемые данные;

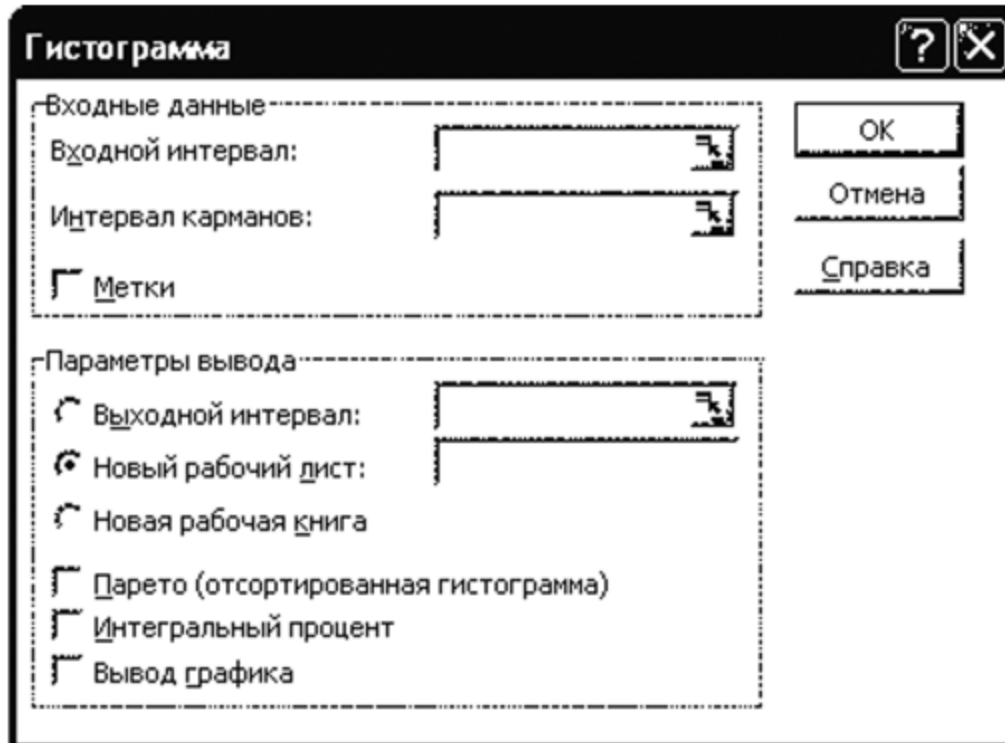


Рис. 2.2. Диалоговое окно Гистограмма

— *Интервал карманов* — поле, в котором может быть указана ссылка на диапазон ячеек, содержащий выбранные интервалы, в которые группируются значения аргумента *Входной интервал*;

— *Выходной диапазон* — поле предназначено для ввода адресной ссылки на верхнюю левую ячейку выходного диапазона;

— флажок *Интегральный процент* устанавливает режим генерации интегральных процентных соотношений и включает в гистограмму график интегральных процентов;

— флажок *Вывод графика* устанавливает режим автоматического вывода графика на рабочий лист, содержащий входной диапазон.

Пример 2.9. Требуется построить эмпирическое распределение рейтинга студентов по результатам экзаменов, оцененных в баллах, для следующей произвольной выборки: 48, 51, 64, 62, 55, 71, 74, 79, 80, 86, 91, 99, 83, 50. Задачу решить двумя способами: с применением функции **ЧАСТОТА**, с применением инструмента **Гистограмма** пакета анализа.

Решение с применением функции **ЧАСТОТА**

1. В ячейку A2 рабочего листа введем текст «Наблюдения», а в диапазон A3:A16 — числа из заданной выборки.

2. В ячейке В2 запишем текст «Шкала баллов», а в ячейки диапазона В3:В6 — баллы, соответствующие шкале для вывода пятибалльной оценки, — 50, 70, 85, 100. Это означает, что баллы диапазона 1–50 эквивалентны оценке «неудовлетворительно», баллы, находящиеся в диапазоне 51–70, — оценке «удовлетворительно» и т.д.

3. В ячейки С2, D2 и E2 введем тексты «Абсолютные частоты», «Относительные частоты» и «Накопленные частоты» соответственно. Абсолютные частоты — это частота попадания случайной величины из выборки (количество попаданий) в соответствующий интервал. Относительная частота представляет собой частное от деления значения абсолютной частоты на количество элементов выборки. Накопленные частоты — это сумма относительных частот.

4. Выделим диапазон С3:С7 и выполним команду меню **Вставка/Функция**. В открывшемся окне диалога **Мастер функций** выберем категорию *Статистические*, а в списке функций — функцию ЧАСТОТА. Раскроется диалоговое окно функции ЧАСТОТА.

5. Установим параметры функции:

— *массив_данных* — установим ссылку на диапазон, содержащий выборку случайных величин (А3:А16);

— *массив_интервалов* — установим ссылку на диапазон, содержащий шкалу для вывода оценки (В3:В6).

6. Так как функция ЧАСТОТА возвращает результат в диапазон в виде массива значений, нажмем комбинацию клавиш <Ctrl> + <Shift> + <Enter>. В ячейки диапазона С3:С7 будет выведен результат — абсолютные частоты попадания случайных величин в интервалы, заданные в ячейках диапазона В3:В6.



Таким образом, в результате проведенного исследования получены статистические оценки абсолютных частот по случайной выборке: неудовлетворительно — 2, удовлетворительно — 4, хорошо — 5, отлично — 3.

Решение с применением инструмента Гистограмма

1. В ячейку A2 рабочего листа введем текст «Наблюдения», а в диапазон A3:A16 — числа из заданной выборки.

2. В ячейке B2 запишем текст «Шкала баллов», а в ячейки диапазона B3:B6 — баллы, соответствующие шкале для вывода пятибалльной оценки.

3. Выполним команду меню **Сервис/Анализ данных** — откроется диалоговое окно **Анализ данных**.

4. В окне диалога **Анализ данных** выберем из списка *Гистограмма* — откроется диалоговое окно **Гистограмма**.

5. Введем параметры в соответствующие поля диалогового окна **Гистограмма**:

— *Входной интервал* — укажем ссылку на диапазон ячеек, в котором размещены результаты выборки (A3:A16);

— *Интервал карманов* — укажем ссылку на диапазон ячеек, содержащий выбранные интервалы, — шкалу для вывода оценки (B3:B6);

— установим переключатель *Выходной интервал*;

— *Выходной диапазон* — введем адресную ссылку на верхнюю левую ячейку выходного диапазона (C2);

— установим флажок *Интегральный процент*;

— установим флажок *Вывод графика*.

6. Щелкнем на кнопке ОК. В результате на рабочий лист будут выведены таблица и диаграмма.

Как видно из результатов решения задачи, оба рассмотренных способа дают одинаковые результаты.

Используя данные, полученные в результате расчетов в графе Накопленные частоты (Интегральный %), выборочную функцию распределения можно записать в виде

$$F(x) = \begin{cases} 0,1429 & x \leq 50 \\ 0,4286 & 50 < x \leq 70 \\ 0,7857 & 70 < x \leq 85 \\ 1 & 85 < x \leq 100. \end{cases}$$

	A	B	C	D	E	F	G	
1	Распределение результатов экзаменов							
2	Наблюдение	Шкала баллов	<i>Карман</i>	<i>Частота</i>	<i>Интегральный %</i>			
3	48	50	50	2	14,29%			
4	51	70	70	4	42,86%			
5	64	85	85	5	78,57%			
6	62	100	100	3	100,00%			
7	55	Еще		0	100,00%			
8	71							
9	74							
10	79							
11	80							
12	86							
13	91							
14	99							
15	83							
16	50							
17								
18								
19								

Гистограмма

Карман	Частота	Интегральный %
50	2	14,29%
70	4	42,86%
85	5	78,57%
100	3	100,00%
Еще	0	100,00%

2.5. ТЕХНОЛОГИИ ВЫЧИСЛЕНИЯ ОСНОВНЫХ СТАТИСТИЧЕСКИХ ХАРАКТЕРИСТИК

В библиотеке табличного процессора есть ряд специальных функций для вычисления выборочных характеристик, которые можно по назначению разбить на три группы:

- функции, характеризующие центр распределения;
- функции, характеризующие рассеивание;
- функции, позволяющие оценить форму эмпирического распределения.

Первую группу составляют функции:

СРЗНАЧ – вычисляет среднее арифметическое из одного или нескольких массивов чисел;

СРГАРМ – позволяет получить среднее гармоническое множества данных. Среднее гармоническое – это величина, обратная к среднему арифметическому обратных величин;

СРГЕОМ – вычисляет среднее геометрическое значений массива положительных чисел. Эту функцию можно использовать для вычисления средних показателей динамического ряда;

МЕДИАНА – позволяет получить медиану заданной выборки. Медиана – это элемент выборки, число элементов со значениями больше которого и меньше которого равно. Например, $\text{МЕДИАНА}(5; 6; 8; 5; 9; 10; 8; 9)$ равна 8;

МОДА — вычисляет количество наиболее часто встречающихся значений в выборке (наиболее вероятная величина).

Вторую группу составляют функции:

ДИСП — позволяет оценить дисперсию по выборочным данным — степень разброса элементов выборки относительно среднего значения;

СТАНДОТКЛОН — вычисляет стандартное отклонение — характеризует степень разброса элементов выборки относительно среднего значения;

ПЕРСЕНТИЛЬ — позволяет вычислить квантили заданной выборки.

В третью группу входят функции:

ЭКССЕСС — вычисляет оценку эксцесса по выборочным данным — степень выраженности хвостов распределения, т.е. частоты появления удаленных от среднего значения;

СКОС — позволяет оценить асимметрию выборочного распределения — величину, характеризующую несимметричность распределения элементов выборки относительно среднего значения.

Кроме функций в табличном процессоре есть набор инструментов для углубленного анализа данных, которые объединены общим названием **Пакет анализа**. Одним из них является инструмент **Описательная статистика**. Этот инструмент позволяет вычислить такие статистические характеристики, как среднее, стандартная ошибка (среднего), медиана, мода, стандартное отклонение, дисперсия выборки, эксцесс, асимметричность, интервал, минимум, максимум, сумму, наибольшее, наименьшее, счет, уровень надежности.

Рассмотрим на примерах технологии вычисления статистических характеристик с использованием библиотечных функций и инструмента **Описательная статистика**.

Пример 2.10. В таблице приведены сведения о ежемесячной реализации продукции за периоды до и после начала рекламной кампании.

Месяц	Реализация с рекламой (тыс. руб.)	Реализация без рекламы (тыс. руб.)
1	162	135
2	156	126
3	144	115
4	137	140
5	125	121
6	145	112
7	151	130

Требуется найти средние значения и стандартные отклонения приведенных данных.

Решение

1. На рабочем листе подготовим исходные данные для расчетов.

2. В ячейку В9 рабочего листа для вычисления среднего значения введем формулу =СРЗНАЧ(В2:В8), а в ячейку С9 – формулу =СРЗНАЧ(С2:С8).

3. Для вычисления стандартных отклонений в ячейку В10 введем формулу =СТАНДОТКЛОН(В2:В8), а в ячейку С10 – формулу =СТАНДОТКЛОН(С2:С8).

	А	В	С
	Месяц	Реализация с рекламой (тыс. руб.)	Реализация без рекламы (тыс. руб.)
1			
2	1	162	135
3	2	156	126
4	3	144	115
5	4	137	140
6	5	125	121
7	6	145	112
8	7	151	130
9	Среднее знач.	145,7142857	125,5714286
10	Станд. Откл.	12,29788987	10,27711281
11			

Пример 2.11. Даны выборки зарплат основных групп работников банка: администрации (менеджеров), персонала по работе с клиентами, технических служб. Полученные данные приведены в таблице.

Администрация	Персонал по работе с клиентами	Технические работники
4500	2100	3200
4000	2100	3000
3700	2000	2500
3000	2000	2000
2500	2000	1900
	1900	1800
	1800	
	1800	

Требуется вычислить основные статистические характеристики в группах данных.

Решение

1. Подготовим исходные данные для вычислений на рабочем листе.

2. Проведем статистическую обработку, для чего выполним команду меню **Сервис/Анализ данных**. В появившемся списке выберем инструмент **Описательная статистика**.

3. В открывшемся окне диалога в поле *Входной интервал* укажем исходный диапазон (A2:C10). В рабочем поле *Выходной интервал* укажем ссылку на ячейку, где будет помещаться верхняя левая ячейка таблицы результатов. В разделе *Группирование* установим переключатель *По столбцам* (исходные данные сгруппированы по столбцам). Установим флажок *Метки в первой строке*, в результирующую таблицу будут помещены надписи столбцов исходной таблицы. Установим флажок в поле *Итоговая статистика*. Щелкнем ОК. В указанном диапазоне для каждого столбца исходной таблицы будут выведены соответствующие статистические результаты.

	A	B	C	D	E	F	G	H	I
1	Описательная статистика зарплат работников банка								
2	Администрация	Персонал по работе с клиентами	Технический персонал						
3	4500	2100	3200						
4	4000	2100	3000	Среднее	3540	Среднее	1962,5	Среднее	2400
5	3700	2000	2500	Стандартная ошибка	355,8089375	Стандартная ошибка	41,9927715	Стандартная ошибка	243,584345
6	3000	2000	2000	Медиана	3700	Медиана	2000	Медиана	2250
7	2500	2000	1900	Мода	#N/D	Мода	2000	Мода	#N/D
8		1900	1800	Стандартное отклонение	795,6129712	Стандартное отклонение	118,773494	Стандартное отклонение	596,657356
9		1800		Дисперсия выборки	633000	Дисперсия выборки	14107,1429	Дисперсия выборки	356000
10		1800		Эксцесс	-1,29384635	Эксцесс	-1,2292902	Эксцесс	-2,0688678
11				Асимметричность	-0,24502455	Асимметричность	-0,3943257	Асимметричность	0,45760553
12				Интервал	2000	Интервал	300	Интервал	1400
13				Минимум	2500	Минимум	1800	Минимум	1800
14				Максимум	4500	Максимум	2100	Максимум	3200
15				Сумма	17700	Сумма	15700	Сумма	14400
16				Счет	5	Счет	8	Счет	6



Кроме рассмотренных ранее статистических характеристик в полученной таблице есть еще четыре:

- минимум — значение минимального элемента выборки;
- максимум — значение максимального элемента выборки;
- сумма — сумма значений всех элементов выборки;
- счет — количество элементов выборки.

2.6. ВЫЧИСЛЕНИЕ ДОВЕРИТЕЛЬНОГО ИНТЕРВАЛА ДЛЯ СРЕДНЕГО ЗНАЧЕНИЯ

Для вычисления доверительного интервала в MS Excel можно использовать специальную функцию ДОВЕРИТ или инструмент **Описательная статистика**.

Функция ДОВЕРИТ (*альфа; станд_откл; размер*) вычисляет ширину доверительного интервала. Ее параметрами являются:

- *альфа* – уровень значимости, используемый для вычисления доверительной вероятности;
- *станд_откл* – стандартное отклонение генеральной совокупности для интервала данных (предполагается известным или предварительно вычисляется);
- *размер* – размер выборки.

Пример 2.12. Требуется найти границы 90% интервала для среднего значения, если по результатам 24 торгов среднее значение стоимости доллара составило 28 руб., а стандартное отклонение – 35 коп.

Решение

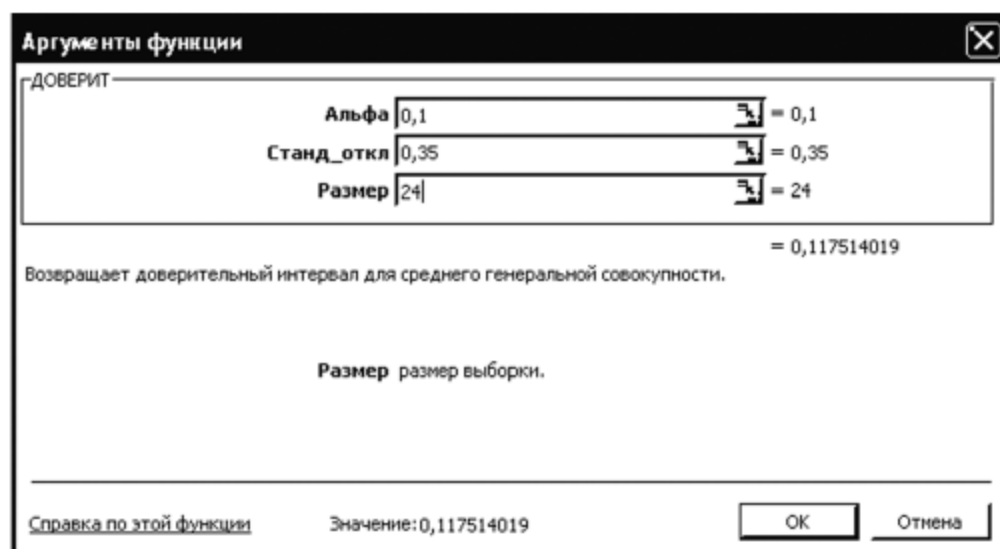
1. Установим курсор в любую свободную ячейку рабочего листа и установим для нее денежный формат.

2. Выполним команду меню **Вставка/Функция**. В окне **Мастер функций** в категории *Статистические* выберем из списка функцию ДОВЕРИТ.

3. В поля аргументов окна **ДОВЕРИТ** введем исходные данные: *альфа* – 0,1; *станд_откл* – 0,35; *размер* – 24.

После щелчка на ОК в ячейке будет вычислена полуширина 90% доверительного интервала для среднего значения выборки – 0,12 руб.

Таким образом, с 90%-ным уровнем надежности можно утверждать, что средняя стоимость доллара лежит в диапазоне 27 руб. 88 коп. – 28 руб. 12 коп.



Пример 2.13. Дана выборка стоимости валюты: 27,70; 27,85; 28,12; 28,20; 28,10; 27,75; 28,25 (рублей). Необходимо определить границы 95% доверительного интервала для среднего.

Решение

1. Введем в диапазон ячеек A2:A8 заданный массив чисел.
2. Включим инструмент **Описательная статистика**.
3. В поле *Входной интервал* диалогового окна **Описательная статистика** укажем ссылку на диапазон, содержащий выборку (A1:A8). Включим переключатель *Выходной диапазон* и в соответствующем поле укажем ссылку на ячейку, где будет размещен верхний левый угол результирующей таблицы (B2). Установим флажок *Уровень надежности* и в соответствующем поле введем число 95%. Установим флажок *Метки в первой строке*.
4. Щелкнем на ОК — на рабочий лист в указанный диапазон будет выведен результат.

	А	В	С
1	Выборка	<i>Выборка</i>	
2	27,7		
3	27,85	Среднее	27,99571
4	28,12	Стандартная ошибка	0,084765
5	28,2	Медиана	28,1
6	28,1	Мода	#Н/Д
7	27,75	Стандартное отклонение	0,224266
8	28,25	Дисперсия выборки	0,050295
9		Эксцесс	-2,07517
10		Асимметричность	-0,32615
11		Интервал	0,55
12		Минимум	27,7
13		Максимум	28,25
14		Сумма	195,97
15		Счет	7
16		Уровень надежности(95,0%)	0,207412

В результате вычислений для доверительной вероятности 0,95 в графе *Уровень надежности* получим величину доверительного интервала 0,207412. Это означает, что с вероятностью 0,95 для заданной генеральной совокупности среднее значение будет находиться в интервале $27,99571 \pm 0,207412$ (нижняя граница доверительного интервала — 27,7883, верхняя граница — 28,20312).

2.7. ТЕХНОЛОГИЯ ПРОВЕРКИ СООТВЕТСТВИЯ ДАННЫХ, ПОЛУЧЕННЫХ ЭКСПЕРИМЕНТАЛЬНО, ТЕОРЕТИЧЕСКОМУ РАСПРЕДЕЛЕНИЮ

При решении практических задач закон распределения случайных величин и его параметры неизвестны. Однако для решения задачи необходимо иметь информацию о том, каков закон распределения и каковы его параметры. В этом случае решают задачу проверки гипотезы. Исходя из предположения (гипотезы), что распределение случайных чисел подчиняется тому или иному закону, выполняют проверку этой гипотезы. Суть задачи проверки соответствия сводится к оценке меры соответствия экспериментальных данных и какого-либо теоретического распределения. Методом проверки соответствия теоретическому распределению является использование критерия согласия. Одним из них является критерий согласия хи-квадрат.

В табличном процессоре проверка согласия по критерию хи-квадрат реализуется функцией ХИ2ТЕСТ. Эта функция вычисляет вероятность совпадения наблюдаемых (фактических) значений и теоретических (гипотетических) значений. Если вычисленная вероятность ниже уровня значимости ($\alpha < 0,05$), то утверждается, что экспериментальные значения не соответствуют теоретическому распределению.

Функция имеет параметры:

ХИ2ТЕСТ(фактический интервал; ожидаемый интервал),

где *фактический интервал* — диапазон данных, который содержит результаты наблюдения, подлежащие сравнению с ожидаемыми значениями;

ожидаемый интервал — диапазон данных, который содержит теоретические (ожидаемые) значения для соответствующих наблюдаемых.

Для получения правильных результатов необходимо, чтобы объем выборки был не менее 40, выборочные данные сгруппированы в интервальный ряд с количеством интервалов не менее 7, а количество наблюдений в каждом интервале (частот) не менее 5.

Пример 2.14. Требуется проверить соответствие нормальному закону распределения выборочных данных результатов сдачи экзамена, оцененных в следующих баллах: 48, 51, 67, 70, 64, 71, 85, 79, 80, 83, 86, 91, 99, 56, 66, 65, 84, 84, 84, 75, 76, 77, 78, 80, 86, 88, 58, 69, 65, 81, 75, 78, 85, 80, 80, 83, 86, 80, 89, 60, 68, 55, 82, 64, 71, 72, 72, 73, 74, 74, 79.

Решение

1. В диапазон ячеек рабочего листа введем исходные данные в виде таблицы, содержащей баллы из приведенной выборки.

2. Выберем ширину интервала, равную 5 баллам, начиная от 50 до 100, и введем в диапазон F2:F12 граничные значения интервалов.

3. Подготовим заголовки создаваемой таблицы (ячейки G1, H1, I1).

4. Применяя функцию ЧАСТОТА, рассчитаем абсолютные частоты попаданий случайных величин в установленные интервалы — столбец Абсолютные частоты.

5. В ячейке H15 вычислим общее количество наблюдений, используя формулу =СУММ(G2:G11).

	A	B	C	D
1	выборка			
2	48	56	58	60
3	51	66	69	68
4	67	65	65	55
5	70	84	81	82
6	64	84	75	64
7	71	84	78	71
8	85	75	85	72
9	79	76	80	72
10	80	77	80	73
11	83	78	83	74
12	86	80	86	74
13	91	86	80	79
14	99	88	89	
15				

6. В ячейке H16 вычислим среднее значение выборки, а в ячейке H17 — стандартное отклонение.

7. Вычислим теоретические частоты распределения. Поскольку мы проверяем соответствие заданной совокупности случайных величин нормальному закону распределения, то для расчета применим функцию НОРМРАСП. Установим курсор в ячейку H2 и вызовем из Мастера функций функцию НОРМРАСП. Заполним поля аргументов: x — F2, среднее — \$H\$16, стандартное_откл. — \$H\$17, интегральный — 0, щелкнем на ОК.

8. В ячейку H3 введем формулу

=НОРМРАСП(F3;\$H\$16;\$H\$17;1)–СУММ(\$H\$2:H2).

9. Скопируем введенную формулу в ячейки диапазона H4:H12.

Для вычисления теоретических частот установим курсор в ячейку I2 и введем формулу = \$H\$16* H2. Скопируем содержимое этой ячейки в ячейки диапазона I3:I12.

E	F	G	H	I
	Шкала баллов	Абсолютные частоты	Теоретические частоты	Теоретические частоты
	50	1	0,009537747	0,486425085
	55	2	0,021071843	1,074663987
	60	3	0,050182132	2,559288749
	65	4	0,096009959	4,896507932
	70	5	0,147580522	7,526606645
	75	9	0,182265414	9,295536138
	80	12	0,180863255	9,224026027
	85	9	0,144200566	7,354228847
	90	4	0,092373018	4,711023935
	95	1	0,047540898	2,424585813
	100	1	0,019656589	1,00248604
Соответствие нормальному закону				0,917143314
		Всего	51	
		Среднее	74,82352941	
Стандартное отклонение			10,59000639	

10. Применяя функцию ХИ2ТЕСТ, определим соответствие данных выборки нормальному закону распределения. Для этого:

– установим курсор в свободную ячейку I14, включим **Мастер функций**, выберем категорию *Статистические*, а в списке функций – функцию ХИ2ТЕСТ;

– заполним поля аргументов функции: *фактический* – введем адрес диапазона абсолютных частот G2:G12, *ожидаемый* – адрес диапазона теоретических частот I2:I12. После щелчка на кнопке ОК в ячейке I14 будет вычислено значение вероятности того, что выборочные данные соответствуют нормальному закону распределения – 0,917143314.

Поскольку полученная вероятность соответствия экспериментальных данных $p = 0,917143314$ намного больше уровня значимости $\alpha = 0,05$, то можно утверждать, нулевая гипотеза не может быть отвергнута и экспериментальные данные не противоречат нормальному закону распределения. Но так как полученное значение вероятности очень мало отличается от 1, то можно говорить о высокой степени вероятности того, что экспериментальные данные соответствуют нормальному закону.

2.8. РЕШЕНИЕ ЗАДАЧ СТАТИСТИЧЕСКОГО АНАЛИЗА

2.8.1. Технология решения задач дисперсионного анализа

Известно, что методы дисперсионного анализа используются для оценки достоверности различий между несколькими группами наблюдений. Задача дисперсионного анализа заключается в исследовании воздействия на изменяемую случайную величину одного или нескольких независимых факторов, имеющих несколько градаций.

В табличном процессоре для проведения однофакторного дисперсионного анализа применяется инструмент **Однофакторный дисперсионный анализ**. Кроме этого инструмента в табличном процессоре есть инструменты **Двухфакторный дисперсионный анализ с повторениями** и **Двухфакторный дисперсионный анализ без повторений**.

Для выполнения дисперсионного анализа необходимо выполнить следующую последовательность операций.

1. Сформировать таблицу данных таким образом, чтобы в каждом столбце рабочего листа были представлены данные, соответствующие одному значению исследуемого фактора, при этом столбцы должны располагаться в порядке возрастания (убывания) исследуемого фактора.

2. Выполнить команду меню **Сервис/Анализ данных**. В диалоговом окне **Анализ данных** в списке *Инструменты анализа* выбрать инструмент **Однофакторный дисперсионный анализ**, щелкнуть на кнопке ОК.

3. В раскрывшемся окне диалога поле *Входной интервал* ввести ссылку на диапазон исследуемых данных, в группе *Группировка* установить переключатель *По столбцам*. Ввести ссылку на выходной диапазон, в который будут выведены результаты анализа, щелкнуть на кнопке ОК.

Выходной диапазон содержит следующие результаты: средние, дисперсии, критерии Фишера и др.

Влияние исследуемого фактора определяется по величине значимости критерия Фишера, находящегося в таблице *Дисперсионный анализ* на пересечении строки *Между группами* и столбца *P-Значение*. Если величина значимости P-Значение меньше 0,05, то критерий Фишера значим и, следовательно, влияние исследуемого фактора существует.

Пример 2.15. Необходимо выявить, влияет ли расстояние от центра города на степень заполнение гостиниц. Пусть расстояние от центра разбито на 3 уровня: 1) до 3 км; 2) от 3 до 5 км; 3) более 5 км.

В этой задаче исследуемым фактором является расстояние гостиницы от центра города.

Заполняемость гостиниц

Расстояние (км)	Заполняемость в %					
До 3	92,00	98,00	89,00	97,00	90,00	94,00
От 3 до 5	90,00	86,00	84,00	91,00	83,00	82,00
Более 5	87,00	79,00	74,00	85,00	73,00	77,00

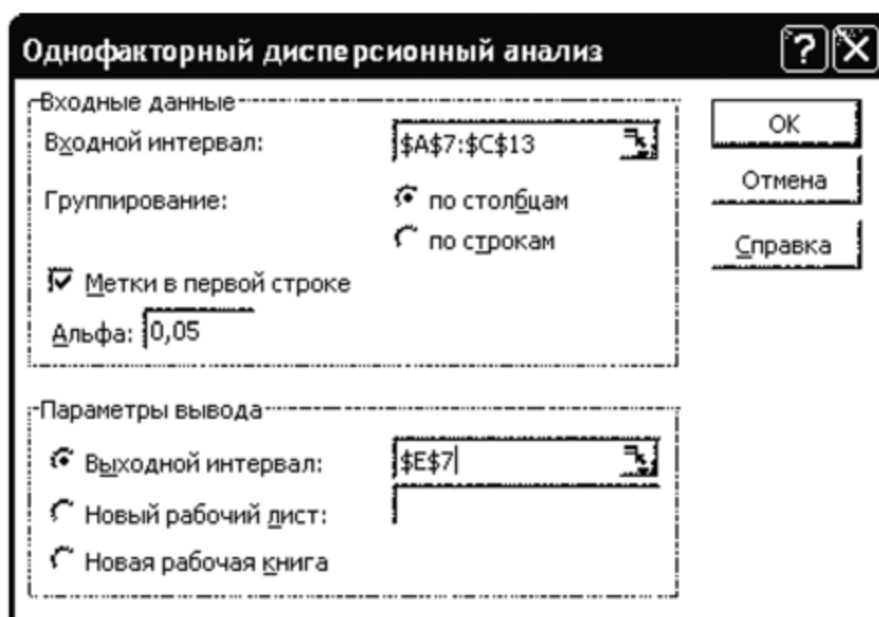
Решение

1. Подготовим на рабочем листе исходные данные для расчетов в соответствии с таблицей.

	A	B	C
6			
7	до 3 км	от 3 до 5 км	более 5 км
8	0,92	0,9	0,87
9	0,98	0,86	0,79
10	0,89	0,84	0,74
11	0,97	0,91	0,85
12	0,9	0,83	0,73
13	0,94	0,82	0,77
14			

2. Выполним команду меню **Сервис/Анализ данных**. Выберем из списка диалогового окна **Анализ данных** элемент *Однофакторный дисперсионный анализ*.

3. В раскрывшемся диалоговом окне **Однофакторный дисперсионный анализ** установим параметры, как показано на рисунке.



В результате решения задачи на рабочем листе будет сформирована таблица Однофакторный дисперсионный анализ.

	E	F	G	H	I	J	K
7	Однофакторный дисперсионный анализ						
8							
9	ИТОГИ						
10	<i>Группы</i>	<i>Счет</i>	<i>Сумма</i>	<i>Среднее</i>	<i>Дисперсия</i>		
11	до 3 км	6	5,6	0,933333	0,0013467		
12	от 3 до 5 км	6	5,16	0,86	0,0014		
13	более 5 км	6	4,75	0,791667	0,0032967		
14							
15							
16	Дисперсионный анализ						
	<i>Источник вариации</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-Значение</i>	<i>F критическое</i>
17	Между группами	0,060233	2	0,030117	14,950359	0,000268401	3,682316674
18	Внутри групп	0,030217	15	0,002014			
19							
20							
21	Итого	0,09045	17				
22							

В этой таблице на пересечении строки *Между группами* и столбца *P-Значение* записано число 0,000268401. Эта величина *P-Значения* меньше 0,05. Следовательно, критерий Фишера значим. А так как мы оценивали различие между тремя группами наблюдений, то следует сделать вывод, что влияние фактора расстояния от центра города на заполнение гостиниц подтверждено статистически.

2.8.2. Технологии решения задач корреляционного анализа

Одной из задач статистического моделирования является изучение связи между некоторыми наблюдаемыми переменными. Результаты, полученные при таком исследовании, позволяют прогнозировать развитие ситуации в случае изменения конкретных характеристик изучаемого объекта или процесса. Задача подобного исследования решается методами корреляционного анализа.

Под корреляцией понимается взаимосвязь некоторых величин, представленных данными — векторами или матрицами. Общепринятой мерой линейной корреляции является коэффициент корреляции. Его близость к единице указывает на высокую степень линейной зависимости. Целью решения задачи является получение корреляционной матрицы, которая представляет собой симметричную матрицу с единичной диагональю.

В табличном процессоре для целей корреляционного анализа служит инструмент **Корреляция**, который позволяет получить корреляционную матрицу, содержащую коэффициенты корреляции между различными параметрами. Корреляционная матрица — это квадратная таблица, на пересечении соответствующих строк и столбцов которой располагаются корреляционные коэффициенты.

Для выполнения корреляционного анализа необходимо выполнить следующую последовательность операций.

1. Выполнить команду меню **Сервис/Анализ данных**.

2. В списке *Инструменты анализа* диалогового окна **Анализ данных** выбрать строку *Корреляция*.

3. В раскрывшемся окне диалога **Корреляция** в поле *Входной интервал* ввести адресную ссылку на диапазон, содержащий анализируемые данные (входной интервал должен состоять не менее чем из двух столбцов), установить соответствующий переключатель *Группировка*, установить переключатель *Параметры вывода* и ввести адрес верхней левой ячейки, с которой будет начинаться диапазон для вывода результатов вычислений.

4. Щелкнуть на кнопке **ОК**.

В результате выполнения вычислений в выходной диапазон будет помещена корреляционная матрица.

Пример 2.16. Есть статистические данные, регистрирующие количество выходных и праздничных дней в месяце в период с января по июнь и снимаемые со счетов суммы.

Данные наблюдений

Месяц	Количество праздничных и выходных дней	Суммы, снятые со счетов
Январь	12	120 458
Февраль	6	95 700
Март	5	75 678
Апрель	4	51 675
Май	7	81 900
Июнь	4	59 500

Требуется определить, существует ли взаимосвязь между количеством выходных и праздничных дней и величиной сумм, снимаемых со счетов.

Решение

1. Введем в диапазон рабочего листа (A1:C8) данные приведенной таблицы.

2. Выполним команду меню **Сервис/Анализ данных**.

3. В раскрывшемся окне диалога **Анализ данных** из списка выберем *Корреляция*.

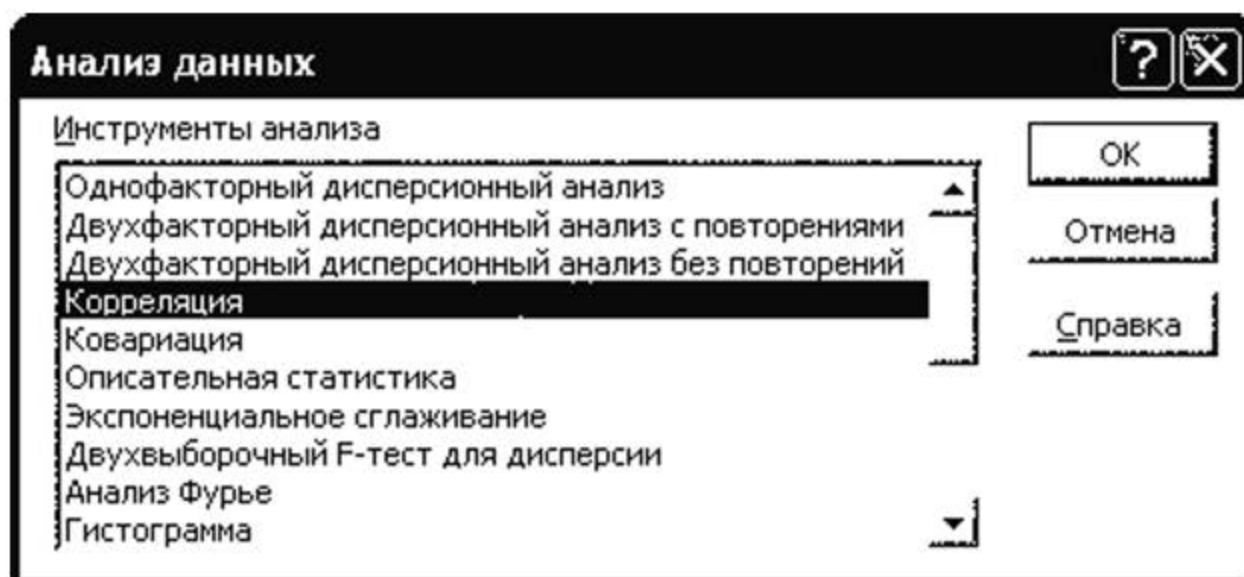
4. В окне диалога **Корреляция**:

– в группе *Входные данные* в поле *Входной интервал* укажем адресную ссылку на диапазон, содержащий исходную таблицу (B2:C8), установим переключатель *Группирование по столбцам* и установим флажок *Метки в первой строке*;

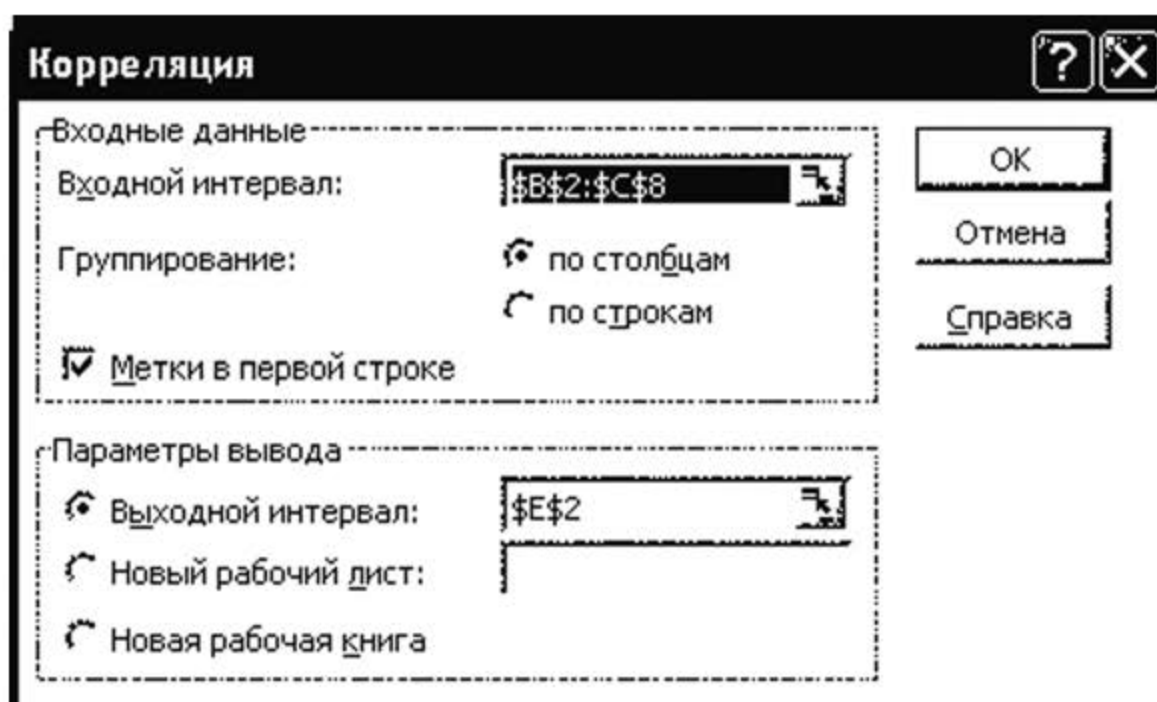
– в группе *Параметры вывода* установим переключатель *Выходной интервал* и введем адресную ссылку на ячейку, которая будет началом диапазона выходного интервала (E2).

Щелкнем на ОК.

	А	В	С
1	Данные наблюдений		
2	Месяц	Количество праздничны и выходных дней	Суммы, снятые со счетов
3	январь	12	120458
4	февраль	6	95700
5	март	5	75678
6	апрель	4	51675
7	май	7	81900
8	июнь	4	59500
9			



После выполнения вычислений в диапазон E2:G4 будет выведена результирующая корреляционная матрица.



	E	F	G
1	Корреляционная матрица		
2		<i>Количество праздничны и выходных дней</i>	<i>Суммы, снятые со счетов</i>
3	<i>Количество праздничны и выходных дней</i>	1	
4	<i>Суммы, снятые со счетов</i>	0,916688656	1
5			

Корреляционная матрица является треугольной с единичной диагональю. На самом деле она должна рассматриваться как симметричная, т.е. $r_{ij} = r_{ji}$.

Из анализа полученной корреляционной матрицы видно, что коэффициент корреляции между количеством выходных дней и величиной суммы, снимаемой со счетов $r = 0,91668$. Таким образом, существует сильная прямая линейная связь между количеством выходных и праздничных дней и величинами сумм, снимаемых со счетов.

2.8.3. Технология решения задач регрессионного анализа

Важную роль при исследовании взаимосвязей между статистическими выборками кроме корреляционного и дисперсионного анализа играет регрессионный анализ. Регрессия позволяет проанализировать воздействие на какую-либо зависимую переменную одной или более независимых переменных и позволяет установить аналитическую форму (модель) этой зависимости в виде аппроксимирующего полинома.

Если рассматривается зависимость между одной зависимой переменной y и несколькими независимыми x_1, x_2, \dots, x_n , то речь идет о множественной линейной регрессии. В этом случае уравнение регрессии имеет вид

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n,$$

где a_1, a_2, \dots, a_n — коэффициенты при независимых переменных, которые нужно вычислить (коэффициенты регрессии);

a_0 — константа.

При построении регрессионной модели важнейшими моментами являются оценка ее адекватности (эффективности) и значимости, на основании которых можно судить о возможности применения в практике полученной модели.

Мерой оценки адекватности регрессионной модели является коэффициент детерминации R^2 (R -квадрат), который определяет, с какой степенью точности полученное уравнение регрессии аппроксимирует исходные данные.

Значимость регрессионной модели оценивается с помощью критерия Фишера (F -критерия). Если величина F -критерия значима ($p < 0,05$), то регрессионная модель является значимой.

В табличном процессоре можно аппроксимировать экспериментальные данные линейным уравнением до 16-го порядка

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_{16}x_{16}.$$

Для вычисления коэффициентов регрессии служит инструмент **Регрессия**, который можно включить следующей последовательностью операций.

1. Выполнить команду **Сервис/Анализ данных**.
2. В раскрывшемся окне диалога **Анализ данных** выбрать из списка строку *Регрессия* — раскроется окно диалога **Регрессия**.
3. В группе *Входные данные* в поле *Входной интервал y* указать адресную ссылку на диапазон, содержащий значения зависимой переменной, а в поле *Входной интервал X* — ссылку на диапазон, содержащий значения независимых переменных, т.е. переменных, влияние

которых на зависимую переменную y оценивается. Установить флажок *Метки*, если исходная таблица имеет названия столбцов и флажок *Константа-ноль*, если $a_0 = 0$.

В группе *Параметры выхода* указать адресную ссылку на ячейку рабочего листа, которая будет являться верхней левой ячейкой результирующей таблицы.

Если необходимо получить визуальную картинку отличия экспериментальных точек от предсказанных регрессионной моделью, то установить флажок *График подбора*.

Если нужно получить график нормальной вероятности, то установить флажок *График нормальной вероятности*.

В выходном диапазоне после выполнения вычислений отображаются результаты дисперсионного анализа, коэффициенты регрессии, стандартная погрешность вычисления y , среднеквадратичные отклонения, количество наблюдений, стандартные погрешности для коэффициентов.

Значения коэффициентов регрессии размещаются в столбце *Коэффициенты*:

- y – пересечение a_0 ;
- x_1 – коэффициент a_1 ;
- x_2 – коэффициент a_2 и т.д.

В столбце *P-Значение* содержится оценка достоверности отличия соответствующих коэффициентов от нуля. Если $P > 0,05$, то коэффициент можно считать нулевым. Это означает, что соответствующая независимая переменная практически не влияет на зависимую переменную.

Значение *R-квадрат* определяет, с какой степенью точности регрессионное уравнение будет аппроксимировать экспериментальные данные. Если *R-квадрат* $> 0,95$, то точность аппроксимации высокая. При $0,8 < R\text{-квадрат} < 0,95$ аппроксимация удовлетворительная. В случае когда *R-квадрат* $< 0,6$, точность аппроксимации недостаточна и модель требует улучшения.

Кроме инструмента **Регрессия** в табличном процессоре для получения параметров уравнения регрессии есть функция ЛИНЕЙН и функция ТЕНДЕНЦИЯ для получения значения y в требуемых точках.

Пример 2.16. Имеются статистические данные о затратах, связанных с рекламой по телевидению, с рекламой в метро и объеме реализации продукции в рублях, приведенные в таблице.

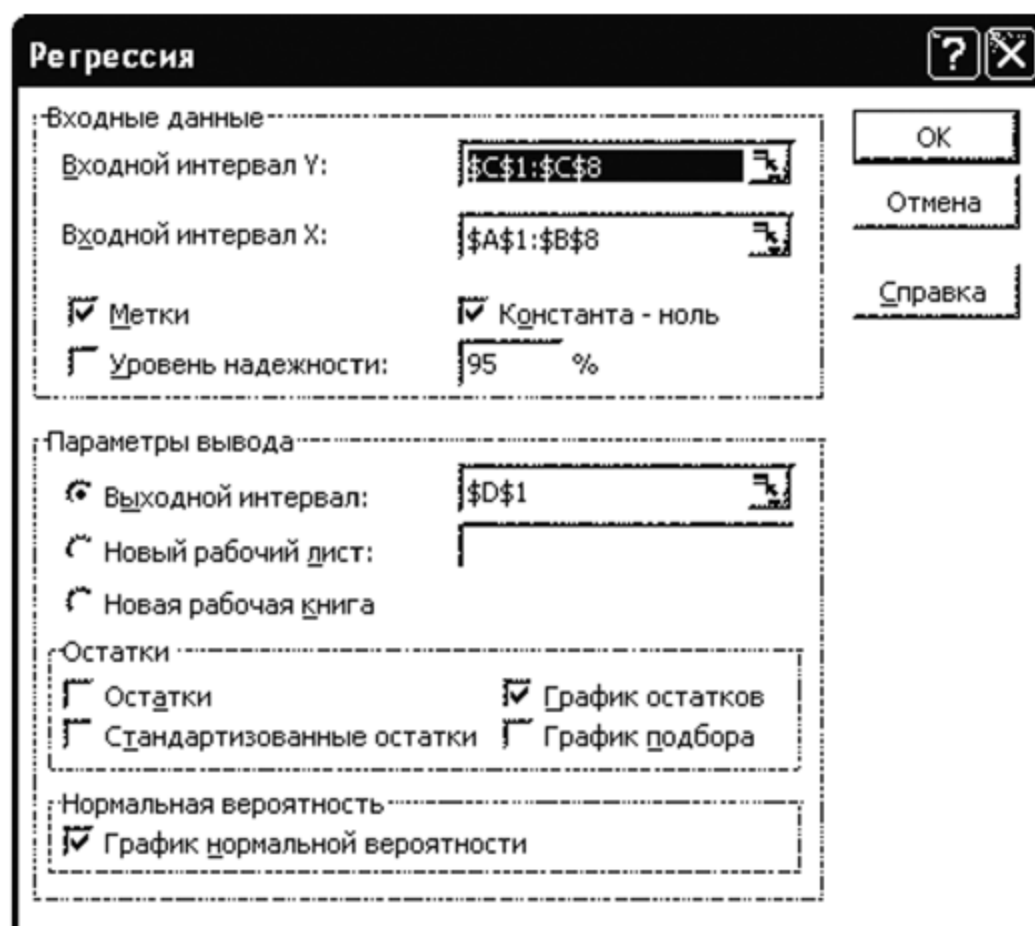
Требуется найти регрессионные коэффициенты для независимых переменных *Расходы на рекламу по телевидению* и *Расходы на рекламу*

в метро на объем реализации продукции и построить уравнение регрессии.

Затраты на рекламу по телевидению (руб.)	Затраты на рекламу в метро (руб.)	Объем реализации (руб.)
125850	20000	1850000
260500	21000	2500000
150000	22000	1900000
300000	21000	3000000
280000	24000	2600000
290000	23000	2700000
140000	20000	1900000

Решение

1. На рабочем листе в диапазон A1:C8 введем данные приведенной таблицы.
2. Включим инструмент **Регрессия**.
3. В открывшемся диалоговом окне **Регрессия** установим параметры:



- *Входной интервал y* – диапазон C1:C8;
- *Входной интервал x* – диапазон A1:B8;
- Флажок *Метки*;
- *Выходной интервал* – адрес D1;
- Флажок *График нормальной вероятности*;
- Флажок *График остатков*.

После щелчка на кнопке ОК в диапазон D1:L21 будет выведен результат регрессионного анализа.

	A	B	C	D	E	F	G	H	I	J	K	L																																				
1	Затраты на рекламу по телевидению	Затраты на рекламу в метро	Объем реализации	Вывод Итогов																																												
2	125850	20000	1850000																																													
3	260500	21000	2500000	<i>Регрессионная статистика</i>																																												
4	150000	22000	1900000	Множественный R	0,987239																																											
5	300000	21000	3000000	R-квадрат	0,974642																																											
6	280000	24000	2600000	Нормированный R-квадрат	0,961962																																											
7	290000	23000	2700000	Стандартная ошибка	90257,48																																											
8	140000	20000	1900000	Наблюдения	7																																											
9	<i>Дисперсионный анализ</i>																																															
10					<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	<i>значимость F</i>																																							
11	Регрессия				2	1,25E+12	6,26E+11	76,869077	0,000643																																							
12	Остаток				4	3,26E+10	8,15E+09																																									
13	Итого				6	1,29E+12																																										
14																																																
15																																																
16	<table border="1"> <thead> <tr> <th></th> <th><i>Коэффициент</i></th> <th><i>Стандартная ошибка</i></th> <th><i>Статистика</i></th> <th><i>P-Значение</i></th> <th><i>Верхние 95%</i></th> <th><i>Нижние 95%</i></th> <th><i>Верхние 95%</i></th> <th><i>Нижние 95%</i></th> </tr> </thead> <tbody> <tr> <td>Y-пересечение</td> <td>2102439</td> <td>586505,7</td> <td>3,584686</td> <td>0,0230709</td> <td>474037,8</td> <td>3730839</td> <td>474037,8</td> <td>373083</td> </tr> <tr> <td>Затраты на рекламу по телевидению</td> <td>6,400415</td> <td>0,585</td> <td>10,94088</td> <td>0,0003964</td> <td>4,776195</td> <td>8,024635</td> <td>4,776195</td> <td>8,02463</td> </tr> <tr> <td>Затраты на рекламу в метро</td> <td>-54,0686</td> <td>30,2618</td> <td>-1,78669</td> <td>0,1485215</td> <td>-138,089</td> <td>29,95166</td> <td>-138,089</td> <td>29,9516</td> </tr> </tbody> </table>													<i>Коэффициент</i>	<i>Стандартная ошибка</i>	<i>Статистика</i>	<i>P-Значение</i>	<i>Верхние 95%</i>	<i>Нижние 95%</i>	<i>Верхние 95%</i>	<i>Нижние 95%</i>	Y-пересечение	2102439	586505,7	3,584686	0,0230709	474037,8	3730839	474037,8	373083	Затраты на рекламу по телевидению	6,400415	0,585	10,94088	0,0003964	4,776195	8,024635	4,776195	8,02463	Затраты на рекламу в метро	-54,0686	30,2618	-1,78669	0,1485215	-138,089	29,95166	-138,089	29,9516
	<i>Коэффициент</i>	<i>Стандартная ошибка</i>	<i>Статистика</i>	<i>P-Значение</i>	<i>Верхние 95%</i>	<i>Нижние 95%</i>	<i>Верхние 95%</i>	<i>Нижние 95%</i>																																								
Y-пересечение	2102439	586505,7	3,584686	0,0230709	474037,8	3730839	474037,8	373083																																								
Затраты на рекламу по телевидению	6,400415	0,585	10,94088	0,0003964	4,776195	8,024635	4,776195	8,02463																																								
Затраты на рекламу в метро	-54,0686	30,2618	-1,78669	0,1485215	-138,089	29,95166	-138,089	29,9516																																								
17	Вывод ОСТАТКА																																															
18	<table border="1"> <thead> <tr> <th><i>Наблюдение</i></th> <th><i>Объем</i></th> <th><i>Остатки</i></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1826560</td> <td>23440,28</td> </tr> <tr> <td>2</td> <td>2634307</td> <td>-134307</td> </tr> <tr> <td>3</td> <td>1872993</td> <td>27007,38</td> </tr> <tr> <td>4</td> <td>2887123</td> <td>112876,6</td> </tr> <tr> <td>5</td> <td>2596909</td> <td>3090,547</td> </tr> <tr> <td>6</td> <td>2714982</td> <td>-14982,2</td> </tr> </tbody> </table>												<i>Наблюдение</i>	<i>Объем</i>	<i>Остатки</i>	1	1826560	23440,28	2	2634307	-134307	3	1872993	27007,38	4	2887123	112876,6	5	2596909	3090,547	6	2714982	-14982,2															
<i>Наблюдение</i>	<i>Объем</i>	<i>Остатки</i>																																														
1	1826560	23440,28																																														
2	2634307	-134307																																														
3	1872993	27007,38																																														
4	2887123	112876,6																																														
5	2596909	3090,547																																														
6	2714982	-14982,2																																														
19																																																
20																																																
21																																																
22																																																
23																																																
24																																																
25																																																
26																																																

Полученные результаты и их интерпретация

- Коэффициент детерминации R-квадрат = 0,974641 (аппроксимация высокая).
- Значимость $F = 0,000643$ ($p < 0,05$ – регрессионная модель значима).
- Y-пересечение $a_0 = 2102438,6$
- $a_1 = 6,4004$ – коэффициент при независимой переменной *Затраты на рекламу по телевидению*.
- $a_2 = -54,068$ – коэффициент при независимой переменной *Затраты на рекламу в метро*.

С учетом полученных данных функциональная зависимость величины прибыли от затрат на рекламу запишется в виде полинома

$$y = 2102438,6 + 6,4004x_1 - 54,068x_2,$$

- где x_1 – величина затрат на рекламу по телевидению;
 x_2 – величина затрат на рекламу в метро.

Используя полученное уравнение регрессии, можно решить задачу оптимизации прибыли или спрогнозировать ожидаемую прибыль при другом распределении средств на рекламу.

Пример 2.17. Организация может истратить на рекламу по телевидению и на рекламу в метро 170 000 руб. Требуется оптимальным образом распределить затраты на различные виды рекламы с целью получения максимальной прибыли.

Решение

Математическая модель

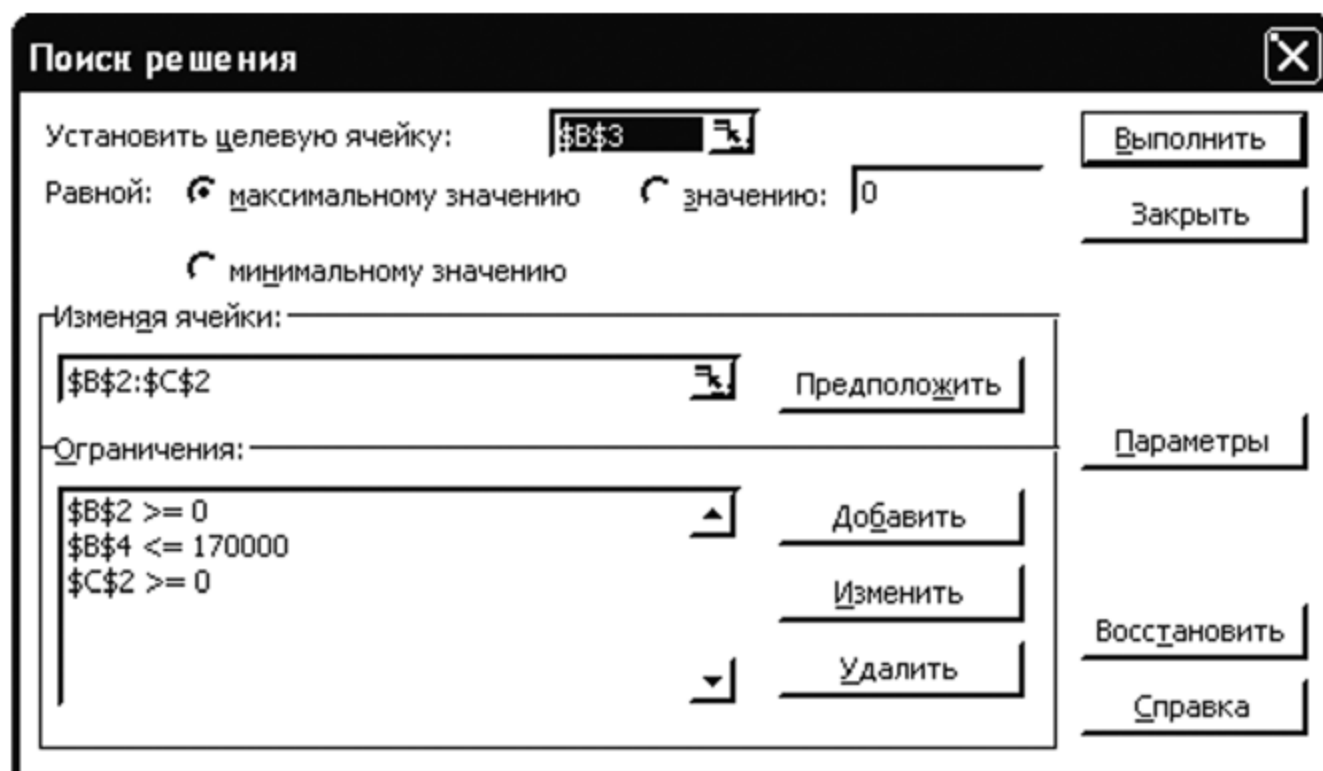
В качестве целевой функции возьмем уравнение регрессии, полученное в примере 2.16:

$$y = 2\,102\,438,6 + 6,4004x_1 - 54,068x_2 \Rightarrow \max.$$

Ограничения: $x_1 + x_2 \leq 170\,000$

$x_1 \geq 0, x_2 \geq 0.$

Используя инструмент **Поиск решения**, решим задачу. Модель и результат решения приведены на рисунке.



	В3	fx = 2102438,6 + 6,4004 *B2-54,068*C2			
	A	B	C	D	E
1		Затраты на рекламу по телевидению	Затраты на рекламу в метро		
2	Решение	170000	0		
3	Ф.Ц.	3190506,6			
4	Ограничение	170000			

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какими способами можно получить массив случайных чисел, распределенных по стандартному нормальному закону?
2. Какими способами можно получить массив случайных чисел, распределенных по нормальному закону, отличающегося от стандартного?
3. Поясните технологии вычисления математического ожидания, дисперсии стандартного отклонения?
4. Какие функции табличного процессора предназначены для вычисления числовых характеристик биномиального распределения?
5. В чем заключается суть численных методов построения выборочной функции распределения?
6. Поясните содержание технологии проверки статистической гипотезы.
7. Назовите основные статистические характеристики и поясните технологии их вычисления.
8. Какими способами можно вычислить доверительный интервал для среднего?
9. Какие инструменты служат для решения задач дисперсионного и корреляционного анализа?
10. Поясните содержание технологии регрессионного анализа.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задание 2.1.

1. Создайте массив из 100 чисел, распределенных равномерно на интервале 1–100.
2. Создайте массив из 100 чисел, распределенных нормально с математическим ожиданием 50, стандартное отклонение — 10, случайное рассеивание — 2.
3. Создайте массив случайных чисел, распределенный по биномиальному закону для параметров: количество испытаний $N = 10$, вероятность благоприятного исхода одного испытания $P = 0,3$.

Задание 2.2.

1. Вычислите математическое ожидание и стандартное отклонение дискретного распределения 0,2; 0,5; 2; 3; 5,1; 8; 2; 3.
2. Найдите 25%-ную квантиль для дискретного распределения 0,2; 0,5; 2; 3; 5,1; 8; 2; 3.
3. Вычислите дисперсию для дискретного распределения 0,2; 0,5; 2; 3; 5,1; 8; 2; 3.

Задание 2.3.

1. Ежедневные расходы на обслуживание и рекламу автомобилей в автосалоне составляют в среднем 120 тыс. ден. ед., а число продаж X автомобилей подчиняется закону распределения.

X	0	1	2	3	4	5	6	7	8	9
P	0,25	0,2	0,1	0,1	0,1	0,1	0,05	0,05	0,025	0,025

Вычислите математическое ожидание ежедневной прибыли при цене автомобиля 150 тыс. ден. ед.

2. Найти дисперсию и среднее квадратичное отклонение случайной величины X , заданной законом распределения.

X	-5	2	3	4
P	0,4	0,3	0,1	0,2

Задание 2.4. Книга издана тиражом 100 тыс. экземпляров. Вероятность брака в экземпляре равна 0,0001. Требуется найти вероятность того, что тираж содержит 5 бракованных книг.

Задание 2.5.

1. Постройте диаграмму стандартного нормального интегрального распределения (стандартное нормальное распределение имеет $M = 0$ и $\sigma = 1$). Используйте функцию НОРМСТРАСП.

2. Вычислите верхний и нижний квартили для нормальной функции плотности вероятности $f(x)$ при $M = 24,3$ и $\sigma = 1,5$.

3. Вычислите вероятность того, что появится случайная величина $x \leq 42$ при нормальном законе распределения вероятностей с $M = 40$ и $\sigma = 1,5$.

4. Вычислите квантиль для $p = 0,908789$ нормального распределения с $M = 20$, $\sigma = 1,5$.

5. Магазин продает мужские костюмы. Распределение спроса по размерам является нормальным с математическим ожиданием $M = 48$ и $\sigma = 2$. Вычислите процент спроса на 50-й размер при условии, что разброс этой величины лежит в интервале 49–51.

Задание 2.6. По данным, приведенным в таблицах, постройте эмпирическую функцию распределения роста.

Рост			Шкала
193	205	177	150
172	190	177	160
185	189	184	170
209	200	162	180
172	157	200	190
175	208	165	200
168	205	176	210
209	157	182	
185	200	197	
197	187	166	
182	176	179	
155	162	185	

Задание 2.7. Определите, лежит ли значение 19 внутри 95%-го доверительного интервала выборки 2, 3, 5, 7, 4, 9, 6, 4, 9, 10, 4, 7, 19.

Задание 2.8. Найдите распределение по абсолютным частотам для следующих результатов тестирования в баллах: 79, 85, 78, 85, 83, 81, 95, 88, 97 для интервалов с границами: 70, 79, 89.

Задание 2.9. Определите, имеется ли взаимосвязь и какая между годовым уровнем инфляции, ставкой рефинансирования и курсом валюты по следующим данным ежегодных наблюдений.

Уровень инфляции (%)	Ставка рефинансирования (%)	Курс (руб./дол.)
84,00	85,00	6,3
45,00	55,00	13
56,00	64,00	22
34,00	40,00	27
23,00	25,00	29
12,00	15,00	31
10,00	12,00	31,5

ТЕХНОЛОГИИ ФИНАНСОВЫХ ВЫЧИСЛЕНИЙ

3.1. ФИНАНСОВЫЕ РАСЧЕТЫ ПО ПРОСТЫМ ПРОЦЕНТАМ

3.1.1. Расчет наращенной суммы

Под наращенной суммой ссуды (депозита, долга) понимается ее первоначальная сумма плюс начисленные на нее к концу срока проценты. Наращенная сумма вычисляется как последний элемент прогрессии, имеющей общий член,

$$P(1+ni), \text{ т.е. } S = P(1+ni),$$

где P — первоначальная сумма;
 n — количество периодов;
 i — ставка за период.

3.1.2. Вычисление количества дней в периоде, заданном начальной и конечной датами

При продолжительности ссуды, не кратной году, необходимо определить, какая часть процента выплачивается кредитору. В табличном процессоре для вычисления количества дней между двумя датами служит функция

$$\text{ДНЕЙ360}(\text{Нач_дата}; \text{Кон_дата}; \text{Метод}).$$

Параметр *Метод* определяет метод расчета: 1 — дата выдачи и дата погашения считается как один день (Европейский); 0 — дата выдачи и дата погашения считается как разные дни (Американский).

3.1.3. Вычисления по простым переменным ставкам

В течение расчетного периода процентные ставки могут дискретно изменяться во времени, при этом они остаются постоянными до следующего дискретного изменения. В этом случае формула для расчета наращенной суммы имеет вид

$$S = P \left(1 + \sum_{t=1}^m n_t i_t \right),$$

где P — первоначальная сумма;

i_t — ставка простых процентов в периоде с номером $t = 1 \dots m$;
 n_t — продолжительность t периода начисления по ставке i_t .

Пример 3.1. В договоре, рассчитанном на год, принята ставка простых процентов на I квартал в размере 8% годовых, а на каждый последующий квартал — на 0,5% меньше, чем в предыдущем. Определить сумму на счете в конце года.

Решение

Применим приведенную выше формулу и реализуем ее на рабочем листе в виде модели, приведенной на рисунке.

C10		fx	
	A	B	C
1	Начальная сумма	2000	
2	Квартал	Ставка	Множитель наращенния
3	1	0,08	=B3/4
4	2	=B3-0,005	=B4/4
5	3	=B4-0,005	=B5/4
6	4	=B5-0,005	=B6/4
7		Итог	=B1*(1+СУММ(C3:C6))
8			

Результат решения приведен на рисунке.

C10		fx	
	A	B	C
1	Начальная сумма	2000	
2	Квартал	Ставка	Множитель наращенния
3	1	8,00%	0,02
4	2	7,50%	0,01875
5	3	7,00%	0,0175
6	4	6,50%	0,01625
7		Итог	2145
8			

3.1.4. Вычисление накопленной суммы при реинвестировании по простым процентам

Сумма с начисленными на нее процентами может быть вновь инвестирована под эту или другую процентную ставку. В случае многократного периодического реинвестирования с применением простой

процентной ставки наращенная сумма для всего срока $N = \sum ni$ вычисляется по формуле

$$S = \prod_{t=1}^m [P(1 + n_t i_t)],$$

где n_t — продолжительности последовательности периодов реинвестирования;
 i_t — ставки, по которым производится реинвестирование.

В табличном процессоре приведенную формулу реализует финансовая функция БС. Эта функция также учитывает равномерные периодические выплаты (если они производятся), которые выполняются в те же периоды, что и реинвестирование.

Пример 3.2. На сумму 100 тыс. ден. ед. начисляется 10% годовых. Проценты простые, точные. Вычислить сумму наращения к концу квартала, если в течение квартала ежемесячно производится реинвестирование.

Решение

Применим приведенную выше формулу и реализуем ее на рабочем листе в виде модели, приведенной на рис. П3.2а.

	А	В	С	Д
1	Начальная сумма	Ставка		
2	100000	0,1		
3	Расчет с реинвестированием			Без реинвестирования
4	Месяц	Сумма		Кoeffициенты наращения
5	1	=\$B\$2*30/360		=\$B\$2*30/360
6	2	=\$B\$2*30/360		=\$B\$2*30/360
7	3	=\$B\$2*30/360		=\$B\$2*30/360
8	Итого	=A2*(1+B5)*(1+B6)*(1+B7)		=A2*(1+СУММ(D5:D7))

Рис. П3.2а

Результат решения приведен на рис. П3.2б.

	А	В	С	Д
1	Начальная сумма	Ставка		
2	100000	0,1		
3	Расчет с реинвестированием			Без реинвестирования
4	Месяц	Сумма		Кoeffициенты наращения
5	1	0,008333333		0,008333333
6	2	0,008333333		0,008333333
7	3	0,008333333		0,008333333
8	Итого	102520,89		102500,00

Рис. П3.2б

Решение этой задачи с использованием функции БС приведено на рис П3.2в.

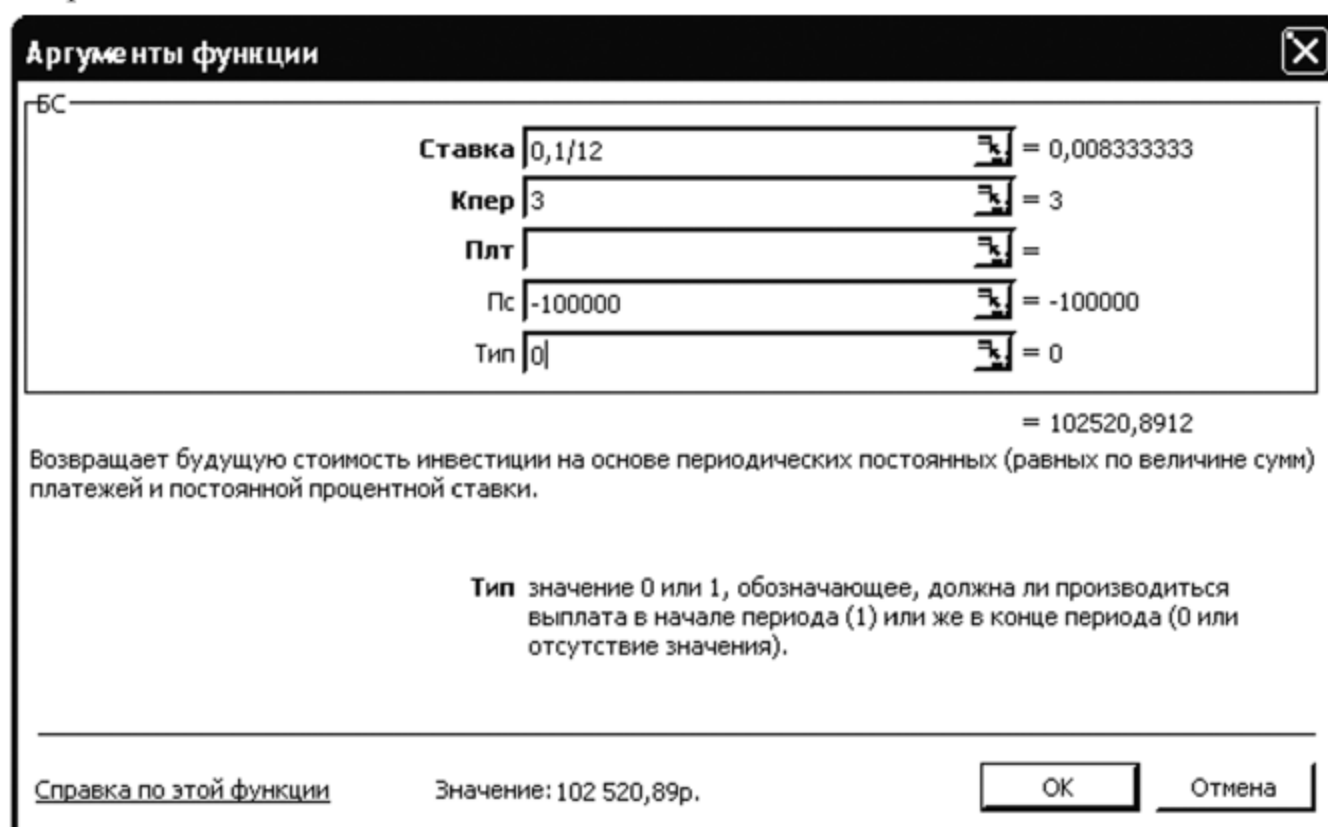


Рис. П3.2в

Пример 3.3. Клиент в начале года получил кредит по годовой ставке 15% (проценты простые) в размере 100 000 руб. Погашение кредита выполнялось неравномерными платежами в конце каждого года в течение 3 лет: 15 000 руб., 20 000 руб., 40 000 руб.

Требуется вычислить остаток долга в конце третьего года.

Решение

В предположении, что начисление по процентам выполняется в конце года без учета текущего платежа, разработаем в табличном процессоре модель вычислений.

	А	В	С
1	Расчет остатка долга по кредиту		
2	Ставка	0,15	
3	Номер года выплаты	Сумма платежа (руб.)	Остаток долга (руб)
4		100000	
5	1	-15000	=B4+B4*\$B\$2+B5
6	2	-20000	=C5+C5*\$B\$2+B6
7	3	-40000	=C6+C6*\$B\$2+B7
8			

Результат вычисления приведен на рисунке.

	А	В	С
1	Расчет остатка долга по кредиту		
2	Ставка	15,00%	
3	Номер года выплаты	Сумма платежа (руб.)	Остаток долга (руб)
4		100000	
5	1	-15000	100000
6	2	-20000	95000
7	3	-40000	69250

Остаток долга по кредиту составляет 69 250 руб.

3.1.5. Дисконтирование по простым процентам

Операция дисконтирования заключается в вычислении исходной суммы P при заданной сумме S , соответствующей концу финансовой операции. Начисления по процентам в виде разности $D = S - P$ называют дисконтом (скидкой).

Дисконтная сумма по простой ставке вычисляется по формуле $P = S / (1 + ni)$.

В табличном процессоре для вычисления дисконтной суммы предназначена функция ПС.

Пример 3.4. Определить сумму вклада, которую нужно положить в банк сроком на два месяца под 10% годовых, чтобы к концу срока получить 101 667 руб. Расчеты выполнить для случая простых процентов.

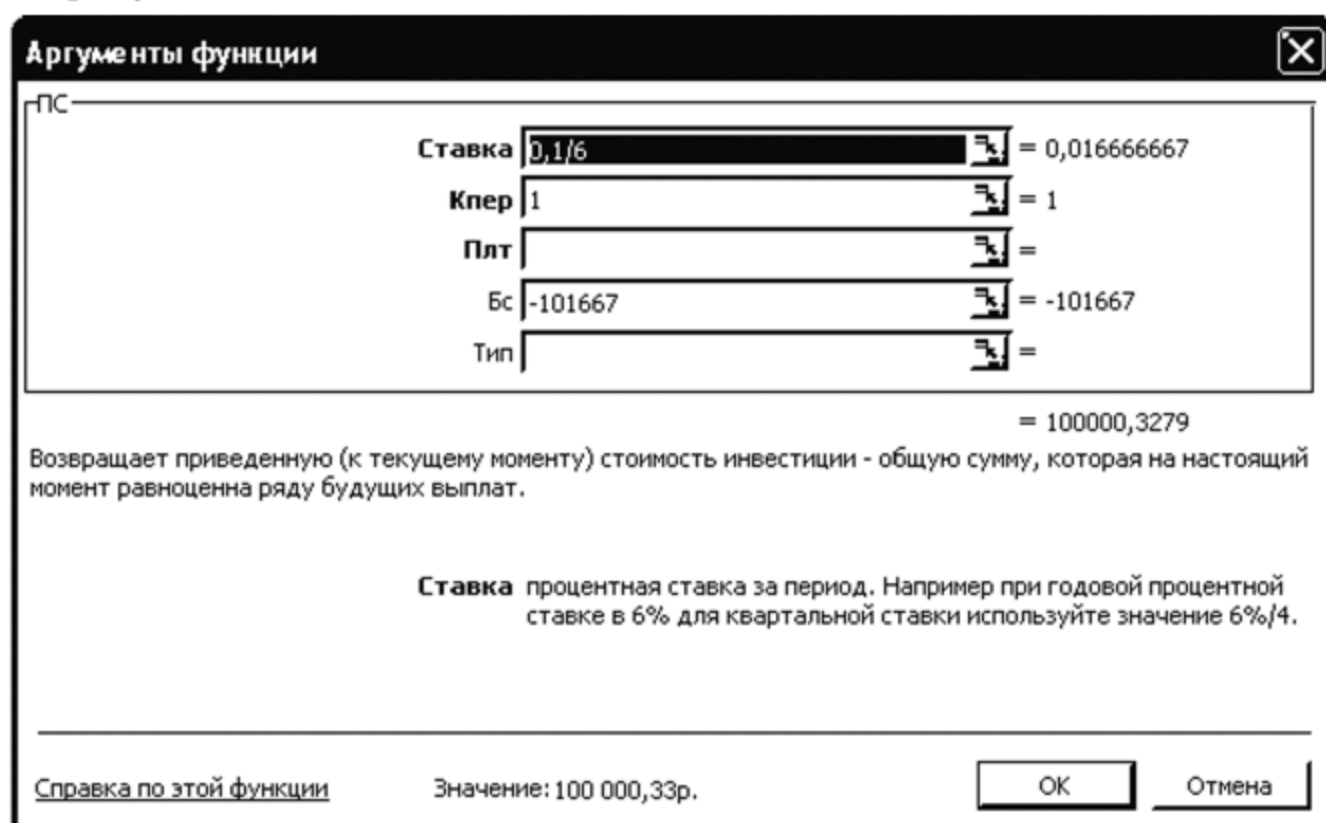
Решение

Решение и полученный результат приведен на рисунке.

	А	В	С
1	Дисконтирование		
2	Простые проценты		
3		Накопленная сумма	Ставка
4		101667	0,1
5	Вклад	=B4/(1+2*C4/12)	

	А	В	С
1	Дисконтирование		
2	Простые проценты		
3		Накопленная сумма	Ставка
4		101667	10,00%
5	Вклад	100000,33	
6			

Решение этой задачи с использованием функции ПС приведено на рисунке.



Пример 3.5. Платежное обязательство уплатить через 60 дней 200 000 руб. с процентами, начисляемыми по ставке простых процентов $i = 15\%$ годовых, было учтено за 10 дней до срока погашения по учетной ставке 10%. Вычислить сумму, получаемую при учете (число дней в году 365).

Решение

Используем приведенную выше формулу и реализуем ее на рабочем листе в виде модели, приведенной на рисунке.

	A	B	C	D	E	F
1	Вычисление суммы, получаемой при учете					
2	Срок учета до погашения (дней)	Сумма	Срок (дней)	Ставка	Учетная ставка	Сумма к возрату
3	10	200000	60	0,15	0,12	=B3*(1+(C3-A3)*D3/365)*(1+E3*A3/365)
4						
5						

Результат решения приведен на рисунке.

	A	B	C	D	E	F
1	Вычисление суммы, получаемой при учете					
2	Срок учета до погашения	Сумма	Срок (дней)	Ставка	Учетная ставка	Сумма к возрату
3	10	200000	60	0,15	0,12	204780,63
4						
5						

3.2. ФИНАНСОВЫЕ РАСЧЕТЫ ПО СЛОЖНЫМ ПРОЦЕНТАМ

3.2.1. Вычисление наращенной суммы

Формула наращенной суммы для сложных процентов имеет вид

$$S = P(1 + i)^n,$$

где P — сумма инвестиций;
 S — наращенная сумма;
 i — годовая ставка сложных процентов;
 n — срок ссуды (количество периодов).

Если количество начислений в году несколько, то для расчетов применяется формула

$$S = P(1 + j/m)^N,$$

где N — число периодов начисления, $N = mn$;
 j — номинальная годовая ставка сложных процентов;
 m — число начислений в году;
 n — количество лет.

Для вычисления наращенной суммы в табличном процессоре есть специальная финансовая функция БС (*Ставка; Кпер; Плт; Пс; Тип*).

Параметры функции:

- *Ставка* — ставка за период;
- *Кпер* — количество периодов;
- *Плт* — величина постоянного платежа в каждом периоде;
- *Пс* — сумма инвестиции, указывается со знаком минус;
- *Тип* — значение 1 указывает, что расчет производится на начало периода, значение 0 — на конец периода.

Пример 3.6. Исходная сумма кредита 100 тыс. ден. ед. Ставка 30% годовых. Вычислить наращенную сумму по простым и сложным процентам за 1,5 года.

Решение

Модель и результат решения приведены на рисунке.

Используя приведенную выше формулу, можно вычислить величину годовой ставки при известных значениях S , P и n , а также количество периодов n при известных S , P и i . Для этого формулу нужно представить в виде уравнения, например $P(1 + i)^n - S = 0$, и решать его относительно n или i , используя соответствующие инструменты табличного процессора или функции системы Matlab.

Для вычисления ставки при заданных значениях S , P и n в табличном процессоре есть функция СТАВКА, которая имеет синтаксис

$$\text{СТАВКА}(\text{Кпер}; \text{Плт}; \text{Пс}; \text{Бс}; \text{Тип}),$$

	А	В	С
1	Вычисление наращенной суммы		
2	Исходные данные		
3	Р	і	п
4	100000	0,3	1,5
5	Результаты вычислений		
6	Простые проценты	=A4*(1+B4*C4)	
7	Сложные проценты	=A4*(1+B4)^C4	
8	Функция БС	=БС(В4;С4;;-А4)	
а			

Результаты вычислений

	А	В	С
1	Вычисление наращенной суммы		
2	Исходные данные		
3	Р	і	п
4	100000	30,00%	1,5
5	Результаты вычислений		
6	Простые проценты	145000	
7	Сложные проценты	148222,8053	
8	Функция БС	148 222,81р.	
а			

а для вычисления количества периодов – функция КПЕР
 КПЕР(*Ставка; Плт; Пс; Бс; Тип*).

3.2.2. Расчет номинальной и эффективной ставки процентов

Эффективная процентная ставка показывает, какая годовая ставка сложных процентов дает тот же финансовый результат, что и m – разовое наращение по ставке j/m .

Если проценты капитализируются m раз в год каждый раз со ставкой j/m , то связь между эффективной и номинальной ставками выражается соотношением

$$i_s = (1 + j/m)^m - 1.$$

Для вычисления эффективной ставки в табличном процессоре служит финансовая функция ЭФФЕКТ(*Номинальная ставка; Количество периодов*).

Для вычисления номинальной ставки при заданной эффективной служит финансовая функция НОМИНАЛ(*Эффективная ставка; Количество периодов*).

3.2.3. Дисконтирование по сложной ставке процентов

Дисконтная сумма в случае сложной процентной ставки вычисляется по формуле

$$P = S/(1 + i_t)^n,$$

где i_t — ставка за учетный период;

n — число периодов.

Если дисконтирование применяют m раз в году, то используют номинальную учетную ставку f . Процесс дисконтирования по этой сложной учетной ставке описывается формулой

$$P = S/(1 + f/m)^N,$$

где $N = nm$ — общее число периодов дисконтирования.

3.2.4. Расчет стоимости ценных бумаг

Для расчета стоимости векселя в табличном процессоре используется финансовая функция ЦЕНАСКИДКА, синтаксис которой имеет вид

ЦЕНАСКИДКА(*Дата_соглашения*; *Дата_вст.в_силу*; *Скидка*;
Погашение; *Базис*),

где *Дата соглашения* — дата соглашения для ценных бумаг, выраженная как дата в числовом формате (дата покупки, учета);

Дата_вст.в_силу — дата вступления в силу ценных бумаг, выраженная как дата в числовом формате (дата погашения);

Скидка — норма скидки для ценных бумаг (годовая номинальная учетная ставка);

Погашение — цена при погашении (за 100 руб. нарицательной стоимости ценных бумаг);

Базис — тип используемого способа вычисления дня.

Тип	Длительность	
	месяц	год
0	30	360
1	Фактическая	Фактический
2	Фактическая	360
3	Фактическая	365
4	Европейская 30	360

Функция ЦЕНАСКИДКА выполняет расчет по формуле

$$P = S(1 - dt),$$

где $dt = dT$ — учетная ставка за период времени T до погашения долга (в долях года) равное количеству дней от даты продажи до даты погашения, деленному на количество дней в году.

Дату следует выражать в числовом формате как номер дня в соответствующей системе с помощью функции ДАТА.

Пример 3.7. Владелец векселя на сумму 10 тыс. руб. учел его в банке за два месяца до срока погашения по годовой учетной ставке 20%. Требуется определить выкупную (учетную) стоимость векселя, т.е. сумму, которую получит владелец векселя.

Решение

Для определенности положим, что вексель учтен 01.01.2007, а срок погашения наступает 01.03.2007. Тогда формула для вычисления, использующая функцию ЦЕНАСКИДКА, будет иметь вид

$$=ЦЕНАСКИДКА(ДАТА(2007;1;1);ДАТА(2007;3;1);0,2;10;0).$$

Формула для расчета без использования функции ЦЕНАСКИДКА имеет вид

$$=10*(1-0,2/360*(ДАТА(2007;3;1)-ДАТА(2007;1;1))).$$

Результат вычисления выкупной стоимости векселя 9,66667 тыс. руб.

Для расчета простой учетной ставки в табличном процессоре служит функция ИНОРМА

ИНОРМА (*Дата_согл; Дата_вст_в_силу; Инвестиция; Погашение; Базис*),

где *Дата_согл* — дата соглашения для ценных бумаг, выраженная как дата в числовом формате;

Дата_вст_в_силу — дата вступления в силу для ценных бумаг, выраженная как дата в числовом формате;

Инвестиция — объем инвестиции в ценные бумаги;

Погашение — объем средств, которые должны быть получены на дату вступления в силу ценных бумаг;

Базис — способ вычисления дня.

Пример 3.8. Требуется определить простую ставку процентов для контракта сроком на четыре месяца, если сумма долга равна 100 тыс. руб., а сумма, подлежащая возврату, — 110 тыс. руб.

Решение

Для определенности положим, что дата соглашения 01.03.2005, а дата вступления в силу 01.07.2005. Тогда формула для вычисления в Excel для условия задачи будет иметь вид

$$=ИНОРМА(ДАТА(2005;3;1);ДАТА(2005;7;1);100;110).$$

Результат решения равен 0,3.

Эту задачу в табличном процессоре можно также решить с помощью функции

ДОХОДПОГАШ(*Дата_согл*; *Дата_вст_в_силу*; *Дата_выпуска*; *Ставка*; *Цена*).

Простую учетную ставку можно вычислить с помощью функции **СКИДКА**(*Дата_согл*; *Дата_вст_в_силу*; *Цена*; *Погашение*; *Базис*),

где *Дата_согл* — дата соглашения для ценных бумаг, выраженная как дата в числовом формате;

Дата_вст_в_силу — дата вступления в силу для ценных бумаг, выраженная как дата в числовом формате;

Цена — цена ценных бумаг за 100 руб. в нарицательной стоимости;

Погашение — выкупная цена ценных бумаг за 100 руб. в нарицательной стоимости;

Базис — способ вычисления дня.

Пример 3.9. Вексель на сумму 110 тыс. руб. выдан сроком на 4 месяца. Требуется вычислить простую учетную ставку, если полученная под вексель сумма равна 100 тыс. руб.

Решение

Для определенности положим, что дата соглашения 01.03.2007, дата погашения (вступления в силу) 01.07.2007. Тогда формула для вычисления будет иметь вид

=СКИДКА(ДАТА(2007;3;1);ДАТА(2007;7;1);100;110).

Результат вычисления равен 0,2727.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Поясните назначение функции БС табличного процессора.
2. Каким образом можно вычислить количество дней между двумя датами в табличном процессоре?
3. Какую функцию табличного процессора можно применить для вычисления дисконтной суммы?

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задание 3.1. Вычислите сумму, причитающуюся к возврату, если сумма кредита составляет 200 тыс. ден. ед., срок 6 месяцев при ставке простых процентов 12% годовых.

Задание 3.2.

1. Вычислите количество дней финансового периода между 02.06.2006 и 21.01.2007 в соответствии с Европейским стандартом.

2. Вычислите сумму, причитающуюся к возврату, если сумма кредита составляет 200 тыс. ден. ед., кредит взят 01.02.2005, срок погашения кредита 01.08.2005 при ставке простых процентов 12% годовых.

Задание 3.3. Размер ссуды, предоставленной на 28 месяцев, равен 20 млн. Номинальная ставка равна 60% годовых, начисление процентов ежеквартальное. Требуется вычислить наращенную сумму.

Задание 3.4.

1. Банк начисляет сложные проценты на вклад исходя из годовой номинальной ставки 12%. Вычислить эффективную годовую процентную ставку при ежемесячной капитализации процентов. Решить задачу с использованием приведенной формулы и функции ЭФФЕКТ. Сравнить полученные результаты.

2. Вычислить, какой должна быть номинальная ставка при ежеквартальном начислении процентов, чтобы обеспечить эффективную ставку 12% годовых. Решить задачу с использованием приведенной формулы и функции ЭФФЕКТ. Сравнить полученные результаты.

Задание 3.5.

1. Требуется вычислить текущую стоимость суммы $S = 100$ тыс. руб., подлежащей уплате через 3 года, при использовании сложной процентной ставки 30% годовых.

2. Кредит выдается на полгода по годовой учетной ставке 30%. Определить сумму, получаемую заемщиком, и величину дисконта, если сумма займа равна 50 тыс. руб.

ТЕХНОЛОГИИ АНАЛИЗА ДАННЫХ, ПРЕДСТАВЛЕННЫХ ТАБЛИЧНО

4.1. ОРГАНИЗАЦИЯ МОДЕЛИ ДАННЫХ В ВИДЕ СПИСКОВ

Для организации данных на рабочем листе Excel часто применяются модели, называемые списками. Список — это таблица, состоящая из именованных столбцов. Строку рабочего листа, содержащую названия столбцов списка, называют строкой заголовков. Ячейки таблицы одного столбца содержат данные только одного типа. Список можно интерпретировать как простую однотабличную базу данных (БД), где строки — это записи, а столбцы — поля. При выполнении большинства операций с БД (например, поиск и сортировка, подведение итогов и т.п.) Excel воспринимает список как базу данных.

Эффективная работа со списками возможна при выполнении некоторых правил:

- одна или две верхние строки должны содержать заголовки, раскрывающие смысл расположенного ниже столбца;
- каждый столбец должен содержать информацию одного типа;
- в списках не должно быть пустых строк и столбцов;
- если нет возможности расположить список на отдельном листе, то нужно отделить его как минимум одним пустым столбцом и одной пустой строкой;
- для длинных списков рекомендуется закрепить заголовки, чтобы они не исчезали при перемещении таблицы по экрану.

На рисунке 4.1 в качестве примера приведен фрагмент списка, в котором учитываются операции банка по счетам клиентов.

4.2. ПРИМЕНЕНИЕ ФОРМЫ ПРИ РАБОТЕ СО СПИСКАМИ

При работе с небольшими списками добавление в список нескольких строк или редактирование можно сделать непосредственно на рабочем листе. В больших списках это проще выполнить с помощью специального диалогового окна, называемого **Форма**. С помощью этого диалогового окна можно быстро найти нужную строку в таблице, отредактировать данные или удалить их, сделать добавления.

	A	B	C	D	E	F
1	Таблица 1. Операции банка по счетам клиентов					
2	№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
3	1	1	Петров	03.01.02	1000	
4	2	2	Разумовская	03.01.02	2000	
5	3	1	Петров	03.01.02		500
6	4	2	Разумовская	03.01.02		1000
7	5	3	Наумов	03.01.02	1500	
8	6	4	Крутиков	03.01.02	1300	
9	7	5	Павлова	04.01.02	1050	

Рис. 4.1. Пример списка в Excel

Для того чтобы включить **Форму**, нужно выполнить операции:

- установить курсор в любую ячейку таблицы (списка);
- выполнить команду меню **Данные/Форма**, которая включает форму для вывода или редактирования (рис. 4.2).

Рис. 4.2. Диалоговое окно Форма

В полях формы отображаются данные той строки списка, в ячейке которой находился курсор при включении формы.

Для добавления новой строки в список нужно щелкнуть на кнопке *Добавить* — поля формы будут очищены и в них можно вводить новые данные. После ввода данных новой строки списка можно продолжить ввод новой строки, для этого следует щелкнуть на кнопке *Добавить*. Чтобы окончить ввод данных в список нужно щелкнуть на кнопке *Закрыть*.

Чтобы, применяя форму, выполнить переход от одной строки списка к другой, можно щелкнуть на кнопке *Далее (Назад)* или использовать вертикальную линейку прокрутки, расположенную в форме.

Для удаления строки из списка нужно перейти к ней и щелкнуть на кнопке *Удалить*.

Применяя форму, можно быстро найти нужную строку в списке. Для этого нужно выполнить операции:

- щелкнуть на кнопке формы *Критерии*;
- в поля формы, значения которых являются условиями для поиска, ввести соответствующие данные (рис. 4.3);
- щелкнуть на кнопке *Далее (Назад)* — в полях формы будут отображаться данные списка, удовлетворяющие условиям поиска.

The image shows a window titled "Операции" (Operations) with a close button in the top right corner. The window contains several input fields and a list of buttons. The input fields are labeled: "№ операции:" (Operation No.), "№ счета:" (Account No.), "Фамилия клиента:" (Client Surname), "Дата операции:" (Operation Date), "Приход:" (Income), and "Расход:" (Expense). The "№ счета:" field contains the value "12" and the "Дата операции:" field contains "05.01.2008". To the right of the input fields is a vertical list of buttons: "Критерии" (Criteria), "Добавить" (Add), "Очистить" (Clear), "Вернуть" (Return), "Назад" (Back), "Далее" (Next), "Правка" (Edit), and "Закрыть" (Close). There are also small up and down arrow icons on the right side of the input fields.

Рис. 4.3. Поиск строки в списке по заданному критерию

4.3. АНАЛИЗ ДАННЫХ НА ОСНОВЕ ИХ СОРТИРОВКИ

4.3.1. Сортировка строк списка

Табличный процессор предоставляет различные способы сортировки списков, размещенных на рабочем листе. Можно сортировать строки или столбцы в возрастающем или убывающем порядке. При сортировке строк изменяется порядок их следования, а порядок следования столбцов остается неизменным. Строка заголовков обычно не сортируется.

Чтобы сортировать список, нужно выполнить операции:

– установить курсор в любую ячейку списка;

– выполнить команду меню **Данные/Сортировка**. Табличный процессор предварительно определит размер списка, определит строку заголовка, исключаемую из сортировки, и выведет на экран диалоговое окно **Сортировка диапазона** (рис. 4.4):

– выбрать в списках окна **Сортировка диапазонов** нужные названия столбцов списка и установить соответствующие переключатели, определяющие порядок сортировки;

– щелкнуть на кнопке **ОК** – список будет отсортирован в соответствии с заданными параметрами;

– приведения списка в исходное состояние нужно выполнить команду меню **Правка/Отменить/Сортировка**.

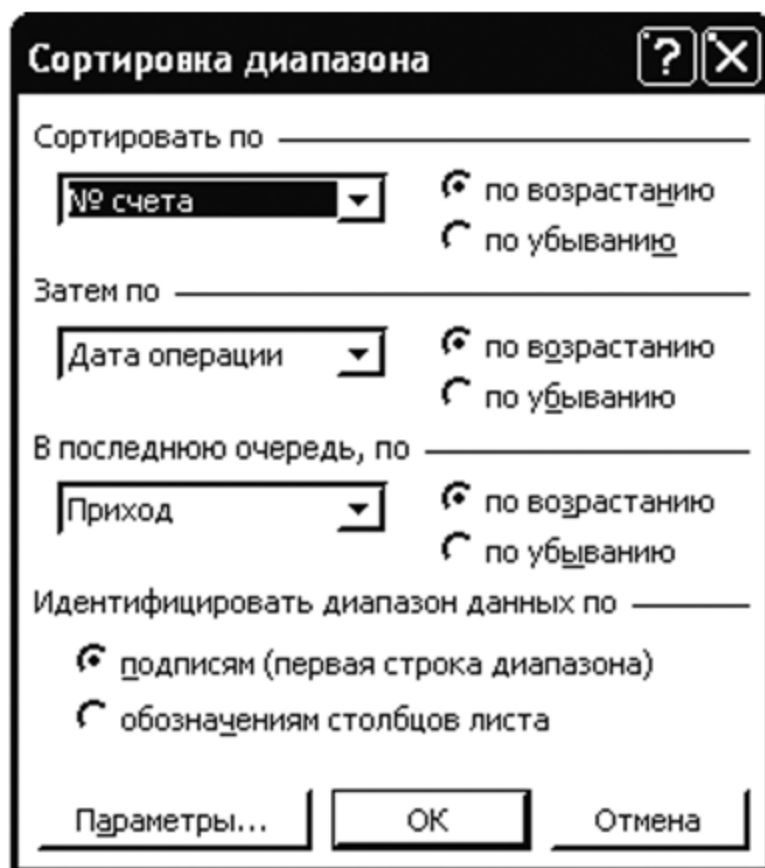


Рис. 4.4. Диалоговое окно **Сортировка диапазона**

Сортировка может выполняться по трем ключам — сначала по первому, указанному пользователем в поле *Сортировать по*, затем в пределах строк, у которых данные в этих полях имеют одинаковое значение, будет произведена сортировка по второму ключу, установленному в поле *Затем по*. Сортировка по третьему ключу производится только для строк с одинаковыми ключевыми первым и вторым полями.

Для каждого ключа сортировки можно установить переключатель *По возрастанию* или *По убыванию*.

Чтобы заголовки списков не включались в сортировку, нужно в группе *Идентифицировать данные по* установить переключатель *Подписям*.

4.3.2. Сортировка строк диапазона

Для сортировки не всего списка, а только его части нужно выполнить операции:

- выделить диапазон, который нужно сортировать;
- выполнить команду меню **Данные/Сортировка** — откроется диалоговое окно **Сортировка диапазона** (рис 4.5);
- установить нужные параметры сортировки и щелкнуть на кнопке ОК.



Рис. 4.5. Сортировка части списка

4.3.3. Сортировка столбцов списка

При решении некоторых задач бывает необходимо сортировать список не по строкам, а по столбцам. Чтобы сортировать список по столбцам, следует выполнить операции:

— в диалоговом окне **Сортировка диапазона** щелкнуть на кнопке *Параметры*. Откроется диалоговое окно **Параметры сортировки** (рис. 4.6);

— в диалоговом окне **Параметры сортировки** установить переключатель *Столбцы диапазона* и щелкнуть на кнопке **ОК**;

— в диалоговом окне **Сортировка диапазона** в качестве ключей сортировки указать соответствующие строки (рис. 4.7).

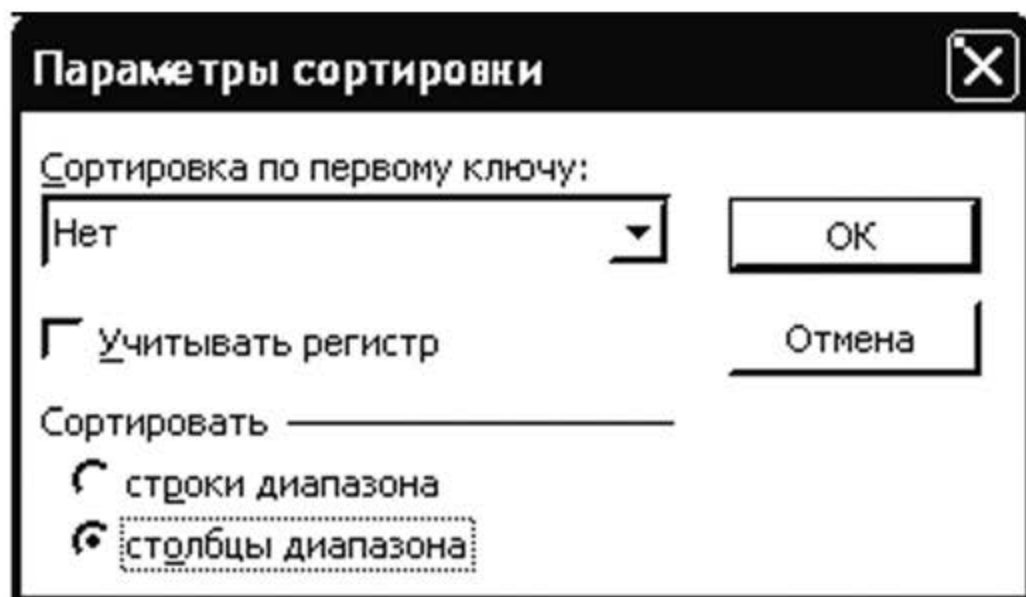


Рис. 4.6. Диалоговое окно **Параметры сортировки**



Рис. 4.7. Сортировка списка по столбцам

4.3.4. Пользовательский порядок сортировки

Иногда возникает необходимость применения нестандартного порядка сортировки. Например, сортировку не в алфавитном порядке, а в виде списка: Группа, Фамилия, Имя. Для создания пользовательского порядка сортировки нужно выполнить следующие действия:

- выполнить команду меню **Сервис/Параметры**;
- щелкнуть на закладке *Списки* диалогового окна **Параметры**;
- в поле *Списки* выбрать *Новый список*;
- в поле *Элементы списка* ввести элементы нового пользовательского списка, отделяя их друг от друга запятыми или записывая каждый элемент с новой строки (рис. 4.8);

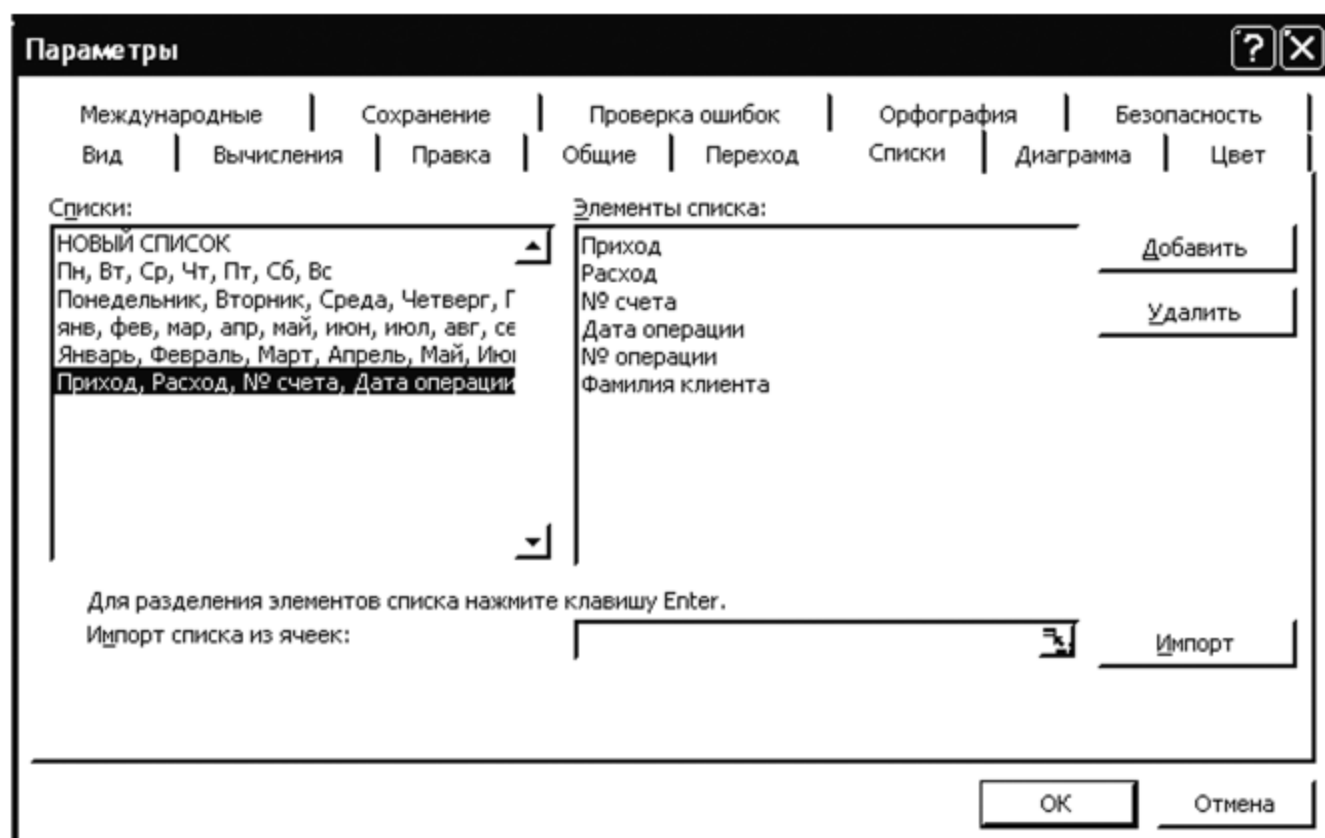


Рис. 4.8. Вкладка *Списки* диалогового окна **Параметры**

- щелкнуть на кнопке *Добавить*;
- после появления пользовательского списка в поле *Списки* щелкнуть на кнопке **ОК**;
- выделить диапазон списка, который нужно сортировать, выполнить команду меню **Данные/Сортировка** и в диалоговом окне **Сортировка диапазона** щелкнуть на кнопке *Параметры*;
- в диалоговом окне **Параметры сортировки** выбрать нужный список, устанавливающий порядок сортировки и щелкнуть на кнопке **ОК** (рис. 4.9, 4.10);

— в открывшемся диалоговом окне **Сортировка диапазона** щелкнуть на кнопке **ОК** — выполнится сортировка списка (рис. 4.11).

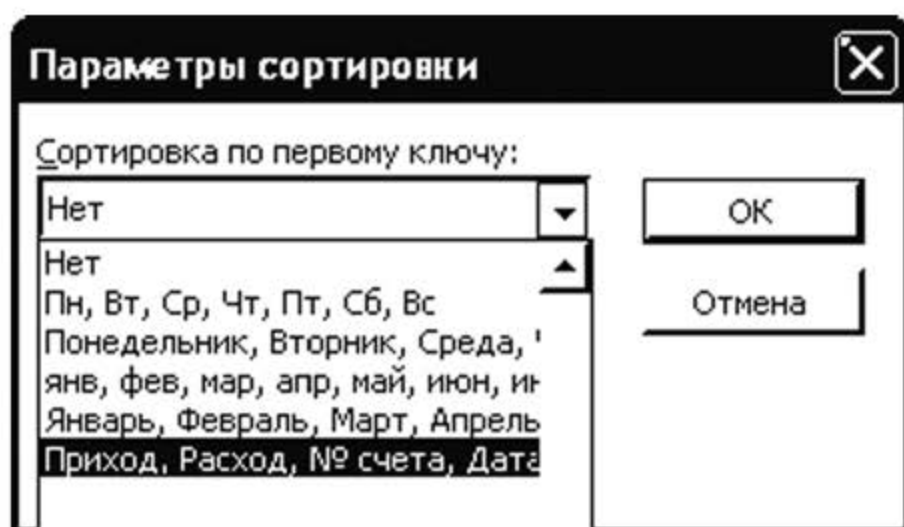


Рис. 4.9. Диалоговое окно **Параметры сортировки**

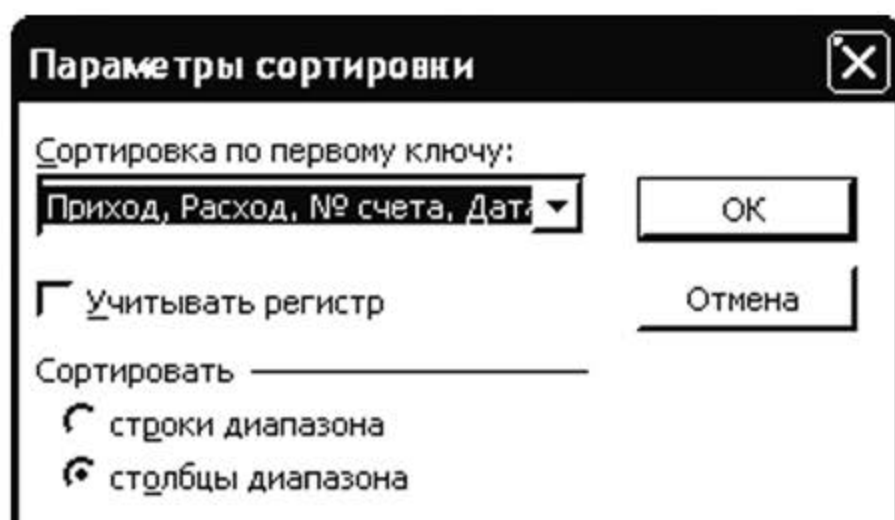


Рис. 4.10. Выбор порядка сортировки

	А	В	С	Д	Е	Ф
1	Таблица 1. Операции банка по счетам клиентов					
2	Приход	Расход	№ счета	Дата операции	№ операции	Фамилия клиента
3	1000		1	03.01.02	1	Петров
4	2000		2	03.01.02	2	Разумовская
5		500	1	03.01.02	3	Петров
6		1000	2	03.01.02	4	Разумовская
7	1500		3	03.01.02	5	Наумов
8	1300		4	03.01.02	6	Крутиков
9	1050		5	04.01.02	7	Павлова
10	1100		6	04.01.02	8	Круглова
11	1200		7	04.01.02	9	Наумов

Рис. 4.11. Результат сортировки списка

4.4. ФИЛЬТРАЦИЯ СПИСКОВ

Под фильтрацией списков понимается выделение из всего списка только тех строк, значения полей которых удовлетворяют заданным критериям. Табличный процессор имеет несколько инструментов с различными функциональными возможностями, которые позволяют выполнять фильтрацию несколькими способами.

4.4.1. Фильтрация списков с помощью инструмента Автофильтр

Отфильтровать список — означает оставить в таблице только строки, удовлетворяющие заданным условиям. В табличном процессоре есть два инструмента — **Автофильтр**, предназначенный для фильтрации списков по простым условиям, и **Расширенный фильтр**, выполняющий фильтрацию для усложненных условий.

Чтобы применить инструмент **Автофильтр**, нужно выполнить операции:

- установить курсор в любую ячейку списка;
- выполнить команду меню **Данные/Фильтр/Автофильтр** (рис. 4.12), в фильтруемом списке рядом с названиями столбцов появятся кнопки со стрелками раскрывающихся списков (рис. 4.13);



Рис. 4.12. Запуск инструмента **Автофильтр**

— щелкнуть на стрелке столбца, данные которого являются условием фильтрации, и в раскрывшемся списке выбрать значение для условия фильтрации (рис. 4.14) — выполнится фильтрация по заданному условию (рис. 4.15).

На рисунке 4.15 показан результат фильтрации списка по номеру счета со значением 2.

	A	B	C	D	E	F
1	Таблица 1. Операции банка по счетам клиентов					
2	№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
3	1	1	Петров	03.01.02	1000	
4	2	2	Разумовская	03.01.02	2000	
5	3	1	Петров	03.01.02		500
6	4	2	Разумовская	03.01.02		1000
7	5	3	Наумов	03.01.02	1500	
8	6	4	Крутиков	03.01.02	1300	

Рис. 4.13. Кнопки раскрывающихся списков

	A	B	C	D	E	F
1	Таблица 1. Операции банка по счетам клиентов					
2	№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
3	(Все)		Петров	03.01.02	1000	
4	(Первые 10...)		Разумовская	03.01.02	2000	
5	(Условие...)		Петров	03.01.02		500
6	1		Разумовская	03.01.02		1000
7	2		Наумов	03.01.02	1500	
8	3		Крутиков	03.01.02	1300	
9	4		Павлова	04.01.02	1050	
10	5		Круглова	04.01.02	1100	
11	6		Наумов	04.01.02	1200	

Рис. 4.14. Выбор условия фильтрации

	A	B	C	D	E	F
1	Таблица 1. Операции банка по счетам клиентов					
2	№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
4	2	2	Разумовская	03.01.02	2000	
6	4	2	Разумовская	03.01.02		1000
41	38	2	Разумовская	10.01.02	120	
54	51	2	Разумовская	14.01.02		100
75	71	2	Разумовская	21.01.02		3550
94	89	2	Разумовская	25.01.02	4500	

Рис. 4.15. Результат фильтрации списка

Последовательно фильтруя список по значениям разных столбцов, можно получить результат по нескольким условиям, которые удовлетворяются совместно.

Элемент *Первые 10* в списке условий (рис. 4.14) позволяет отфильтровать заданное число (заданный процент) наибольших или наименьших элементов в списке. Выбор этого условия выводит диалоговое окно *Наложение условий по списку* (рис. 4.16).

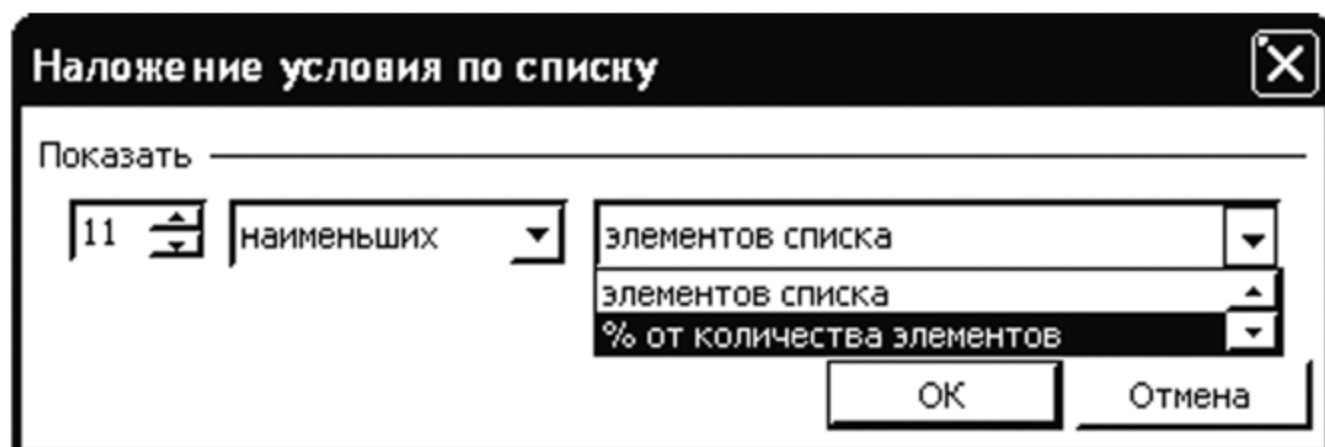


Рис. 4.16. Диалоговое окно *Наложение условий по списку*

Результат фильтрации первых 11 значений приведен на рис. 4.17.

	A	B	C	D	E	F
1	Таблица 1. Операции банка по счетам клиентов					
2	№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
3	1	1	Петров	03.01.02	1000	
4	2	2	Разумовская	03.01.02	2000	
5	3	1	Петров	03.01.02		500
6	4	2	Разумовская	03.01.02		1000
7	5	3	Наумов	03.01.02	1500	
8	6	4	Крутиков	03.01.02	1300	
9	7	5	Павлова	04.01.02	1050	
10	8	6	Круглова	04.01.02	1100	
11	9	7	Наумов	04.01.02	1200	
12	10	8	Сидоров	04.01.02	5000	
13	11	9	Охрименко	04.01.02	1500	

Рис. 4.17. Результат фильтрации первых 11 значений

Пример 4.1. Требуется определить, какова максимальная сумма прихода по счету 10.

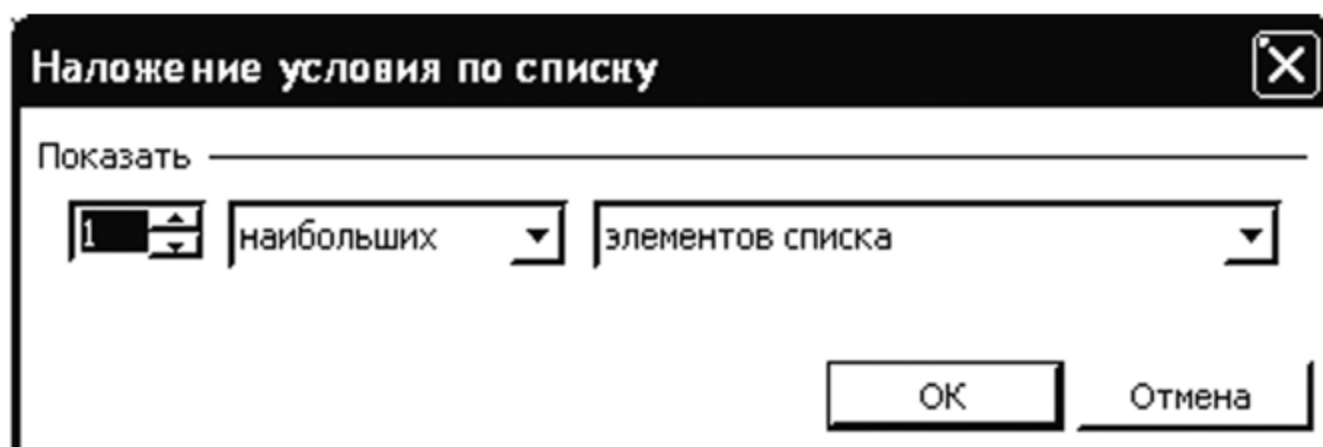
Решение

Выполним команду меню **Данные/Фильтр/Автофильтр**.

Раскроем список условий в столбце с названием *№ счета* и выберем в нем значение 10.

Раскроем список условий в столбце с названием *Приход* и выберем в нем элемент *Первые 10*.

В раскрывшемся окне диалога **Наложение условия по списку** в группе *Показать* установим значения 1 и наибольших.



Щелкнем на кнопке ОК. Результат фильтрации приведен на рисунке.

	A	B	C	D	E	F
1	Таблица 1. Операции банка по счетам клиентов					
2	№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
36	34	10		08.01.02	20000	
175						

4.4.2. Фильтрация с применением инструмента Пользовательский автофильтр

В инструменте **Автофильтр** в качестве условий фильтрации применяются простые условия равенства по значению в столбце. Однако при решении задач бывает необходимо использовать условия больше (меньше), не равно.

Для решения таких задач в случае несложных условий можно применить инструмент **Пользовательский автофильтр**. Для того что-

бы применить инструмент **Пользовательский автофильтр**, нужно выполнить операции:

- включить автофильтр;
- щелкнуть на стрелке столбца, данные которого являются условием фильтрации, и в раскрывшемся списке выбрать элемент *Условие* — откроется диалоговое окно **Пользовательский автофильтр** (рис. 4.18);
- установить в полях диалогового окна значения условий для фильтрации;
- щелкнуть на кнопке ОК.

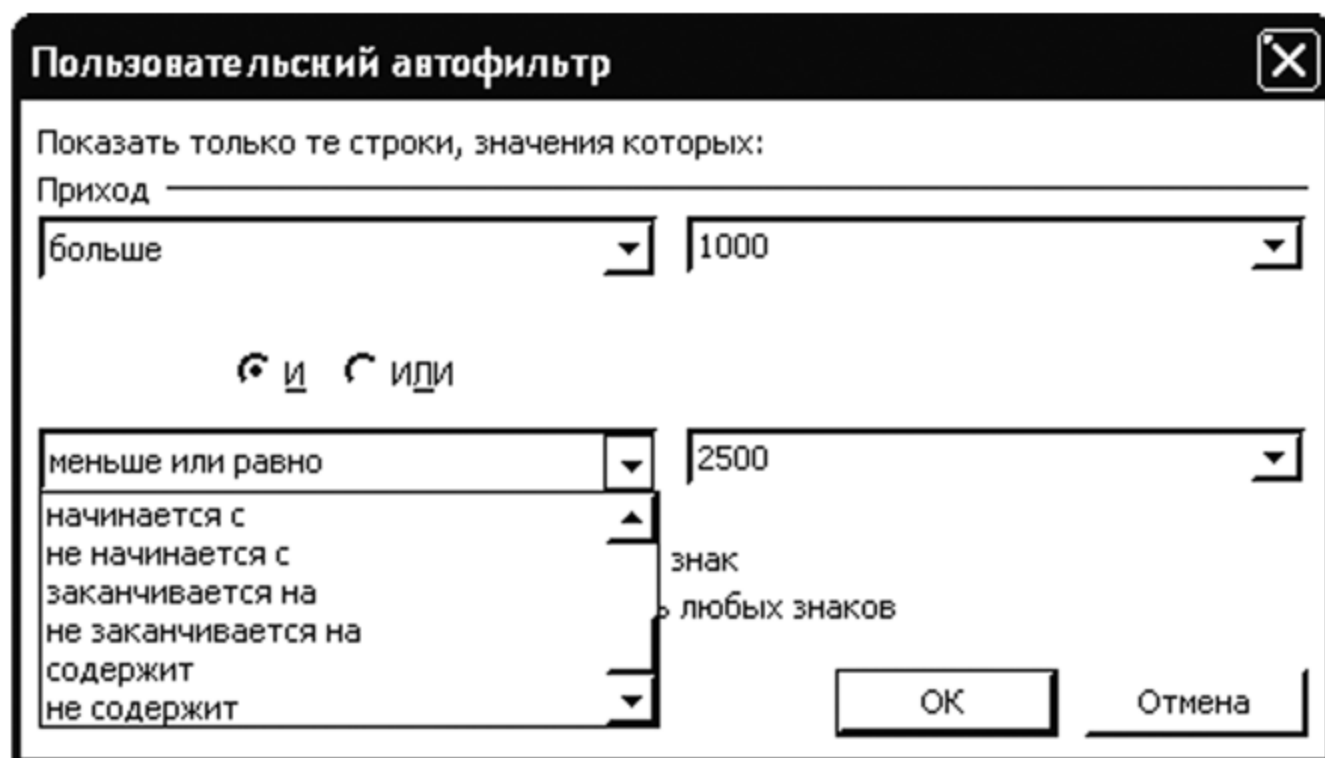


Рис. 4.18. Диалоговое окно **Пользовательский автофильтр**

Пример 4.2. Требуется определить, сколько было приходных операций с суммами величиной от 1000 до 2500 рублей включительно по счету 10.

Решение

Включим инструмент **Автофильтр**.

Щелкнем на стрелке в столбце с названием *№ счета* и в раскрывшемся списке выберем элемент со значением 10.

Раскроем список в столбце с названием *Приход* и выберем в нем элемент *Условие* — откроется диалоговое окно **Пользовательский автофильтр**.

В диалоговом окне **Пользовательский автофильтр** в верхнем левом списке выберем элемент *Больше или равно*, а в поле правого списка введем число 1000, в нижнем левом списке выберем элемент *Мень-*

ше или равно и в поле правого списка введем число 2500, установим переключатель с надписью И.

Щелкнем на кнопке ОК — получим результат фильтрации для заданных условий.

Пользовательский автофильтр ✕

Показать только те строки, значения которых:

Приход _____

И
 ИЛИ

Знак вопроса "?" обозначает один любой знак
 Знак "*" обозначает последовательность любых знаков

	A	B	C	D	E	F
1	Таблица 1. Операции банка по счетам клиентов					
2	№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
14	12	10		04.01.02	1300	
175						

Таким образом, условию задачи отвечает запись в одной строке списка. Сумма приходной операции равна 1300 руб.

При выборе операций отношения И (ИЛИ) следует руководствоваться следующим:

— если из всего множества записей в исходном списке нужно отобрать подмножество, значения которого определены правой и левой границами (a, b) (замкнутое подмножество), то следует установить операцию отношения И;

— если из всего множества записей в исходном списке нужно отобрать подмножества, значения которых ограничены только справа или слева, то следует устанавливать операцию отношения ИЛИ (рис. 4.19).

В полях диалогового окна **Пользовательский автофильтр** можно использовать заменители символов — знаки «?» и «*». Заменители

Множество значений в исходном списке

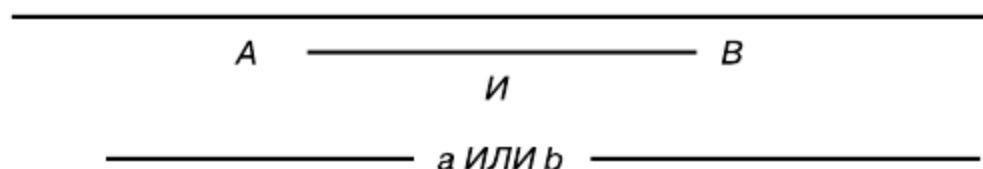


Рис. 4.19. Выбор операций отношения

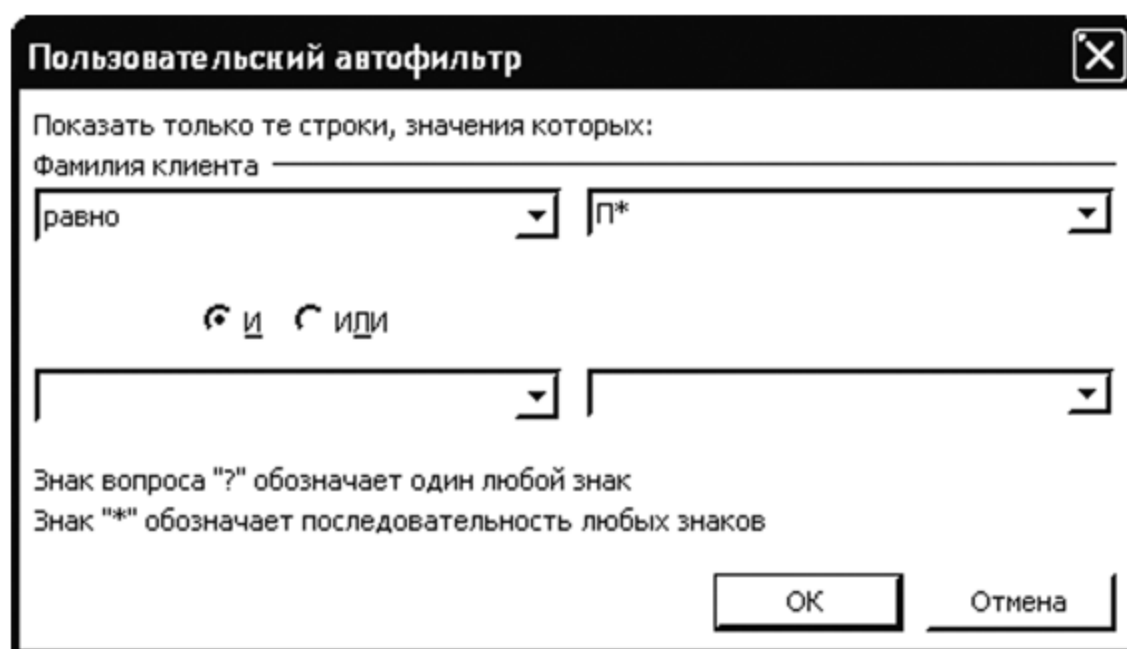
символов применяются традиционным образом: знак «?» заменяет одно любое символместо, а «*» — все следующие символы.

Пример 4.3. Требуется получить сведения по операциям клиентов, фамилии которых начинаются с буквы П.

Решение

Включим инструмент **Пользовательский автофильтр**.

В полях диалогового окна **Пользовательский автофильтр** установим параметры фильтрации, как показано на рисунке.



Результат выборки в соответствии с указанным условием приведен на рисунке.

	A	B	C	D	E	F
1	Таблица 1. Операции банка по счетам клиентов					
2	№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
3	1	1	Петров	03.янв	1000	
5	3	1	Петров	03.янв		500
9	7	5	Павлова	04.янв	1050	

4.4.3. Фильтрация с применением инструмента Расширенный фильтр

Инструмент **Расширенный фильтр** имеет гораздо большие функциональные возможности. В отличие от инструментов **Автофильтр** и **Пользовательский автофильтр** он позволяет:

- задавать условия фильтрации для нескольких столбцов, соединенных оператором отношения;
- задавать несколько условий для данных столбца списка;
- задавать вычисляемые условия;
- помещать результаты фильтрации в любое место рабочего листа.

Особенностью инструмента **Расширенный фильтр** является то, что условия фильтрации записываются не в диалоговых окнах, а в диапазоне смежных ячеек рабочего листа в соответствии с определенными правилами.

Формирование диапазона условий

Условия фильтрации для инструмента **Расширенный фильтр** формируются в соответствии с правилами:

- в верхней строке диапазона (строке заголовков) записываются названия столбцов списка, на значения которых накладываются условия фильтрации (чтобы избежать ошибок, эту операцию лучше выполнять копированием);
- в ячейках строк, расположенных ниже строки заголовков, записываются выражения условий с использованием знаков сравнения (=, >=, <=, <>);
- считается, что выражения условий, записанные в одной строке, соединены отношением И, а записанные в разных строках — отношением ИЛИ;
- диапазон условий не должен содержать пустых строк или пустых столбцов.

Пример 4.4. Требуется составить выражение условий для фильтрации исходного списка в соответствии с условиями: дата операции 10—20 января включительно, номера счетов 5—14 включительно.

Решение

Поскольку условия фильтрации накладываются на значения столбцов с названиями *Дата операции* и *№ счета*, в свободные ячейки рабочего листа введем названия этих столбцов, применяя копирование.

В ячейках строки, расположенной под заголовком, запишем выражения условий фильтрации.

Дата операции	Дата операции	№ счета	№ счета
>=10.01.2008	<= 20.01.2008	>=5	<=14

Выполнение фильтрации

Если выражение условий записано, то можно приступить к фильтрации таблицы. Для этого нужно:

— установить курсор в область исходного списка и выполнить команду меню **Данные/Фильтр/Расширенный фильтр** — откроется окно диалога **Расширенный фильтр** (рис. 4.20);

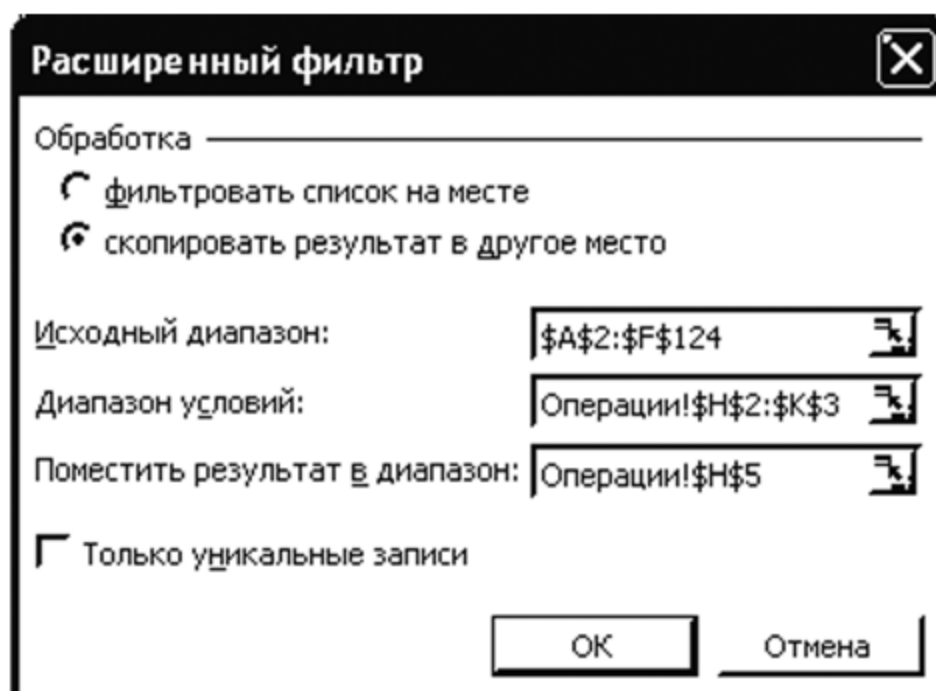


Рис. 4.20. Диалоговое окно **Расширенный фильтр**

— в группе *Обработка* диалогового окна установить соответствующий переключатель *Фильтровать список на месте* или *Скопировать результат в другое место*;

— в поле *Исходный диапазон* ввести адрес диапазона, содержащего исходный список (если курсор был предварительно установлен в области списка, то адрес диапазона будет установлен автоматически);

— в поле *Диапазон условий* ввести адрес диапазона, содержащего условия;

— если результаты фильтрации помещаются в другое место, то в поле *Поместить результат в диапазон* ввести адрес верхней левой ячейки, с которой будет размещаться результат;

— если в исходном списке могут быть повторяющиеся записи, а в результирующую таблицу нужно поместить только одну из них, то нужно установить флажок *Только уникальные записи*, эта опция добавляет к имеющемуся еще один фильтр — *Скрытие полностью повторяющихся строк*, который работает только в том случае, если результат копируется в другое место;

— щелкнуть на кнопке ОК, и в указанном для вывода диапазоне будет получен результат фильтрации.

Для условий примера 4.4 результат фильтрации приведен на рис. 4.21.

Н	І	Ј	К	Л	М
Дата операции	Дата операции	№ счета	№ счета		
>=10.01.08	<=20.01.08	>=5	<=14		

№ операции	№ счета	Фамилия клиента	Дата операции	Приход	Расход
44	8	Петрова	11.01.02		800
45	9	Охрименко	12.01.02		1100
47	10	Куликов	12.01.02	700	
50	12	Сидорова	14.01.02	1500	
57	8	Петрова	17.01.02		1100
58	9	Охрименко	17.01.02	1250	
62	5	Павлова	18.01.02	1500	
63	7	Трусов	18.01.02		240
64	9	Охрименко	18.01.02		500
65	11	Круглов	18.01.02	2500	
66	13	Наумова	19.01.02	1000	

Рис. 4.21. Результат фильтрации по заданным условиям

В диапазоне условий можно ввести любое количество условий.

Пример 4.5. Требуется отфильтровать записи исходной таблицы, которые отвечают условиям: дата операции 10–20 января включительно, номера счетов с 5-го по 14-й, сумма прихода от 1000 до 2500 включительно, а также сведения обо всех расходных операциях в период с 24 по 29 января с суммой больше 1100 по клиентам, фамилии которых начинаются с букв «Я» или «Б».

Решение

Сформируем строку заголовка для условия фильтрации. Из условия задачи следует, что условия должны быть наложены на данные

столбцов исходного списка с именами *Дата операции*, *№ счета*, *Приход*, *Расход* и *Фамилия клиента*, поэтому на свободном участке рабочего листа в ячейки диапазона, например Н3:О3, введем эти названия.

В строках, расположенных ниже строки заголовка, запишем условия, исходя из следующих соображений: условия «дата операции 10–20 января включительно, номера счетов с 5-го по 14-й, сумма прихода от 1000 до 2500 включительно» выполняются совместно, к ним применима операция отношения И, поэтому запишем их в одной строке под строкой заголовка; условия «дата операции 10–20 января включительно, номера счетов с 5-го по 14-й, сумма прихода от 1000 до 2500 включительно, фамилии которых начинаются с буквы Я» связано с первым условием отношением ИЛИ, поэтому в диапазоне условий запишем его в строке ниже; условия «дата операции 10–20 января включительно, номера счетов с 5-го по 14-й, сумма прихода от 1000 до 2500 включительно, фамилии которых начинаются с буквы Б» связано с предыдущими условиями отношением ИЛИ, поэтому в диапазоне условий запишем его в следующей строке.

Результат фильтрования по сформированному условию приведен на рисунке.

	Н	І	Ј	К	Л	М	Н	О
1	Условие запроса							
2								
3	Дата операции	Дата операции	№ счета	№ счета	Приход	Приход	Расход	Фамилия клиента
4	>=10.01.08	<=20.01.08	>4	<15	>=1000	<=2500		
5	>=24.01.08	<=29.01.08					>1100	Я*
6	>=24.01.08	<=29.01.08					>1100	Б*
7	Результат							
8	№ операции	Дата операции	№ счета	Фамилия клиента	Приход	Расход		
9	50	14.01.02	12	Ниязова	1500			
10	58	17.01.02	9	Охрименко	1250			
11	62	18.01.02	5	Павлова	1500			
12	65	18.01.02	11	Углова	2500			
13	66	19.01.02	13	Горбатенко	1000			
14	85	24.01.02	19	Ямщикова		2350		
15	90	25.01.02	10	Борисов		6100		
16	105	28.01.02	10	Борисов		2500		

Создание вычисляемых условий

Вычисляемые условия позволяют использовать в выражении условий для расширенного фильтра вычисленные значения, возвращаемые формулой. При формировании вычисляемых условий необходимо соблюдать правила:

— заголовок над вычисляемым условием должен отличаться от заголовков столбцов в списке. Это условие противоположно требованиям для обычных условий;

— в формулах адресные ссылки на ячейки, находящиеся вне исходного списка, должны быть абсолютными;

— адресные ссылки на ячейки в списке должны быть относительными.

Пример 4.6. Требуется создать таблицу, содержащую сведения об операциях, отвечающих условиям: сумма операции прихода больше среднего значения всех таких операций.

Решение

В любой свободной ячейке рабочего листа введем формулу для вычисления среднего значения всех операций расхода (например, в ячейке Н2) =СРЗНАЧ(Е4:Е125).

Сформируем условие для фильтрации: в ячейку, например Н3, введем любое слово или сочетание, которое не совпадает с названиями столбцов исходного списка, например «Выборка», а в ячейку Н4 вычисляемое условие =D6>\$D\$2. Следует обратить внимание на то, что адресная ссылка в условии должна быть сделана на ячейку первой строки столбца (в нашем случае D6). Если формула записана правильно, то после окончания ввода в ячейке появится значение ИСТИНА или ЛОЖЬ.

Результирующая отфильтрованная таблица приведена на рисунке.

Н2		fx =СРЗНАЧ(Е4:Е125)						
	Е	F	G	Н	И	J	К	L
2				2308,833333				
3	Приход	Расход		Выборка				
4	1000			ЛОЖЬ				
5	2000							
6		500		№ операции	Дата операции	№ счета	Фамилия клиента	Приход
7		1000		10	04.01.02	8	Сидоров	5000
8	1500			16	05.01.02	14	Гобачева	5000
9	1300			23	06.01.02	21	Белкина	5000
10	1050			30	08.01.02	28	Проклова	5000
11	1100			34	08.01.02	10	Борисов	20000
12	1200			36	08.01.02	20	Крутиков	15000
13	5000			41	10.01.02	15	Сидоренко	2800
14	1500			56	17.01.02	23	Березин	4700
15	1300			65	18.01.02	11	Углова	2500
16	1050			76	22.01.02	17	Миронова	3150
17	1100			80	24.01.02	29	Проклова	3500
18	1200			89	25.01.02	2	Разумовская	4500
19	5000							

Вывод в итоговую таблицу только нужных граф

В исходной таблице может быть много столбцов, отображать их все при создании отфильтрованной таблицы не всегда необходимо. Для того чтобы отобразить только нужные столбцы исходного списка, нужно:

– скопировать в смежные ячейки строки рабочего листа названия только тех столбцов, которые нужно отобразить;

– в диалоговом окне **Расширенный фильтр** в поле *Поместить результат в диапазон* указать адрес диапазона, в который введены названия столбцов.

На рисунке 4.22 показан результат создания таблицы с тремя столбцами для условия задачи примера 4.6.

	Н	І	Ј
3	Выборка		
4	ЛОЖЬ		
5			
6	№ счета	Фамилия клиента	Приход
7	8	Сидоров	5000
8	14	Гобачева	5000
9	21	Белкина	5000
10	28	Проклова	5000
11	10	Борисов	20000
12	20	Крутиков	15000
13	15	Сидоренко	2800
14	23	Березин	4700
15	11	Углова	2500
16	17	Миронова	3150
17	29	Проклова	3500
18	2	Разумовская	4500

Рис. 4.22. Результат создания таблицы с тремя столбцами

4.5. ПРИМЕНЕНИЕ ФУНКЦИЙ БАЗ ДАННЫХ ДЛЯ АНАЛИЗА ДАННЫХ

4.5.1. Функции баз данных табличного процессора

Табличный процессор имеет 12 встроенных функций, называемых функциями баз данных, которые специально предназначены для работы со списками. В списке Мастера функций они размещаются в категории Работа с базой данных.

Функции баз данных являются аналогами статистических функций, но в отличие от статистических, функции баз данных возвращают результат, вычисленный при обработке тех данных списка, которые удовлетворяют заданным условиям.

Все функции баз данных имеют одинаковый синтаксис:

ИМЯ_ФУНКЦИИ (База_данных; Поле; Критерий),

где параметр *База_данных* устанавливает адрес диапазона, в котором размещен список (вместе с названиями столбцов), содержащий анализируемые исходные данные;

параметр *Поле* определяет адрес ячейки, содержащей название столбца списка, в котором группируются данные для расчета;

параметр *Критерий* указывает адрес диапазона, в котором записано условие на выборку данных для последующей обработки.

Перечень функций баз данных табличного процессора, их назначение и синтаксис приведены в табл. 4.1.

Таблица 4.1

Функции баз данных табличного процессора

№ п/п	Имя функции	Назначение	Синтаксис
1	БДДИСП	Оценивает дисперсию генеральной совокупности по выборке, используя числа в указанном поле всех записей, удовлетворяющих критерию	БДДИСП (База_данных; Поле; Критерий)
2	БДДИСПП	Вычисляет дисперсию генеральной совокупности, используя числа в указанном поле всех записей, удовлетворяющих критерию	БДДИСПП (База_данных; Поле; Критерий)
3	БДПРОИЗВЕД	Перемножает значения в указанном поле всех записей, удовлетворяющих критерию	БДПРОИЗВЕД (База_данных; Поле; Критерий)
4	БДСУМ	Суммирует числа в указанном поле всех записей базы данных, удовлетворяющих критерию	БДСУМ (База_данных; Поле; Критерий)
5	БИЗВЛЕЧЬ	Извлекает из базы данных одну запись, которая удовлетворяет заданному критерию	БИЗВЛЕЧЬ (База_данных; Поле; Критерий)
6	БСЧЕТ	Вычисляет количество числовых ячеек в выборке из базы данных по заданному критерию	БСЧЕТ (База_данных; Поле; Критерий)

№ п/п	Имя функции	Назначение	Синтаксис
7	БСЧЕТА	Вычисляет количество не пустых ячеек в выборке из базы данных по заданному критерию	БСЧЕТА (База_данных; Поле; Критерий)
8	ДМАКС	Суммирует числа в указанном поле всех записей базы данных, удовлетворяющих критерию	ДМАКС (База_данных; Поле; Критерий)
9	ДМИН	Возвращает наименьшее число в указанном поле всех записей, удовлетворяющих критерию	ДМИН (База_данных; Поле; Критерий)
10	ДСРЗНАЧ	Усредняет значения всех записей в указанном поле, удовлетворяющих критерию	ДСРЗНАЧ (База_данных; Поле; Критерий)
11	ДСТАНДОТКЛ	Оценивает стандартное отклонение на основе выборки из генеральной совокупности, используя числа всех записей в указанном поле, удовлетворяющих критерию	ДСТАНДОТКЛ (База_данных; Поле; Критерий)
12	ДСТАНДОТКЛП	Вычисляет стандартное отклонение генеральной совокупности, используя числа в указанном поле всех записей, удовлетворяющих критерию	ДСТАНДОТКЛП (База_данных; Поле; Критерий)

4.5.2. Технология анализа данных

Для того чтобы выполнить вычисления по данным списка, нужно:

- на свободном участке рабочего листа сформировать условие для вычислений (критерий). Условия формируются по тем же правилам, что и для инструмента **Расширенный фильтр**;
- поместить курсор в ячейку, где должен помещаться результат, вставить формулу функции с записанными параметрами.

Пример 4.7. Требуется вычислить итоговую сумму по всем расходным операциям с номерами счетов 4–15 включительно, проведенным в период с 12 по 24 января.

Решение

Запишем выражение условия в свободном диапазоне рабочего листа, например Н3:К4.

	E	F	G	H	I	J	K
1	клиентов						
2							
3	Приход	Расход		№ счета	№ счета	Дата операции	Дата операции
4	1000			>=4	<=15	>=12.01.08	<=24.01.08
5	2000						
6		500					

В ячейку, например, I1 введем формулу
 =БДСУММ(A3:F125;F3;H3:K4).

I1		fx =БДСУММ(A3:F125;F3;H3:K4)				
	F	G	H	I	J	
1			Результат	10140		
2						

Следует иметь в виду, что при вычислениях с применением функций баз данных, когда результат вычисления должен оставаться неизменным, нельзя удалять или изменять выражение условия.

4.6. АНАЛИЗ ДАННЫХ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ КОНСОЛИДАЦИИ

Под консолидацией понимается объединение данных, хранящихся в одном или в нескольких списках, выполнение над объединенными данными необходимых вычислений и формирование результирующей таблицы.

Консолидацию можно выполнять по расположению или по категории. Консолидация по расположению предполагает сбор информации из ячеек исходных списков, расположенных одинаково. При консолидации по категории табличный процессор использует в качестве признака для объединения заголовки столбцов или строк списков. Такой способ консолидации предоставляет большую свободу организации данных.

Консолидация выполняется с помощью инструмента **Консолидация**, который включается командой меню **Данные/Консолидация**.

Для выполнения консолидации нужно выполнить операции:

- установить курсор в ячейку рабочего листа, в которой будет размещаться левая верхняя ячейка результирующей таблицы;
- включить инструмент **Консолидация**;
- в диалоговом окне **Консолидация** в списке *Функция* выбрать функцию для выполнения необходимых вычислений;

— в поле *Ссылка* указать адрес диапазона или всего списка, данные которого предполагается консолидировать, если есть еще списки для консолидации щелкнуть на кнопке *Добавить*, ввести следующий диапазон и т.д.;

— если консолидация выполняется по категории, в группе *Использовать в качестве имен* установить флажок *Подписи верхней строки*, при консолидации по расположению — флажок *Значения левого столбца*;

— для установления связи с исходными списками установить флажок *Создавать связи с исходными данными*;

— щелкнуть на кнопке *ОК*.

Пример 4.8. Требуется, используя данные исходной таблицы (рис. 4.1), создать таблицу, отражающую сведения об общих суммах приходных и расходных операций по каждому операционному дню.

Решение

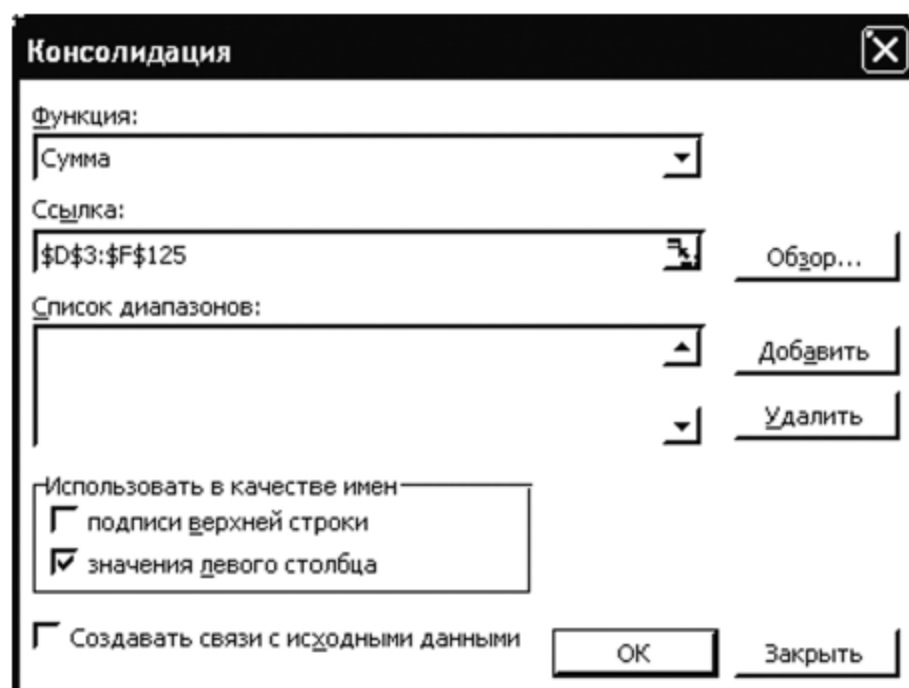
Установим курсор в свободную ячейку рабочего листа, например Н3, и включим инструмент *Консолидация*.

В диалоговом окне **Консолидация**:

— в списке *Функция* выберем *Сумма* (устанавливается по умолчанию);

— в поле *Ссылка* укажем ссылку на диапазон списка так, чтобы самым левым столбцом диапазона был столбец с названием *Дата операции*, т.е. столбец, который содержит признаковые данные для консолидации (консолидация по расположению), и включал столбцы с именами *Приход* и *Расход*, данные которых должны суммироваться при формировании консолидированной таблицы;

— в группе *Использовать в качестве имен* установим флажок *Значения левого столбца* и щелкнем на кнопке *ОК*.



Результирующая консолидированная таблица приведена на рисунке.

	D	E	F	G	H	I	J
1	список				Результат консолидации		
2							
3	Дата операции	Приход	Расход		Дата операции	Приход	Расход
4	03.01.02	1000			3 янв	5800	1500
5	03.01.02	2000			4 янв	11150	
6	03.01.02		500		5 янв	11350	
7	03.01.02		1000		6 янв	9550	
8	03.01.02	1500			7 янв	7550	
9	03.01.02	1300			9 янв		
10	04.01.02	1050			8 янв	43000	2050

В отличие от консолидации по расположению, которая может применяться для данных одной таблицы, консолидация по категориям применяется только в тех случаях, когда нужно консолидировать данные нескольких таблиц, как это показано на рисунке.

	A	B	C	D	E	F	G	H
1	Таблица 1			Таблица 2		Результат консолидации		
2	Приход	Расход		Приход	Расход		Приход	Расход
3	1000			1200			2200	
4	2000			5000			7000	
5		500		1700			1700	500
6		1000		1300			1300	1000
7	1500			1200			2700	
8	1300			1050			2350	
9	1050			1100			2150	

4.7. АНАЛИЗ ДАННЫХ НА ОСНОВЕ СВОДНЫХ ТАБЛИЦ

4.7.1. Общие сведения

Сводная таблица является мощным аналитическим инструментом, позволяющим суммировать данные полей списка табличного процессора. При создании сводной таблицы можно задать способ организации таблицы и тип выполняемых вычислений. После создания сводной таблицы можно менять ее расположение для удобства просмотра данных с других позиций анализа. Сводная таблица связана с данными исходного списка, при изменении этих данных она может обновляться вручную или автоматически.

4.7.2. Технология создания сводной таблицы

Включение инструмента Мастер сводных таблиц

Для создания сводной таблицы нужно выполнить операции:

- установить курсор в любую ячейку диапазона, в котором размещен исходный список;
- выполнить команду меню **Данные/Сводная таблица** – откроется диалоговое окно **Мастер сводных таблиц** (рис. 4.23).

В качестве источника данных для создания сводной таблицы **Мастер сводных таблиц** может использовать список или несколько списков, сформированных на рабочих листах активной рабочей книги, внешний источник данных, например из таблицы базы данных, другие сводные таблицы или сводные диаграммы.

Сводная таблица с помощью мастера сводных таблиц создается за три шага, включающих:

- задание типа источника данных и вида создаваемого отчета;
- указание расположения исходных данных в источнике данных;
- указание места для размещения результирующей таблицы;
- указание полей, размещаемых в таблице, способ организации таблицы и типы вычислений.

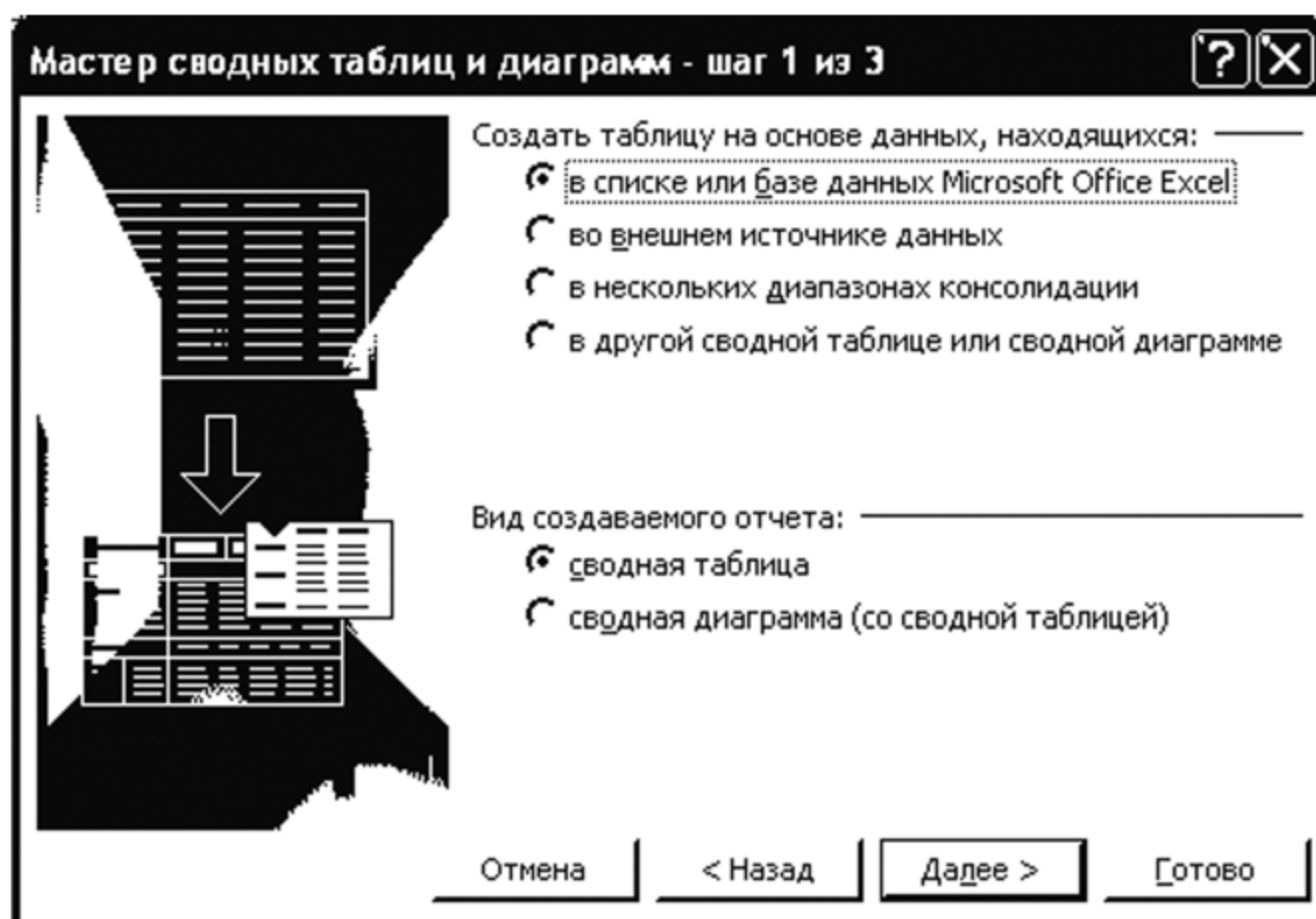


Рис 4.23. Диалоговое окно **Мастер сводных таблиц**

Задание типа источника данных и вида создаваемого отчета

Источник данных задается в диалоговом окне **Мастер сводных таблиц** (см. рис. 4.23). Для указания источника данных в группе *Создать таблицу на основе имеющихся данных* нужно установить соответствующий переключатель. Для указания вида отчета в группе *Вид создаваемого отчета* установить переключатель *Сводная таблица* или *Сводная диаграмма*. После выполнения приведенных операций можно перейти к следующему шагу, щелкнув на кнопке *Далее*.

Указание места расположения исходной таблицы

Если источник данных находится в списке или базе данных табличного процессора, то откроется окно диалога, приведенное на рис. 4.24. Место расположения исходной таблицы задается в этом диалоговом окне в поле *Диапазон*. Если источник данных находится в закрытой рабочей книге, то нужно воспользоваться кнопкой *Обзор*.

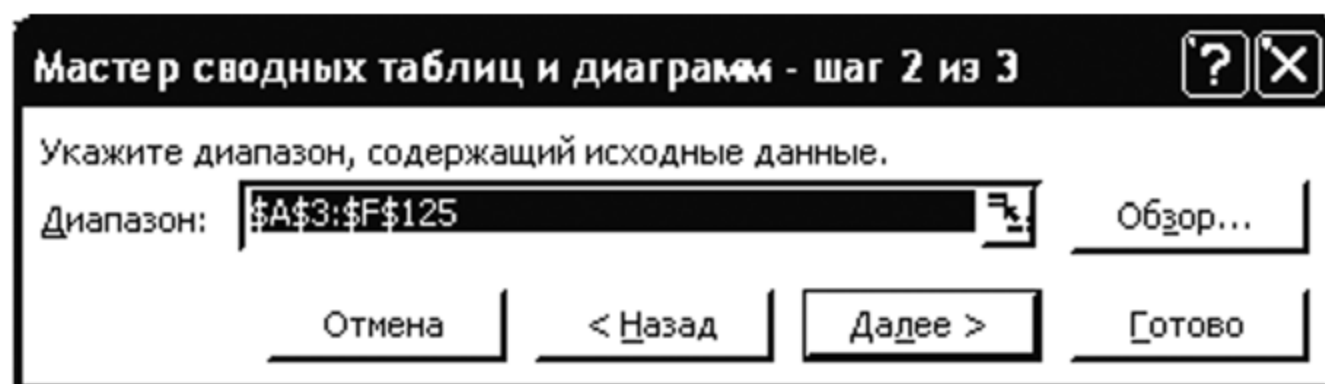


Рис. 4.24. Указание диапазона, содержащего исходные данные

При создании таблицы на основе внешнего источника данных откроется диалоговое окно, вид которого приведен на рис. 4.25.

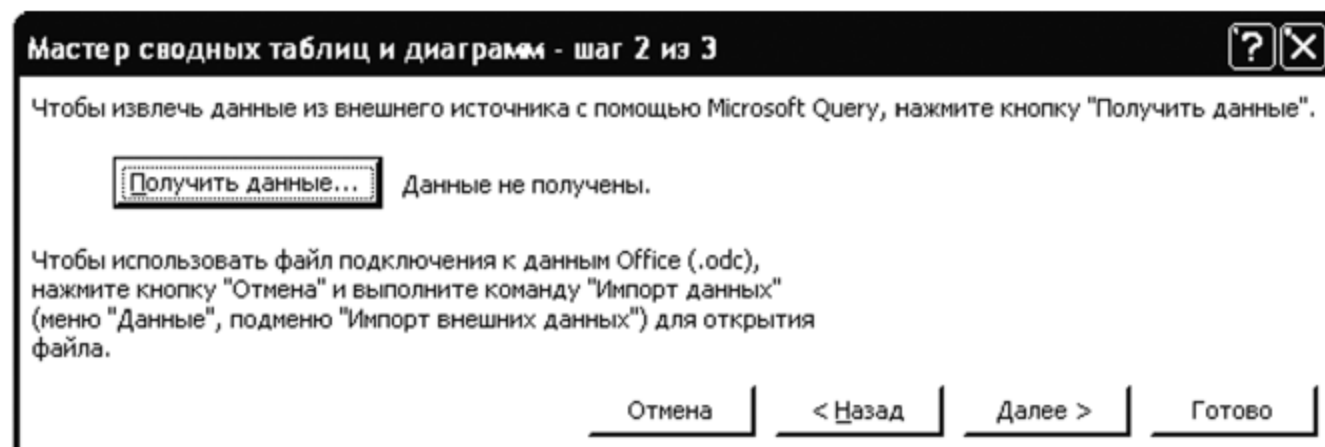


Рис. 4.25. Создание таблицы на основе внешнего источника данных

Для определения внешнего источника нужно щелкнуть на кнопке *Получить* — откроется окно **Получение источника данных** (рис. 4.26). В этом диалоговом окне можно выбрать базу данных реляционного типа, выполнить сохраненный запрос или указать источник данных OLAP.



Рис. 4.26. Диалоговое окно **Выбор источника данных**

При создании сводной таблицы на основе нескольких диапазонов консолидации откроется диалоговое окно, вид которого приведен на рис. 4.27.



Рис. 4.27. Создание таблицы на основе нескольких диапазонов

Указание места расположения результатирующей таблицы

В последнем диалоговом окне (рис. 4.28) **Мастер сводных таблиц** определяется место, где будет располагаться результирующая таб-

лица. Чтобы поместить таблицу в новом рабочем листе, следует установить переключатель *Новый лист*.



Рис. 4.28. Указание места размещения таблицы

Перед тем как щелкнуть на кнопке *Готово*, если это необходимо, можно установить параметры, щелкнув на соответствующей кнопке диалогового окна, — откроется диалоговое окно **Параметры сводной таблицы** (рис. 4.29). К установке параметров можно вернуться и позднее, когда сводная таблица будет уже создана.

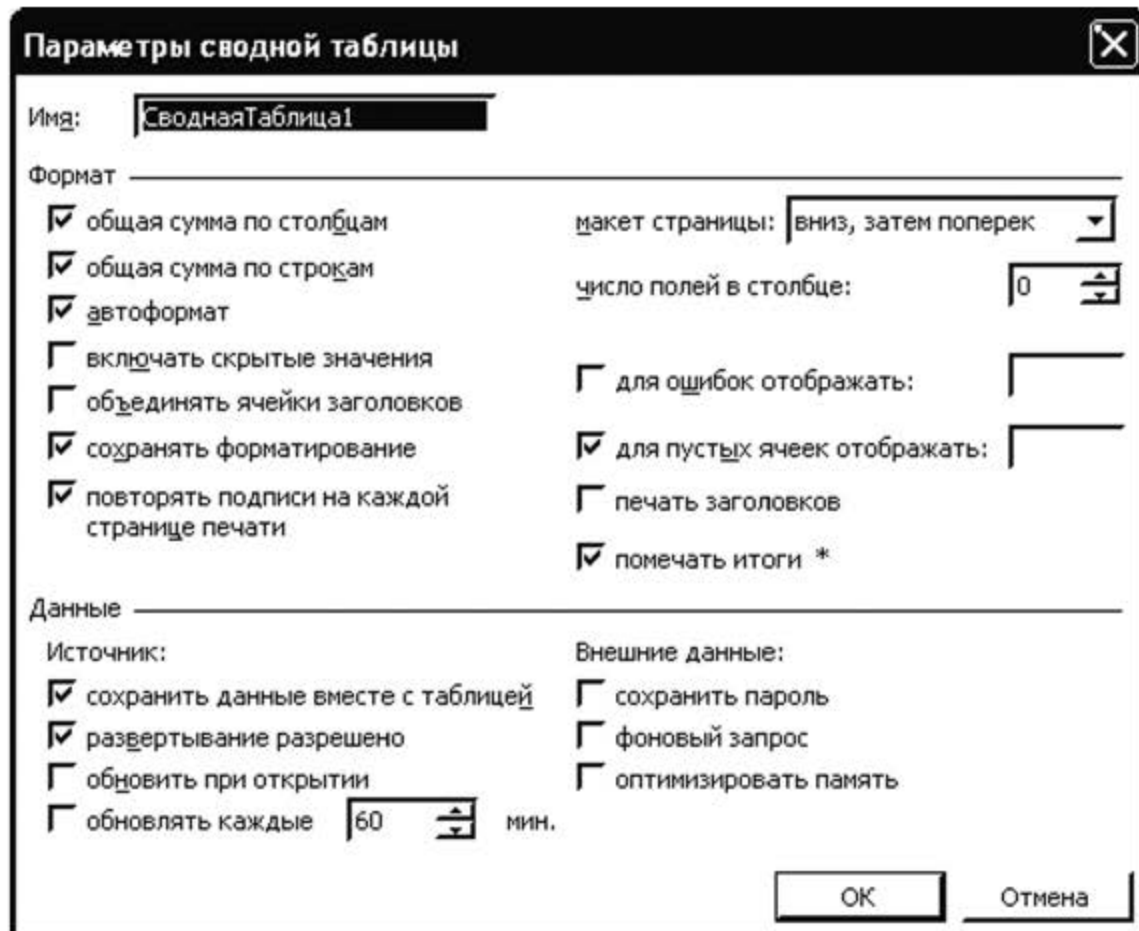


Рис. 4.29. Диалоговое окно **Параметры сводной таблицы**

Создание макета сводной таблицы

После выполнения операций с помощью **Мастер сводных таблиц** и щелчка на кнопке *Готово* в последнем диалоговом окне в рабочем листе будет отображен пустой макет таблицы, панель инструментов **Сводные таблицы** и диалоговое окно **Список полей сводной таблицы** (рис. 4.30).

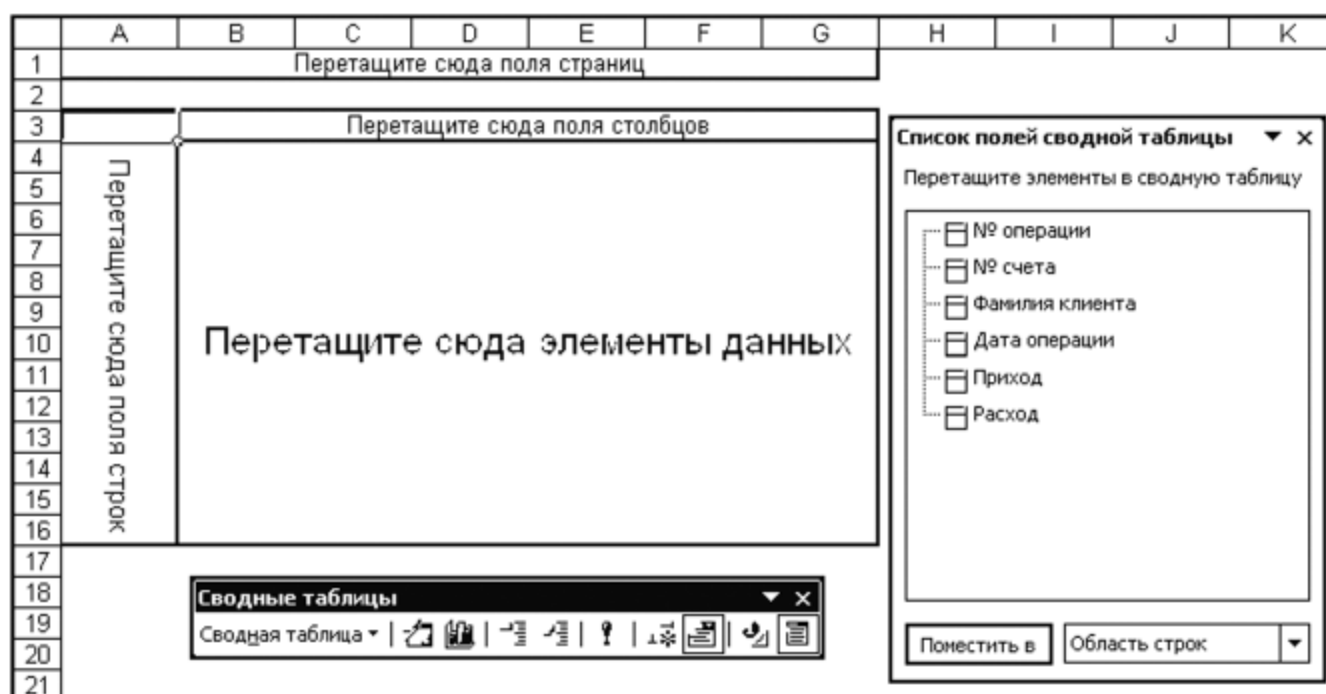


Рис. 4.30. Диалоговое окно **Список полей сводной таблицы**

Формирование макета выполняется путем перетаскивания кнопок с названиями столбцов исходного списка из диалогового окна **Список полей сводной таблицы** в области макета с надписью *Перетащите сюда поля столбцов (строк)*. В область с надписью *Перетащите сюда элементы данных* следует перетащить кнопки с названиями полей, над данными которых должны быть выполнены вычислительные операции.

Пример 4.9. Требуется создать сводную таблицу на основе списка, приводимого в предыдущих примерах, в которой должны быть отражены суммарные значения приходных и расходных операций для каждого клиента за текущий период.

Решение

Установим курсор в область исходного списка и включим инструмент **Мастер сводных таблиц**.

В окне **Мастер сводных таблиц** (шаг 1) установим переключатель *В списке или базе данных Microsoft Office Excel* и переключатель *Сводная таблица*.

В поле *Диапазон* окна **Мастер сводных таблиц** (шаг 2) укажем адрес диапазона, в котором расположен исходный список.

В окне **Мастер сводных таблиц** (шаг 3) установим переключатель *Новый рабочий лист*.

В появившемся на рабочем листе макете выполним операции:

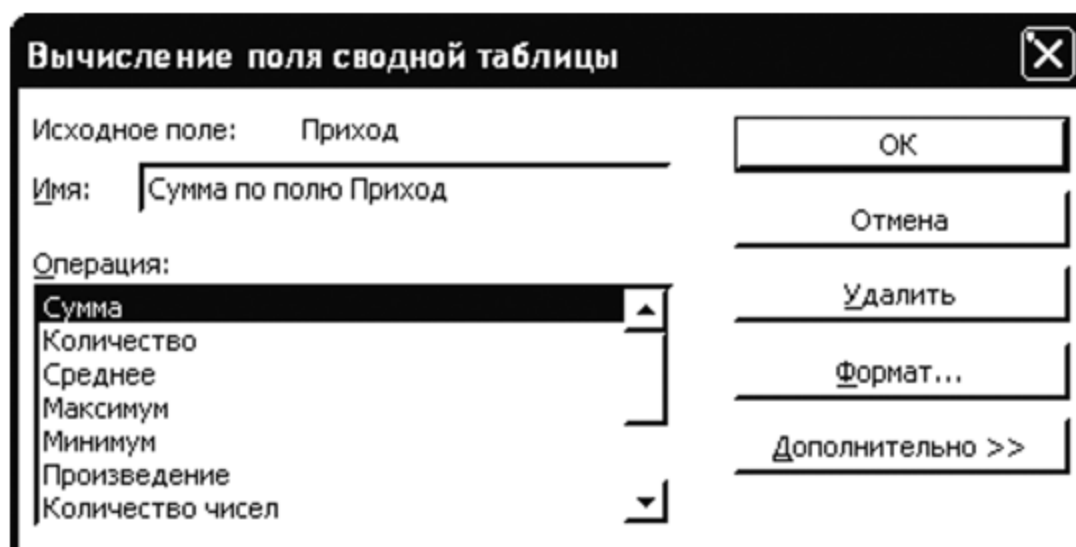
– перетащим кнопку *Фамилия клиента* в область с надписью *Перетащите сюда названия строк*;

– в область с надписью *Перетащите сюда элементы данных* перетащим кнопки с названиями *Приход* и *Расход*. После выполнения этих операций на рабочем листе будет сформирована таблица, фрагмент которой приведен на рисунке. В поле *Итог* этой таблицы помещены числа, соответствующие количеству операций расхода и прихода, выполненных по счетам каждого клиента;

	A	B	C	D
1	Перетащите сюда поля страниц			
2				
3	Фамилия клиента	Данные	Итог	
4	Белкина	Количество по полю Приход	1	
5		Количество по полю Расход	2	
6	Березин	Количество по полю Приход	2	
7		Количество по полю Расход	3	
8	Борисов	Количество по полю Приход	3	
9		Количество по полю Расход	3	

– поместим курсор в любое поле таблицы с надписью *Количество по полю Приход* и включим контекстное меню. В списке меню выберем *Параметры поля*;

– в открывшемся окне диалога *Вычисление поля сводной таблицы* выберем из списка *Операция* элемент *Сумма*, повторим эту операцию для поля таблицы с надписью *Количество по полю Расход*.



После выполнения всех операций будет получена результирующая таблица.

	А	В	С
1	Перетащите сюда поля страниц		
2			
3	Фамилия клиента ▼	Данные ▼	Итог
4	Белкина	Сумма по полю Приход	5000
5		Сумма по полю Расход	2150
6	Березин	Сумма по полю Приход	6000
7		Сумма по полю Расход	2900
8	Борисов	Сумма по полю Приход	22000
9		Сумма по полю Расход	14600
10	Гобачева	Сумма по полю Приход	5000

4.7.3. Изменение сводной таблицы

Созданную сводную таблицу можно изменять. Например, чтобы переместить поле с оси строк на ось столбцов, нужно в диалоговом окне **Список полей сводной таблицы** в списке выбрать *Область столбцов*, выделить кнопку с нужным названием и щелкнуть на кнопке *Поместить в*. Столбцы и строки в сводной таблице поменяются местами (операция транспонирования).

Элементы, которые не нужны для анализа, можно скрывать. Эта операция выполняется путем раскрытия стрелки в нужном поле и последующим выбором только тех элементов, данные которых подлежат анализу.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что представляет собой список?
2. Какие требования предъявляются к спискам?
3. Какие операции можно выполнять со списками?
4. Какие операции можно выполнять при использовании инструмента Форма?
5. В чем заключается основное отличие Автофильтра от Пользовательского автофильтра?
6. Как создается список для пользовательского порядка сортировки, в каких случаях это бывает необходимо?
7. Какова технология создания условий для расширенного фильтра и функций баз данных?

8. Что такое консолидация табличных данных, в каких случаях она бывает необходима?

ЗАДАЧА ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задание. Создайте список, подобный приведенному (рис. 4.1), и выполните все задания, указанные в примерах.

АВТОМАТИЗАЦИЯ РЕШЕНИЯ ТРУДОЕМКИХ И ТИПОВЫХ ЗАДАЧ

5.1. СОЗДАНИЕ МАКРОСОВ

Макрос — это набор инструкций, записанных на языке Visual Basic (VB), которые задают последовательность действий, выполняемую табличным процессором автоматически без участия пользователя. Обычно макросы создаются для автоматизации часто повторяющихся одинаковых задач. Макрос можно создать двумя способами — автоматически записать последовательность необходимых действий с помощью макрорекордера или вручную ввести необходимые инструкции на языке Visual Basic в специальный модуль проекта. При этом если в первом случае пользователь, создающий макрос, может не знать языка программирования Visual Basic, то во втором знание языка программирования необходимо.

5.1.1. Создание макросов с помощью макрорекордера

Процесс создания макроса состоит из трех этапов: активизация режима записи и присвоение макросу имени; выполнение действий, которые следует записать; остановка записи макроса. Рассмотрим на примере процесс записи макроса.

Пример 5.1. Требуется создать макрос, выполняющий решение системы линейных уравнений.

Решение

Для определенности положим, что элементы матрицы A записаны в диапазоне ячеек с адресом A2:D5, элементы вектора B — в диапазоне ячеек F2:F5, а результат помещается в диапазон H2:H5. Макрос должен запускаться комбинацией клавиш <Ctrl> + <U>.

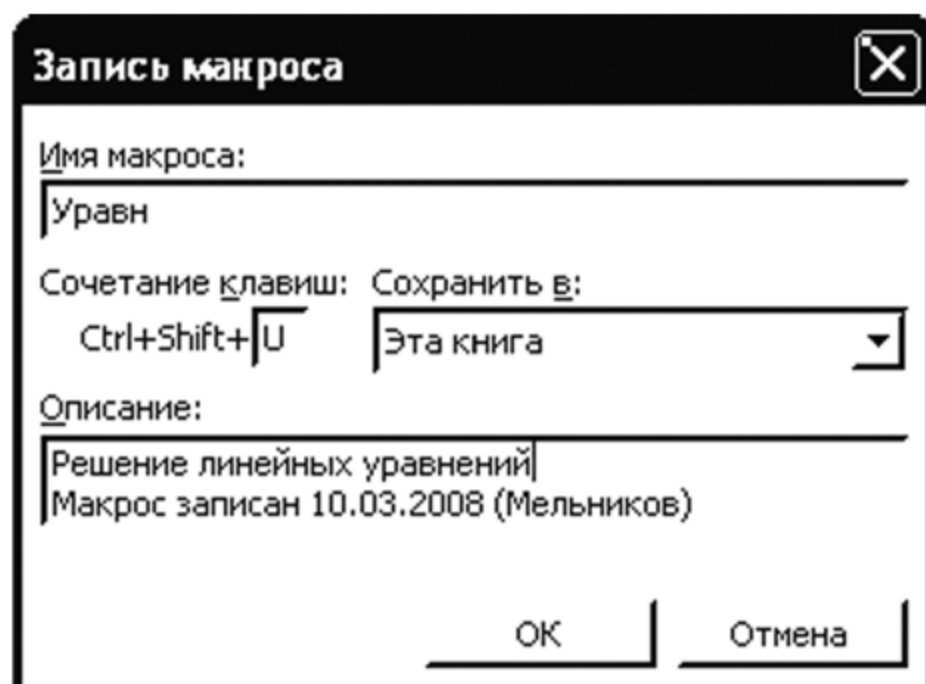
Для создания макроса выполним последовательность действий: Выполним команду меню **Сервис** → **Макрос** → **Начать запись**.

В открывшемся окне диалога **Запись макроса**:

— в поле *Имя макроса* введем имя создаваемого макроса, например «Уравн»;

— назначим комбинацию клавиш для запуска макроса;

- чтобы сохранить создаваемый макрос только в текущей рабочей книге убедимся, что в списке *Сохранить в* выбран пункт *Эта Книга*;
- в поле *Описание* введем краткий комментарий к макросу;



- чтобы начать запись макроса щелкнем на кнопке ОК диалогового окна — на экране появится панель управления с кнопкой *Остановить запись*.



Выполним все действия, вводя в ячейки диапазона Н2:Н5 формулу для решения системы линейных уравнений

$$=МУМНОЖ(МОБР(A2:D5);F2:F5).$$

Щелкнем на кнопке *Остановить запись* панели управления.

Чтобы проверить работу макроса, очистим диапазон Н2:Н5 с имеющимся решением и запустим макрос комбинацией клавиш. Макрос выполнит записанные действия и в ячейках диапазона Н2:Н5 будет вычислен результат.

Н2		fx {=МУМНОЖ(МОБР(A2:D5);F2:F5)}							
	A	B	C	D	E	F	G	H	
1	A				B			X	
2	9	5	4	7		0		1	
3	4	6	8	7		6		-1,4557	
4	5	8	7	6		3		3,113924	
5	5	6	8	7		7		-2,02532	
6									

5.1.2. Использование окна диалога Макрос

Чтобы запустить макрос, необязательно знать назначенное ему сочетание клавиш. Для этой цели можно воспользоваться окном диалога **Макрос** (рис. 5.1).

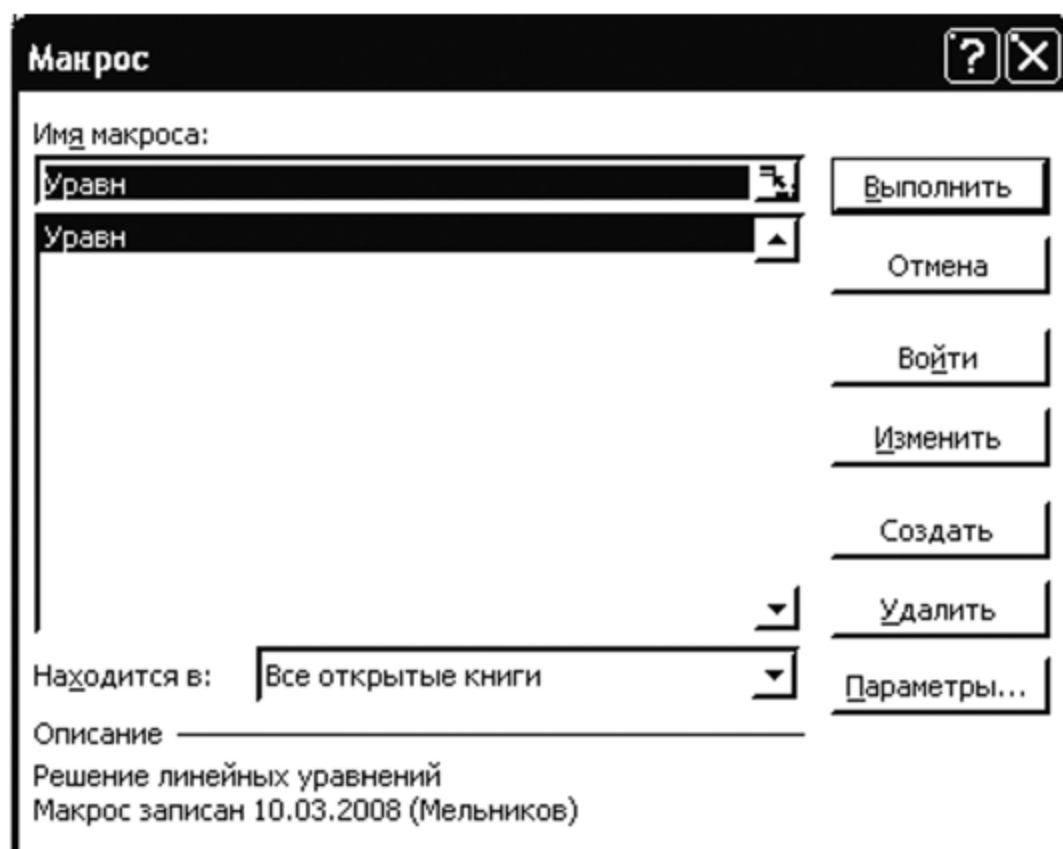


Рис. 5.1. Диалоговое окно **Макрос**

Для запуска макроса с помощью окна диалога **Макрос** выполним действия:

- выполним команду **Сервис/Макрос/Макросы**, откроется окно диалога **Макрос**;
- в списке *Имя макроса* выберем нужный макрос и щелкнем на кнопке *Выполнить*.

Используя окно диалога **Макрос**, можно удалить созданный макрос, создать новый или изменить существующий, а также просмотреть код инструкций макроса.

Для просмотра кода инструкций созданного нами макроса в диалоговом окне **Макрос** нужно щелкнуть на кнопке *Войти*, откроется главное окно редактора Microsoft Visual Basic, в окне редактирования кода которого отобразится код инструкций макроса (рис. 5.2).

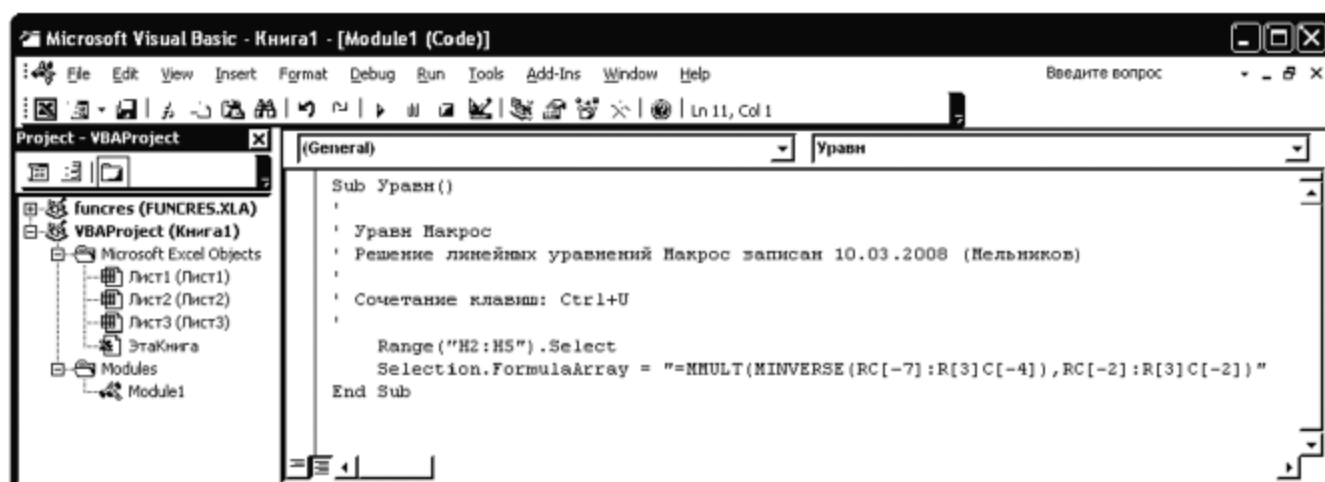


Рис. 5.2. Главное окно редактора Microsoft Visual Basic

Код инструкций созданного макроса приведен в листинге 1.

Листинг 1.

Sub Уравн()

“

“ Уравн Макрос

“ Решение линейных уравнений Макрос записан 10.03.2008 (Мельников)

“ Сочетание клавиш: Ctrl+U

“

Range(«H2:H5»).Select

Selection.FormulaArray="=MMULT(MINVERSE(RC6-7c:R63cC6-4c), RC6-2c:R63cC6-2c)"

End Sub

Созданный макрос не удобен с той точки зрения, что он позволяет решать только системы, состоящие из четырех уравнений, коэффициенты матриц которых записаны в ячейках заранее определенных диапазонов.

Для создания универсального с этих позиций макроса его нужно частично изменить, добавив дополнительные инструкции на языке Visual Basic. Редактирование макроса для придания ему новых функций выполняется в окне редактирования кода редактора Microsoft Visual Basic.

В листинге 2 приведен код инструкций макроса, который является более универсальным с точки зрения отмеченных выше недостатков.

Листинг 2.

Sub Уравн()

“
“ Уравн Макрос
“ Решение линейных уравнений Макрос записан 10.03.2008 (Мельников)
“
“ Сочетание клавиш: Ctrl+U
“

```
Dim rez As String
Dim A As String
Dim B As String
A = InputBox(«Введите адрес диапазона матрицы A»)
B = InputBox(«Укажите адрес диапазона вектора B»)
rez = InputBox(«укажите адрес диапазона для вывода результата»)
Range(rez).Select
Selection.FormulaArray = «=MMULT(MINVERSE(« & A & «),» & B & «)»
End Sub
```


Можно заметить, что для внесения даже простых изменений в код инструкции необходимо знание языка Visual Basic и среды разработки.

В настоящем пособии не ставится задача подробного описания возможностей MS Visual Basic. Поэтому далее рассмотрим только те аспекты, которые будут необходимы для создания макросов, а также функций пользователя.

5.2. КОМПОНЕНТЫ СРЕДЫ VISUAL BASIC FOR APPLICATION


Проекты VBA создаются с помощью среды разработки, создаваемой редактором Visual Basic.

Для активизации редактора можно использовать следующие варианты:

- выполнить команду меню MS Excel **Сервис** → **Макрос** → **Редактор Visual Basic** (Tools Macro Visual Basic Editor);
- щелкнуть на кнопке *Редактор Visual Basic* (Visual Basic Editor) панели инструментов Visual Basic в окне приложения Excel ;
- нажать комбинацию клавиш <Alt> + <F11>.


После выполнения одной из этих команд активизируется редактор Visual Basic и на экране появится его главное окно (рис. 5.2).

Вернуться из редактора в приложение Excel можно одним из следующих способов:

- щелкнуть на клавише Excel панели задач;
- щелкнуть на кнопке *Вид Microsoft Excel* (View Microsoft Excel) на панели инструментов редактора ;
- нажать комбинацию клавиш <Alt> + <F11>.

Чтобы закрыть редактор VBA и вернуться в рабочую книгу, достаточно закрыть главное окно или выполнить команду меню **Файл** → **Закреть** и вернуться в Microsoft Excel.

Главное окно редактора обычно занимает весь экран (рис. 5.3). В окне имеются строка заголовка, меню и панель инструментов. В строке заголовка выводится имя текущей рабочей книги. В главном окне размещаются все другие окна редактора: Окно проекта (Project), Окно свойств (Properties), Окно формы (UserForm), Окно кода (Code).

Окно проекта активизируется выполнением команды меню **Вид/Окно проекта** (View/Project Window) или щелчком на кнопке пиктографического меню *Окно проекта* (Project Window) панели инструментов редактора . В окне проекта отображаются список проектов всех открытых рабочих книг и иерархическая структура каждого проекта (рис. 5.3).

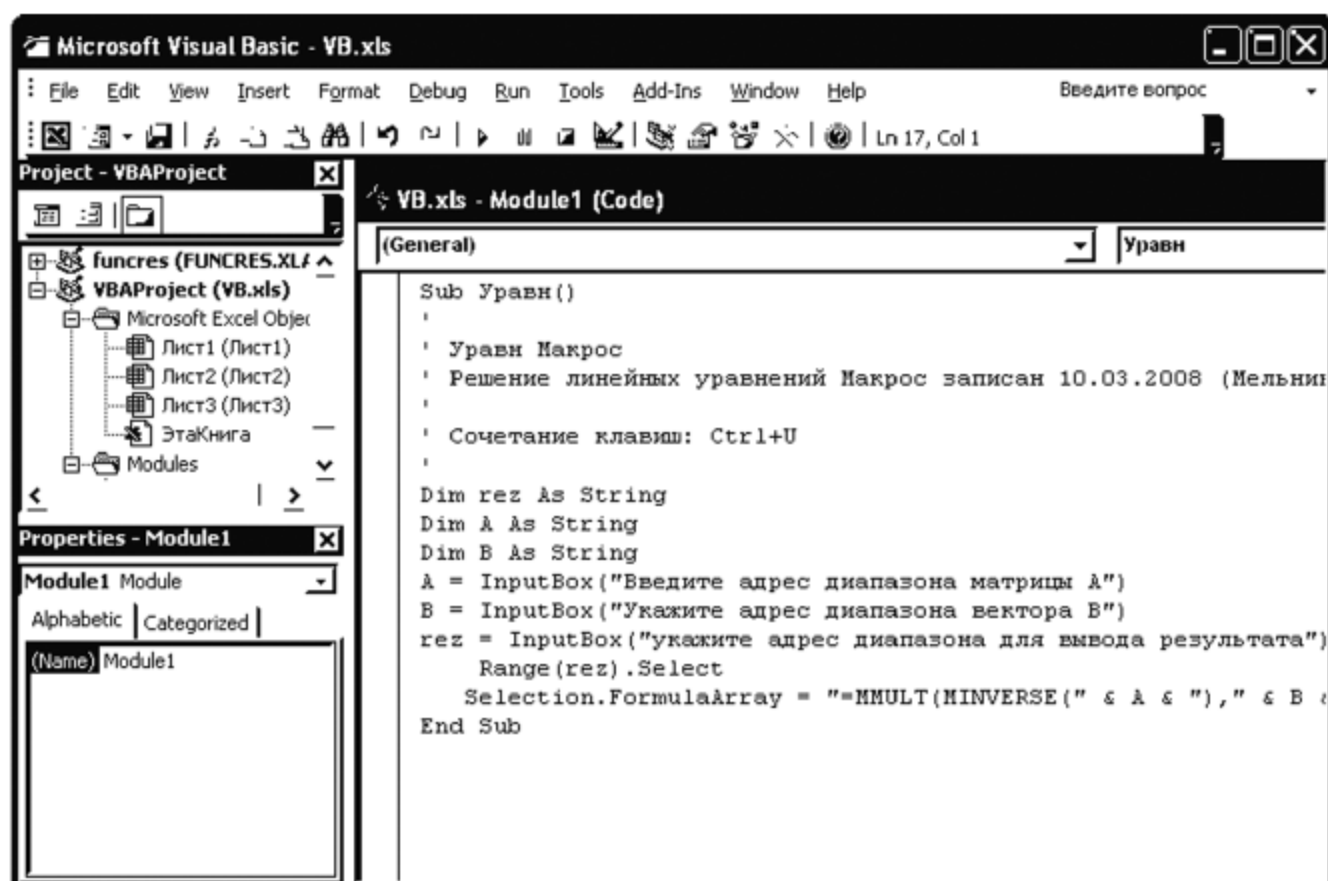


Рис. 5.3. Отображение списка проектов в главном окне редактора

В проекте автоматически создается модуль для каждого рабочего листа и для всей рабочей книги. Кроме того, модули создаются для каждой пользовательской формы, макросов и классов. По назначению модули бывают двух типов: модули объектов и стандартные. К стандартным относятся модули, содержащие макросы или функции. Стандартные модули добавляются в проект командой меню **Вставка/Мо-**

дуль (Insert/Module) или соответствующей командой контекстного меню. Стандартный модуль также может быть создан автоматически при записи макроса.

К модулям объектов относятся модули, связанные с рабочей книгой, рабочими листами, формами и модули классов.

Чтобы удалить какой-либо модуль, его нужно выделить в окне проекта и выполнить команду меню **Файл/Удалить** (File/Remove).

Благодаря тому, что в окне проекта выводятся проекты всех открытых рабочих книг, можно легко копировать программные коды из одного проекта в другой.

Окно кода выполняет функции текстового редактора для ввода и изменения кода процедур и функций проекта. Открыть окно редактирования кода для соответствующего модуля можно, если выполнить одно из следующих действий:

- в окне проекта сделать двойной щелчок на выбранном модуле;
- в окне проекта выделить модуль и выполнить команду меню

Вид/ Программа (View/Code) ;

- в окне проекта выделить модуль и нажать клавишу <F7>.

В окне редактирования можно просмотреть код отдельной процедуры или код всего модуля. Выбор режима просмотра окна редактирования производится одним из двух способов:

— выбором одной из двух кнопок, размещенных в левом нижнем углу окна редактирования кода;

— установкой или снятием флажка *Просмотр всего модуля* (Default to Full Module View) в диалоговом окне **Параметры** (Options), которое открывается при выполнении команды меню **Сервис/Параметры** (Tool/Options) (рис. 5.4).

В верхней части окна редактирования кода размещены два раскрывающихся списка. Левый список содержит имена объектов, а правый содержит перечень событий допустимых для объекта, выбранного в левом списке.

Редактор кода позволяет автоматизировать написание программных операторов, свойств и параметров. Он предлагает список компонентов, логически завершающих вводимую конструкцию. Например, при наборе кода

Dim Переменная1 As

после ввода слова As на экране отобразится список компонентов, которые могут логически завершить эту конструкцию (рис. 5.5). Если выполнить двойной щелчок на выбранном элементе списка или нажать клавишу <Tab>, то выбранное имя будет вставлено в код программы.

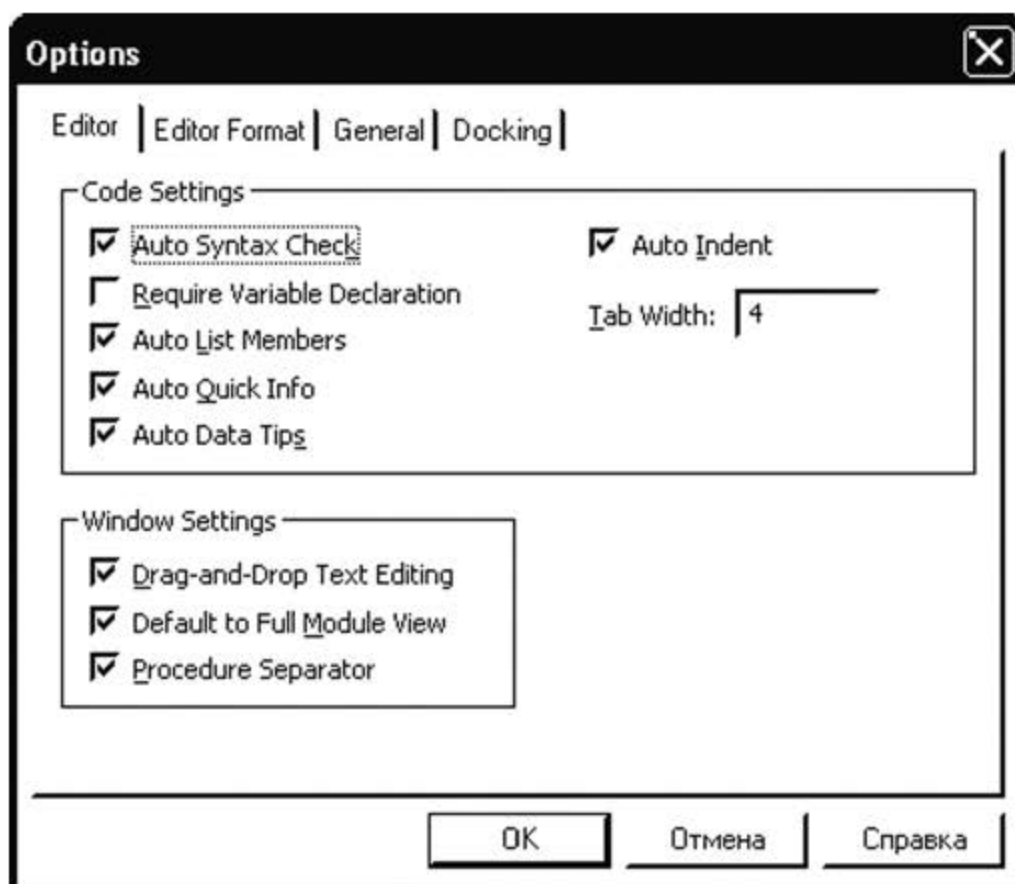


Рис. 5.4. Диалоговое окно Параметры (Options)

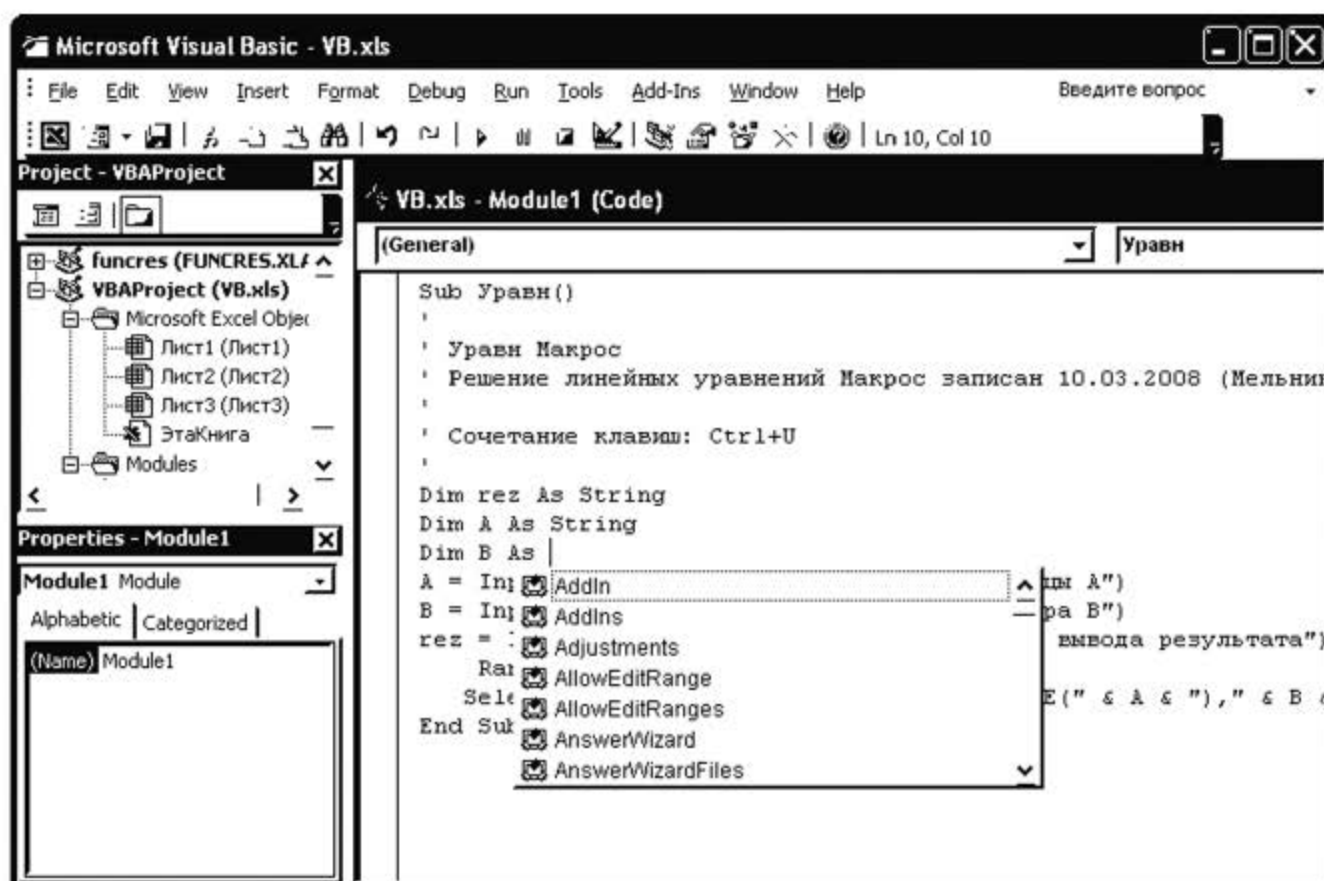



Рис. 5.5. Пример автоматизации написания программных операторов

Редактор кода, кроме перечисленных возможностей, позволяет получить информацию о ключевом слове, процедуре, функции, свойстве или методе. Для этого достаточно установить курсор на нужное ключевое слово и нажать клавишу <F1>.

Для просмотра и изменения свойств объектов проекта служит **Окно свойств**. Открыть окно свойств можно с помощью команды **Вид/Окно свойств** (View/Properties Window) или кнопки пиктографического меню  .

Чтобы просмотреть или изменить свойства объекта, можно использовать два способа:

- выбрать имя объекта в расположенном в верхней части окна кода раскрывающемся списке;
- щелкнуть на объекте в окне проекта или в форме.

После выполнения одного из этих действий в окне **Свойства** (Properties) появится список свойств объекта и их значения. Чтобы изменить значение свойства, нужно найти его в списке и изменить значение в правой колонке. Если в области значений появились три точки, то нужно щелкнуть на них, чтобы открылось диалоговое окно. Если появилась указывающая вниз стрелка, то щелчок на ней раскрывает список возможных значений.

5.3. ЯЗЫК ПРОГРАММИРОВАНИЯ VISUAL BASIC

5.3.1. Переменные и их объявление

Программный код Visual Basic представляет собой последовательность лексических единиц (лексем), записанных в соответствии с принятыми синтаксическими правилами, которая реализует некоторую семантическую конструкцию. Для обеспечения читаемости и понятности в тексте программы помещаются комментарии. В Visual Basic определен однострочный комментарий. Комментарий представляет собой последовательность любых символов, размещаемых на одной строке исходного текста программы, которая начинается со знака «'» (апостроф) или с ключевого слова Rem:

```
' Это комментарий
```

```
Rem Это тоже комментарий
```

Лексема — это единица текста программы, которая имеет определенный смысл для компилятора и которая не может быть разбита в дальнейшем. В Visual Basic различают шесть классов лексем: свободно выбираемые и используемые идентификаторы; служебные (зарезервированные) слова; константы; строки (строковые константы); операции (знаки операций); разделители (знаки пунктуации).

Посредством идентификаторов обозначают имена переменных, констант, процедур и функций. Идентификатор представляет собой последовательность букв, цифр и символов подчеркивания. Выбирая идентификатор для имени, следует учитывать два обстоятельства. Во-первых, имя должно быть содержательным, т.е. отражать назначение переменной, что делает программу более читабельной. Во-вторых, Visual Basic накладывает на имена следующие ограничения:

- имя должно начинаться с буквы;
- имя не должно содержать точки, пробела, разделительных символов, знаков операций, а также специальных символов;
- имя должно быть уникальным, оно не должно совпадать с зарезервированными словами Visual Basic или с другими именами;
- длина имени может включать до 255 символов, но следует иметь в виду, что Visual Basic учитывает только первые 31 символов от начала имени.

Примеры правильных имен:

strMyName, i, intNumOne, StrInputValue, intNumber2, strФамилия, Номер, Адрес_Организации и т.п.

Примеры неправильных имен:

2Week — имя начинается с цифры;

_Номер — имя начинается со знака подчеркивания;

Second . Week — в имени есть точка;

Dim, As, Private — эти слова являются зарезервированными;

Number One — в имени есть пробел.

В Visual Basic, как и в других языках программирования, есть зарезервированные (ключевые) слова, которые нельзя выбирать в качестве идентификаторов имен.

Объявление переменных

Переменные — это объекты, предназначенные для хранения данных. В различные моменты времени переменные могут хранить различные значения. В переменных можно запоминать какие-либо значения и извлекать их. Переменную можно представить как простейший объект программы следующим образом.

Имя переменной связывает переменную с некоторой областью памяти. Имена переменных позволяют их различать в программе, осуществлять доступ к различным участкам памяти для записи данных и их извлечения.

Переменные в программе перед их использованием нужно объявлять (декларировать). При объявлении переменной надо указать, что объявляется переменная, задать имя переменной и указать ее тип. Тип

указывает способ представления переменной. В переменных можно хранить практически любые типы данных: число, строку текста, экземпляр объекта, элементы управления и т.д. В Visual Basic различают две группы типов данных: основные (иногда их называют базовыми, или встроенными) и типы данных, определяемые пользователем.

Для эффективного использования памяти необходимо правильно выбирать тип переменной. В таблице 5.1 приведены базовые типы переменных Visual Basic, необходимая для их размещения память и диапазон возможных значений.

Таблица 5.1

Базовые типы переменных Visual Basic

Тип	Хранимая информация	Занимаемая память	Интервалы значений
Целочисленные типы			
Byte	Целые числа	1 байт	От 0 до 255
Boolean	Логические значения	2 байта	True (Истина) или False (Ложно)
Integer	Целые числа	2 байта	От -32768 до 32767
Long Integer	Длинные целые числа	4 байта	+/-2.1E9
Типы с плавающей точкой			
Single	Вещественные числа одинарной точности с плавающей точкой	4 байта	От -3.402823E38 до -1.401298E-45 для отрицательных чисел и от 1.401298E-45 до 3.402823E38 для положительных
Double	Вещественные числа двойной точности с плавающей точкой	8 байт	От -1.7976313486232E308 до -4.94065645841247E-324 для отрицательных чисел и от 4.94065645841247E-324 до 1.7976313486232E308 для положительных
Строковые типы			
String	Текстовая информация (строка фиксированной длины)	1 байт на каждый символ	От 1 до 65400
String	Текстовая информация (строка переменной длины)	10 байт + 1 байт на каждый символ	От 0 до двух миллиардов символов
Объектные типы			
Object	Рисунок или ссылка на любой другой объект	4 байта	Ссылка на объект

Тип	Хранимая информация	Занимаемая память	Интервалы значений
Типы Variant			
Variant	Значения любого из перечисленных типов данных	16 байт для чисел, 22 байта + 1 байт на каждый символ для строк	Любое числовое или строковое значение
Прочие типы			
Currency	Числа, имеющие до 15 цифр до десятичной точки и 4 цифры после нее (денежные единицы)	8 байт	От -922337203685477.5808 до 922337203685477.5808
Date	Информация о дате и времени	8 байт	От 1 января 1000 г. до 31 декабря 9999 г.
Decimal	Десятичное число	14 байт	Целое — 29 знаков Вещественное — 27 знаков после запятой

Декларация переменных может быть явной или неявной. Для явного определения переменных существует два способа. Первый (предпочтительный) способ предполагает использование следующего синтаксиса

[Static / Public / Private] Dim ИмяПеременной1[As Тип],

где Dim (Размер) — ключевое слово, которое сообщает Visual Basic, что декларируется переменная и резервируется область памяти для ее хранения;

ИмяПеременной — имя переменной (идентификатор, не входящий в перечень ключевых слов Visual Basic);

As (Как) — ключевое слово, которое сообщает Visual Basic, что определяется тип данных для переменной;

Тип — тип данных для объявляемой переменной;

Private (Частный), Public (Общий) — ключевые слова, определяющие область видимости переменной;

Static (Статический) — ключевое слово, которое определяет, сохраняет ли переменная свое значение при завершении блока программы (процедуры, функции) и выходе из него.

Следует отметить, что хороший стиль программирования предполагает использование явной декларации с помощью ключевых слов Dim, Private, Public, Static. Неявное объявление переменных без необходимости применять не следует, так как в последующем могут возникнуть непредвиденные ошибки. Для того, чтобы избежать неприятностей в случае ошибочной записи имени переменной, необходимо в общей области программного модуля помещать оператор Option Explicit. В этом случае Visual Basic будет расценивать любую неявно объявленную переменную как ошибочную.

Строковые переменные

Различают строки переменной и фиксированной длины. Строки переменной длины могут содержать до двух миллиардов символов. Когда такой переменной присваивается значение, то размер переменной изменяется так, чтобы он соответствовал длине присвоенного строкового значения.

Строка фиксированной длины — это строка постоянного размера, указанного при объявлении переменной. Если такой строке присваивается значение более длинное, то лишние символы отбрасываются. Если значение, которое присваивается, короче, то остающееся место заполняется пробелами. Строковые переменные фиксированной длины должны декларироваться явно. Синтаксис декларации следующий:

Dim VarName As String * ДлинаСтроки,

где ДлинаСтроки — целочисленная переменная или константа, содержащая число, которое указывает длину строковой переменной.

Например:

```
Dim strMyName As String* 20 ' Объявляется строковая _  
переменная фиксированной длины в 20 символов
```

или

```
Option Explicit
```

```
Dim intLen As Integer 'Объявляется целочисленная переменная  
intLen = 30 'Инициализация значения целочисленной _  
переменной
```

```
Dim MyName As String * intLen " Объявляется строковая _  
переменная длиной в 30 символов  
MyName = «Петров»
```

Константы

Константы — это объекты, значения которых остаются постоянными и не могут быть изменены во время выполнения программы.

Константы могут быть именованными и неименованными. Синтаксис языка определяет три типа констант: символьные, целые числа и вещественные числа.

Символьная константа служит для изображения отдельных знаков и представляет собой лексему, состоящую из символа (или любой последовательности символов), заключенного в кавычки. Например, «Р», «Program», «3.14», «+» — неименованные символьные константы.

Синтаксисом языка предусмотрены десятичные целые константы, шестнадцатеричные целые константы и восьмеричные целые константы. Целая десятичная константа представляется десятичным целым числом:

44, 684, 0, 1024 — неименованные десятичные целые константы.

Целая шестнадцатеричная константа представляется как последовательность шестнадцатеричных цифр, перед которой записан префикс &H:

&H16 — неименованная шестнадцатеричная константа соответствующая десятичному целому 22;

&HFF — неименованная шестнадцатеричная константа соответствующая десятичному целому 255.

Восьмеричная целая константа представляется как последовательность цифр, не содержащая десятичных цифр старше 7, которой предшествует префикс &O:

&O16 — восьмеричное представление десятичного целого 14;

&O100 — восьмеричное представление десятичного числа 64.

Вещественные константы представляются в памяти ЭВМ в форме с плавающей точкой. Каждая вещественная константа состоит из следующих частей: целая часть (десятичная целая константа); десятичная точка; дробная часть (десятичная целая константа); признак показателя «e» или «E»; показатель (десятичная целая константа): 44, 3.14159, 44e0, .314159E1.

Различают встроенные константы Visual Basic (предопределенные константы) и константы, создаваемые пользователем.

Встроенные константы используются, например, для определения цветовых наборов, задач доступа к данным, кодов клавиш, контуров и т.д. Встроенные в Visual Basic константы имеют префикс vb. Встроенные константы, которые могут использоваться для различных функций, определены в разделах справки для этих функций. Для того чтобы узнать конкретное значение константы, можно воспользоваться окном **Просмотр объектов** (Object Browser) (рис. 5.6), которое вызывается клавишей <F2> или щелчком на соответствующей кнопке

панели пиктографического меню. После выбора константы ее наименование и значение появятся в текстовом поле в нижней части окна Object Browser.

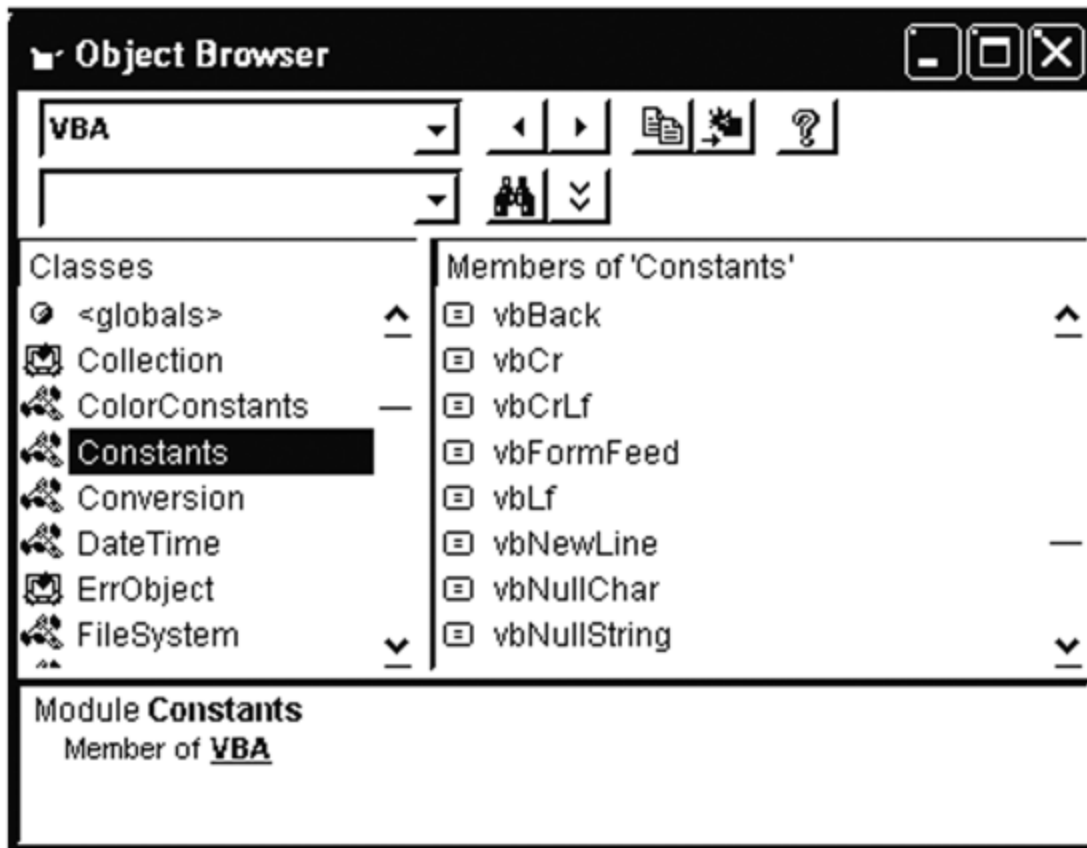


Рис. 5.6. Диалоговое окно Просмотр объектов (Object Browser)

Примеры использования predefined constants.

MsgBox «Текстовое сообщение», vbInformation

В этом примере predefined constant `vbInformation` указывает, что в окне вывода должен быть значок «Информация».

MsgBox «Нет данных для расчета», vbExclamation

В этом примере predefined constant указывает, что в окне вывода должен быть значок «Внимание».

Однако встроенного запаса констант при подготовке программы иногда бывает недостаточно. В этом случае можно создавать свои собственные именованные константы. Для определения констант служит ключевое слово `Const`. Синтаксическая конструкция для декларирования констант напоминает оператор для декларации переменной и имеет вид

`[Public/Private] Const ИМЯ_КОНСТАНТЫ [As Тип] = значение.`

Например:

`Const PI As Single = 3.1415` ' Объявлена именованная _
числовая константа для хранения значения числа *Pi*.

Области видимости переменных и констант

Переменная может быть доступна для всей программы или только для одной или нескольких ее частей. Область программы, в которой может быть использована переменная, называется областью видимости переменной. Переменные могут быть видимы в одной процедуре, в любой процедуре какой-либо формы или во всей программе. Если переменная видима, то она доступна и, следовательно, существует. Переменная может существовать и быть доступной для некоторых частей программы и при этом быть недоступной (невидимой) для других частей программы. Пример, демонстрирующий видимость переменных, приведен в листинге 3.

Листинг 3.

Option Explicit

Dim strФамилия As String * 18 'Глобальная переменная

Dim strИмя As String * 10 'Глобальная переменная

Private Sub UserForm_Initialize()

Dim strАдрес As String * 30 'Локальная переменная

strФамилия = «Петров»

strИмя = «Леонид»

strАдрес = «Москва, Кронштадская, дом 37»

Вывод strАдрес

End Sub

Sub Вывод(strD As String)

Dim strФамилия As String * 18 'Локальная переменная

strФамилия = «Это уже не Петров!»

MsgBox strФамилия

MsgBox strИмя

MsgBox strD

End Sub

В результате выполнения этой программы в окнах вывода будет следующий текст:

Это уже не Петров!

Леонид

Москва, Кронштадтская, дом 37.

Приведем пояснения: strФамилия и strИмя объявлены как глобальные переменные и существуют, пока программа выполняется. StrАдрес и strD являются локальными переменными и существуют лишь тогда, когда выполняется процедура Вывод, в которой они объявлены. В момент вызова этой процедуры глобальная переменная strФамилия существует, но является недоступной, так как ее область

видимости перекрыта одноименной локальной переменной, поэтому в окно вывода будет выдано значение той переменной strФамилия, которая объявлена в процедуре Вывод. Переменная strИмя также является глобальной, но она доступна для процедуры Вывод. Переменная strАдрес объявлена в процедуре UserForm_Initialize, но она доносит значение до окна вывода, так как передается в процедуру Вывод в качестве параметра.

Область видимости переменной задается при ее декларации одним из ключевых слов:

Dim — объявляет локальные переменные, существующие только во время вызова процедур или функций, в которых они объявлены. Если переменная объявляется в разделе глобальных объявлений модуля или формы, то она доступна для всех процедур и функций этого модуля. Для других модулей она будет не видна.

Private — не может объявлять переменную внутри процедуры или функции, при объявлении в разделе глобальных объявлений модуля **Dim** и **Private** равнозначны.

Public — объявленная таким образом переменная является глобальной на уровне приложения и доступна из всех его модулей.

Различают динамические и статические переменные. Статические переменные объявляются внутри процедуры или функции и вне их недоступны. В отличие от обычных локальных переменных они не инициализируются при входе в процедуру или функцию, где они объявлены. Для создания статической переменной необходимо при ее объявлении вместо ключевого слова **Dim** указать слово **Static**:

```
Static intNumber As Integer.
```

Декларация массивов

В Visual Basic различают два вида переменных — простые переменные и переменные структурного типа. Простые переменные служат для идентификации и резервирования памяти под одну единицу данных. Переменные структурного вида предназначены для идентификации и резервирования памяти для нескольких данных. Частным случаем переменной структурного вида является массив. Массив представляет собой структуру, все элементы которой имеют одинаковый тип. Например, это могут быть данные, определяющие вектор или матрицу. Массивы могут быть одномерными и многомерными. Так, для отображения вектора может быть использован одномерный массив, а для отображения матрицы — многомерный.

Декларация массива имеет следующий вид

[Public/ Private] Dim Имя_Массива(индексы) As Тип_Данных

где Dim — ключевое слово, указывающее, что объявляется переменная;

Имя_Массива — идентификатор, определяющий имя массива;

Индексы — значение индекса (номера) последнего элемента в массиве, считая с нулевого;

As — ключевое слово, предваряющее указание типа элементов массива;

Тип_Данных — любой, действительный для Visual Basic тип данных — базовый или созданный пользователем.

Например, декларация одномерного массива из восьми элементов выглядит следующим образом:

Dim MyArray(7) As Integer ' Одномерный массив из 8 элементов

При декларации многомерного массива в поле индекса указывается несколько индексов в соответствии с размерностью массива. Например, двумерный массив из шести столбцов и пяти строк декларируется следующим образом:

**Dim strMyArray(4,5) As String ' Двумерный массив _
из 5 строк и 6 столбцов**

По умолчанию значение нижней границы массива при таком объявлении равно нулю. В этом случае говорят, что 0 — базовый индекс массива. При необходимости базовый индекс можно изменить путем использования ключевого слова To при объявлении массива:

Dim MyArray (3 To 10) As String

В этом примере базовому индексу массива установлено значение 3. Подобным образом можно устанавливать как положительные, так и отрицательные базовые индексы:

Dim MyArray (-3 To 4) As String

Иногда в процессе выполнения программы размер массива требуется изменить. В этом случае первоначально массив декларируют как динамический. Для этого в декларации не указывается размерность, например:

Dim MyArray () As String

Количество элементов в динамическом массиве и его размерность в процессе выполнения программы можно переопределить с помощью ключевого слова ReDim. Синтаксическая конструкция переопределения массива имеет вид

ReDim [Preserve] ИмяМассива(индексы) [As ТипДанных],

где ReDim — ключевое слово, указывающее, что переопределяются размеры массива;

Preserve — необязательное ключевое слово, с помощью которого дается указание, чтобы все элементы переопределяемого массива сохранили свое значение;

индексы — размерности массива (до 60).

Пример:

Dim MyArray () As String 'Декларация динамического массива

Dim ValArray As Integer 'Декларация переменной, _

для хранения размерности массива

ValArray=9 'Инициализация значения

ReDim strMyArray (intValArray) 'Одномерный массив _

из 10 элементов (базовый индекс равен 0)

ReDim MyArray (3 To ValArray, 1 To ValArray) ' Двумерный _ массив

с базовыми индексами, отличными от нуля

5.3.2. Операторы, выражения и операции

Строка с кодом в исходном тексте программы Visual Basic называется программным оператором. Программный оператор — это неделимое предложение, выполняющее какое-либо действие. Он может представлять собой любую комбинацию ключевых слов Visual Basic, свойств, функций, операций и символов, совокупность которых представляет собой корректную конструкцию, распознаваемую Visual Basic. Завершенный программный оператор может состоять из единственного ключевого слова, например «Веер», или же комбинаций элементов, например как следующий оператор, присваивающий значение системного времени свойству Caption (надпись) объекта Label1:

Label1.Caption = Time.

Правила, применяемые при построении программных операторов, называются синтаксисом.

Программный оператор может включать выражения. Выражение (expression) — это комбинация знаков операций и операндов, а также скобки. Назначение любого выражения — получение некоторого значения. Синтаксическая конструкция выражения может быть представлена в виде

Операнд1 [операция Операнд2 [операция Выражение]].

Например, вычисление площади круга может быть записано выражением

$(3.14159 * D^2)/4.$

Так как результатом вычисления выражения является некоторое значение, то в программном операторе выражение может быть представлено непосредственно значением, например:

Pi = 3.14159.

В зависимости от типа формируемых значений определяются типы выражений. Например, если значениями выражения являются целые и вещественные числа, то говорят об арифметических выражениях.

Для формирования и последующего вычисления выражений служат операции. Для записи операций Visual Basic имеет знаки операций, которые воспринимаются компилятором как отдельные лексемы. Каждая операция имеет свой приоритет (ранг). Операции ранга 1 имеют наивысший приоритет и в программном операторе выполняются первыми. Операции одного ранга в выражениях выполняются в соответствии с правилами ассоциативности (слева направо или наоборот).

Операция присваивания

При декларации переменной происходит связывание имени переменной с областью памяти, в которой будет храниться ее значение. Однако это значение после декларации может оказаться произвольным. Для того чтобы присвоить переменной нужное значение, используется операция присваивания. Присваивание имеет следующую синтаксическую конструкцию:

ИмяПеременной = Выражение,

где ИмяПеременной — имя переменной (идентификатор);
символ «=» — знак операции присваивания.

Выражение может быть значением (например, числом) либо комбинацией переменных, констант и функций, связанных знаками операций.

Математические операции

Математические операции применяются для записи формул. Формула представляет собой программный оператор, содержащий числа, переменные, операторы и ключевые слова или же комбинацию этих элементов и вычисляющий новое значение. Список математических операций Visual Basic и их рангов приведен в табл. 5.2.

Таблица 5.2

Математические операции Visual Basic

Операция	Математическое действие	Ранг (приоритет)
[Операнд1] + [Операнд2]	Сложение	7
[Операнд1] - [Операнд2]	Вычитание	7
-[Операнд1]	Изменение знака числа	3
[Операнд1] * [Операнд2]	Умножение	4
[Операнд1] / [Операнд2]	Деление	4
[Операнд1] \ [Операнд2]	Целая часть от деления	5
[Операнд1] Mod [Операнд2]	Остаток от деления	6
[Операнд1] ^ [Операнд2]	Возведение в степень	2

Операции сложения, вычитания, умножения и деления называют основными математическими операциями, и дополнительных пояснений они не требуют.

Остальные математические операции называются дополнительными. Они применяются в специальных математических формулах и при обработке текстовой информации. Для пояснения, как они работают, рассмотрим примеры:

Result = 10 \ 3 — результат 3 (целая часть от деления);

Result = 10 Mod 3 — результат 1 (остаток от деления);

Result = 3^2 — результат 9;

Result = 9^0.5 — результат 3;

Result = 2^-2 — результат 0.25.

Если в программном операторе некоторые выражения заключены в круглые скобки, то такие выражения вычисляются первыми (скобки являются операциями с приоритетом 1). Например:

Result = (25+45)*10^2.

В этом программном операторе в первую очередь будет вычислено выражение в скобках (25+45), затем число 10 будет возведено в степень 2 и только после этого будет выполнено умножение значения результата вычисления в скобках на значение результата возведения в степень. Результат вычисления — 7000.

Операции отношения

В отличие от математических операций, результатом выполнения которых может быть любое значение, операция отношения может иметь только два результирующих значения: True (Истина) и False (Ложь), которые могут быть присвоены переменным типа Boolean или определенному свойству объекта. Перечень операций отношения Visual Basic приведен в табл. 5.3.

Таблица 5.3

Операции отношения Visual Basic

Операция	Описание
[Операнд1] = [Операнд2]	Равно. Результат — True, если первый операнд равен второму
[Операнд1] <> [Операнд2]	Не равно. Результат — True, если первый операнд не равен второму
[Операнд1] > [Операнд2]	Больше. Результат — True, если первый операнд больше второго
[Операнд1] < [Операнд2]	Меньше. Результат — True, если первый операнд меньше второго

Операция	Описание
[Операнд1]=> [Операнд2]	Больше или равно. Результат – True, если первый операнд больше или равен второму
[Операнд1] <= [Операнд2]	Меньше или равно. Результат – True, если первый операнд меньше или равен второму
[Строка] Like [Маска]	Соответствие маске. Результат – True, если строка соответствует маске
[Операнд1] Is [Операнд2]	Ссылка на объект. Результат – True, если обе переменные ссылаются на один и тот же объект

Операции отношения применяются для записи выражений условия, результатом выполнения которых являются значения True (Истина) или False (Ложь). Примеры использования операций отношения приведены в табл. 5.4.

Таблица 5.4

Примеры использования операций отношения

Выражение условия	Результат
25 <> 30	True (Истина) (25 не равно 30)
25 < 30	True (Истина) (25 меньше 30)
25 > 30	False (Ложь) (25 не больше 30)
Text1.Text = " Петров"	True (Истина), если слово «Петров» является содержимым первого текстового поля, в противном случае – False (Ложь)
Number >= 100	True (Истина), если переменная Number содержит значение не меньше 100, в противном случае – False (Ложь)

Логические операции

Логические операции применяются в логических выражениях. Если существует несколько условий выбора в операциях отношения, то эти операции связываются между собой логическими операциями. Логические операции Visual Basic приведены в табл. 5.5, примеры применения – в табл. 5.6.

Таблица 5.5

Логические операции Visual Basic

Логическая операция	Действие
[Операнд1] And [Операнд2]	Если связываемые условия имеют значение True, то результирующее значение также будет True
[Операнд1] Or [Операнд2]	Если одно из связываемых условий будет иметь значение True, то результирующим значением также будет True

Логическая операция	Действие
Not [Операнд]	Если условие имеет значение True, то результирующим значением будет False. Если условие имеет значение False, то результирующее значение будет True
[Операнд1] Xor [Операнд2]	Если только одно из связываемых условий имеет значение True, то результирующее значение также будет True. Если оба условия имеют одинаковые значения, то результирующее значение будет False

Таблица 5.6

Примеры применения логических операций

Логическое выражение	Результат
TextBox1.Text = "Иванов" And Number < 20	True (Истина), если оба выражения сравнения имеют значение True (Истина)
TextBox1.Text = "Иванов" Xor Number < 20	False (Ложь), если оба выражения сравнения имеют значение True (Истина)
TextBox1.Text = "Иванов" Or Number < 20	True (Истина), если хотя бы одно выражение сравнения имеет значение True (Истина). False (Ложь), если оба выражения имеют значение False (Ложь)
Not Number < 20	True (Истина), если выражение сравнения имеет значение False

Операции для работы со строками

В Visual Basic есть только одна операция для работы со строками — это операция конкатенации. Конкатенация позволяет объединить значения двух или нескольких строковых переменных или строковых констант. Знаком операции конкатенации является символ амперсанд (&). При конкатенации строк значение второй строки добавляется в конец первой строки. Результатом операции является более длинная строка, составленная из исходных строк.

Выражение с применением операции конкатенации имеет следующий вид

`strВыражение_1 & strВыражение_2 б . . . & strВыражение_Nс,`

где `strВыражение` — строковые выражения, которые могут быть любыми допустимыми строками (строковыми переменными, строковыми константами или функциями обработки строк). Знак амперсанда между строковыми выражениями указывает, что производится конкатенация этих выражений. Знак амперсанда отделяется от выражения пробельным символом. В одном операторе можно объединять любое количество строковых выражений.

Примеры:

`strResult = «Студент» & «Иванов»` — Здесь объединяются две неименованные строковые константы. Результатом операции конкатенации будет значение «Студент Иванов».

Тот же результат будет получен при использовании следующего кода:

```
Dim strName As String
```

```
Dim strResult As String
```

```
strName = «Иванов»
```

```
strResult = «Студент» & strName
```

Здесь в операции конкатенации участвуют неименованная строковая константа и строковая переменная.

Объединение строк может быть также выполнено с помощью операции сложения (знак операции «+»). В этом случае результат операции зависит от типа операндов. Например, если операнды являются числовыми типами, то будет выполняться арифметическая операция сложения, если операнды строкового типа, то выполняется их объединение.

Операторы управления

Операторы управления языка служат для управления работой программы. К ним относятся:

- операторы передачи управления;
- операторы выбора;
- операторы циклов.

Операторы передачи управления применяются в программе для реализации безусловных алгоритмических конструкций. Они выполняют переход с одного участка программы на любой другой без какого-либо условия. Оператор перехода имеет следующий вид

`GoTo идентификатор,`

где `GoTo` — ключевое слово;
идентификатор — одна из меток программы.

Метка — это идентификатор, помещаемый слева от программного оператора и отделенный от него двоеточием. Например:

```
m1: Text1.Text = «это метка»
```

Оператор перехода для перехода к оператору с меткой `m1` будет иметь вид

`GoTo m1.`

Следует сказать о том, что для получения хорошего стиля программирования следует избегать применения оператора `GoTo`, так как в этом случае ухудшаются читаемость и понимание программы.

Операторы выбора используются в программе для реализации условных алгоритмических конструкций, которые вызывают выполнение различных частей программы в соответствии с условиями, существующими на момент выполнения этих операторов. Одним из важных элементов программного оператора выбора является выражение условия, значением которого может быть Истина (True) или Ложь (False). Такие выражения условия записываются с помощью операторов отношения.

В Visual Basic есть два основных типа операторов выбора: операторы условия If ... Then и переключатели, реализуемые оператором Select Case.

Операторы If . . . Then

В Visual Basic есть два типа операторов If . . . Then — линейный и блочный.

Линейный оператор If . . . Then используется для того, чтобы выполнить какой-либо один оператор, если некоторое условие будет истинным. Условие является выражением или функцией, истинность которого оценивается. Синтаксическая конструкция линейного оператора имеет две формы — безальтернативную и альтернативную:

Безальтернативная форма:

If условие Then Выражение

Альтернативная форма:

If условие Then Выражение_1 Else Выражение_2

В безальтернативной форме при значении условия True сначала выполняется выражение, следующее за ключевым словом Then, а затем следующий в последовательности оператор. Если условие принимает значение False, то выражение, следующее за словом Then, не выполняется, а выполняется следующий оператор.

Примеры линейного оператора If . . . Then безальтернативной формы:

If intNumber < 3 Then intResult = 10 * 2 ' Если переменная _
intNumber содержит значение меньше трех, то переменной _
intResult будет присвоено значение 20

If optAddition.Value = True Then z = x + y

Пример линейного оператора If . . . Then . . . Else:

If intNumber < 3 Then intResult = 10 * 2 Else intResult = 30

В результате выполнения этого оператора переменной intResult будет присвоено значение 20, если переменная intNumber содержит значение меньше трех, в противном случае переменной intResult будет присвоено значение 30.

Блочный оператор If . . . Then используется тогда, когда в случае истинности условия необходимо выполнить несколько программных операторов (блок операторов). Блочный оператор также может быть безальтернативным и альтернативным. Структура безальтернативного блочного оператора следующая:

```
If Условие Then
  Программный оператор 1
  Программный оператор 2
  . . .
  Программный оператор n
End If
```

Здесь End If указывает на окончание блока оператора If.

Альтернативный блочный оператор If применяется в тех случаях, когда при выполнении условия необходимо выполнить один набор программных операторов, а при невыполнении — другой. Это реализуется с помощью оператора If . . . Then . . . Else, синтаксис которого следующий:

```
If условие Then
  Блок программных операторов, выполняемых при значении условия True
Else
  Блок программных операторов, выполняемых при значении условия False
End If
```

Переключатели

Переключатели в Visual Basic реализуются оператором Select Case, который позволяет сделать выбор из нескольких альтернативных вариантов в зависимости от значения условного выражения. Синтаксис его следующий:

```
Select Case Выражение
  Case Значение_1
    Оператор, выполняемый при совпадении Значения_1 и значения Выражения.
  Case Значение_2
    Оператор, выполняемый при совпадении Значения_2 и значения Выражения.
  . . .
  Case Значение_N
    Оператор, выполняемый при совпадении Значения_N и значения Выражения.
End Select
```

В операторе Select Case можно использовать операции отношения. Для этого надо включить в выражение ключевое слово Is или ключевое слово To. Ключевое слово Is дает указание компилятору сравнить значение проверяемой переменной со значением выражения, следующего за ключевым словом Is. Ключевое слово To задает диапазон значений. Например, следующий код фрагмента программы (листинг 4), использующий оператор Select Case, позволяет выставить оценку исходя из общего количества набранных баллов.

Листинг 4.

```
Select Case ПравильныйОтвет.Text
Case 8 To 10 ' Если сумма баллов в диапазоне 8—10
Балл.Text = «Отлично»
Case 6 To 7 ' Если сумма баллов в диапазоне 6—7
Балл.Text = «Хорошо»
Case 4 To 5 ' Если сумма баллов в диапазоне 4—5
Балл.Text = «Удовлетворительно»
Case Is < 4 ' Если сумма баллов меньше 4
Балл.Text = «Неудовлетворительно»
End Select
```

Программирование циклов

Цикл — это оператор или группа операторов, которые программа многократно выполняет до тех пор, пока не получит команду начать выполнение чего-либо другого. В Visual Basic существуют два основных типа циклов: циклы со счетчиком (с известным числом повторений) и циклы с условием. Циклы со счетчиком используют в тех случаях, когда необходимо выполнить некоторые действия определенное число раз. Циклы с условием применяются тогда, когда некоторые действия в программе должны повторяться до тех пор, пока выполняется определенное условие, или до тех пор, пока не будет выполнено определенное условие.

Циклы со счетчиком (с известным числом повторений) в Visual Basic еще называют циклами For, или циклами For ... Next. Так они называются потому, что начало и конец этих циклов определяются операторами For и Next. Синтаксис цикла For. . Next таков:

```
For СчетчикЦикла= НачальноеЗначение To КонечноеЗначение [Step
Шаг]
операторы
[Exit For]
Next [СчетчикЦикла],
```


где For — ключевое слово Visual Basic, обозначающее начало цикла;
СчетчикЦикла — переменная, определенная в качестве счетчика цикла;
НачальноеЗначение — число, задающее начальное значение счетчика;
To — ключевое слово Visual Basic, разделяющее НачальноеЗначение и КонечноеЗначение — число, задающее значение счетчика, при котором цикл завершается;
Step — ключевое слово Visual Basic, используемое для задания шага цикла, необязательный аргумент;
Шаг — число, задающее шаг цикла, т.е. значение, на которое увеличивается (или уменьшается) значение счетчика на каждом шаге, это число может быть отрицательным;
Exit For — оператор досрочного выхода из цикла (необязательный);
Next — ключевое слово Visual Basic, обозначающее конец цикла.

В начале цикла For ...Next определяется переменная-счетчик, а также начальное и конечное значения этой переменной. В самом начале выполнения цикла переменная-счетчик устанавливается в начальное значение. Каждый раз, когда программа пройдя через цикл, возвращается к его началу, значение счетчика увеличивается. Если используется ключевое слово Step, то переменная-счетчик изменяется в соответствии с числом, указанным после ключевого слова Step. Например:

```
For I = 0 To 10 Step 2 “ Значение I будет увеличиваться на 2
```

Если ключевое слово Step отсутствует, то значение шага равно единице.

Каждый раз, когда значение переменной-счетчика изменяется, оно сравнивается с заданным конечным значением счетчика. Если значение счетчика превышает конечное значение, программа сразу выходит из цикла и переходит к той строке кода, которая следует за циклом.

Пример:

```
Option Explicit  
Dim I As Integer  
For I = 1 To 4  
Sum = Sum + 2  
Next I
```

Этот цикл эквивалентен написанию четырех операторов Sum = Sum + 2 в тексте программы.

Цикл For . . . Next Может быть прерван досрочно, например, при достижении какого-либо условия. Для этого в нужном месте цикла нужно поместить оператор Exit For. Например:

```
Option Explicit  
Dim Sum As Integer
```

```

Dim j As Integer
Sum = 2
For j = 1 To 10
Sum = Sum + j
If Sum > 6 Then
Exit For ' Выход из цикла, если значение Sum больше 6
End If
Next j
TextBox1.Text = Sum

```

В этом примере цикл прерывается досрочно, когда значение переменной Sum будет больше 6. В результате выполнения этого кода в текстовом поле будет выведено вычисленное значение переменной Sum, равное 8.

Главной особенностью циклов с условием является условие, которое может быть любым выражением, принимающим значение True (Истина) или False (Ложь). В Visual Basic есть два основных цикла с условием: цикл Do While ... Loop и цикл Do Until ... Loop. Оба они могут быть с предусловием или с постусловием.

Циклы Do While | Until имеет следующий синтаксис:

Цикл с предусловием	Цикл с постусловием
Do While Until Выражение	Do
Операторы	Операторы
[Exit Do]	[Exit Do]
Loop	Loop While Until Выражение

где Do — ключевое слово;

While | Until — ключевые слова, указывающие тип цикла;

Выражение — выражение условия, принимающее значение True или False;

Loop — ключевое слово, указывающее на окончание цикла.

Цикл Do While выполняется до тех пор, пока выражение условия имеет значение True.

Пример цикла Do While, реализующего алгоритм программы, аналогичный приведенной в примере (листинг 5) для цикла For с досрочным прерыванием.

Листинг 5.

```

Option Explicit
Dim Sum As Integer
Dim j As Integer
Sum = 2

```

```
Do While Sum < 7
Sum = Sum + j
j = j + 1
Loop
TextBox1.Text = Sum
```

В результате выполнения этого кода в текстовом поле будет выведено вычисленное значение переменной Sum, равное 8.

В отличие от цикла Do While цикл Do Until выполняется до тех пор, пока выражение условия имеет значение False.

Пример цикла Do Until, реализующего алгоритм программы аналогичный приведенному выше.

Листинг 6.

```
Option Explicit
Dim Sum As Integer
Dim j As Integer
Sum = 2
Do Until Sum > 7
Sum = Sum + j
j = j + 1
Loop
TextBox1.Text = Sum
```

В результате выполнения этого примера в текстовом поле будет выведено вычисленное значение переменной Sum, равное 8.

Иногда бывает необходимо прервать цикл Do ... Loop, если выполняется какое-либо дополнительное условие. Это может быть сделано с помощью оператора Exit Do. Например:

Листинг 7.

```
Option Explicit
Dim Sum As Integer
Dim j As Integer
Sum = 2
Do Until Sum > 7
Sum = Sum + j
j = j + 1
If j > 3 Then
Exit Do ' Досрочный выход из цикла Do ... Loop
End If
Loop
TextBox1.Text = Sum
```

В этом примере цикл с условием досрочно прерывается, если выполняется дополнительное условие $j > 3$. В результате выполнения программного кода переменная Sum будет иметь значение, равное 8.

Используя цикл с условием можно организовать бесконечный цикл. Иногда это бывает необходимо, а иногда это происходит из-за ошибки пользователя. Для создания бесконечного цикла условное выражение должно быть таким, которое никогда не выполняется или выполняется всегда. Например:

Do Until 1

Операторы

Loop

Выйти из такого бесконечного цикла и прервать работу программы можно, нажав комбинацию клавиш <Ctrl> + <Break>.

5.3.3. Встроенные диалоговые окна Visual Basic

Окно вывода

Встроенные диалоговые окна представляют собой операторы или функции языка. К ним относятся окно вывода (MessageBox) и окно ввода (InputBox). Окна диалогов используют системные функции Windows. Поэтому надписи на кнопках, названия полей и другие элементы окон диалога могут отображаться на экране в английском или русском варианте, в зависимости от версии и настроек Windows.

Окно сообщения создается функцией MsgBox, которая имеет следующий синтаксис:

MsgBox(prompt[, buttons] [, title] [, helpfile, context])

Функция возвращает значение, соответствующее выбранной пользователем кнопке в окне сообщения. В таблице 5.7 приведены значения и имена возвращаемых констант. Имена можно использовать вместо значений в любом месте программы.

Таблица 5.7

Значения и имена возвращаемых констант

Константа	Значение	Нажатая кнопка
vbOK	1	ОК
vbCancel	2	Отмена (Cancel)
vbAbort	3	Прервать (Abort)
vbRetry	4	Повторить (Retry)
vbIgnore	5	Пропустить (Ignore)
vbYes	6	Да (Yes)
vbNo	7	Нет (No)

Параметр `prompt` обязательный. Это строка, которая выдается в окне сообщения. Ее длина ограничена 1024 символами.

Параметр `buttons` необязательный. Значение параметра — целое число, равное сумме значений, определяющих набор кнопок, коды значков, кнопки по умолчанию в окне сообщения, а также модальность окна. Возможные значения описаны в табл. 5.8. По умолчанию значение параметра равно 0.

Параметр `title` задает строку, которая является заголовком окна сообщения. Если параметр отсутствует, то в качестве заголовка используется имя приложения.

Функция окна вывода может иметь еще два необязательных параметра (`helpfile` и `context`), касающихся справочной информации, относящейся к данному сообщению. О них можно прочитать в справочной системе.

Таблица 5.8

Константы для определения параметра `buttons`

Константа	Значение	Описание
Наборы кнопок окна сообщения		
<code>VbOKOnly</code>	0	Окно содержит только кнопку ОК
<code>VbOKCancel</code>	1	Окно содержит кнопки ОК и Cancel (Отмена)
<code>vbAbortRetryIgnore</code>	2	Окно содержит кнопки Abort (Прервать), Retry (Повторить) и Ignore (Пропустить)
<code>vbYesNoCancel</code>	3	Окно содержит кнопки Yes (Да), No (Нет) и Cancel (Отмена)
<code>vbYesNo</code>	4	Окно содержит кнопки Yes (Да) и No (Нет)
<code>vbRetryCancel</code>	5	Окно содержит кнопки Retry (Повторить) и Cancel (Отмена)
Пиктограммы окна сообщения		
<code>vbCritical</code>	16	Добавляет пиктограмму «Критическое сообщение». Часто после такого сообщения программа прекращает работу
<code>vbQuestion</code>	32	Добавляет пиктограмму «Запрос». Обычно используется, когда для продолжения работы программы требуется дополнительная информация
<code>vbExclamation</code>	48	Добавляет пиктограмму «Предупреждение»
<code>vbInformation</code>	64	Добавляет пиктограмму «Информация». Чаще всего используется для сообщения о завершении выполнения некоторой задачи
Кнопка по умолчанию		
<code>vbDefaultButton1</code>	0	Первая

Константа	Значение	Описание
vbDefaultButton2	256	Вторая
vbDefaultButton3	512	Третья
vbDefaultButton4	768	Четвертая
Модальность окна		
vbApplicationModal	0	Модальность уровня приложения. Пока вы не нажмете одну из кнопок окна, вы не сможете вернуться в приложение, породившее это окно. При этом можно переключиться на другое приложение
vbSystemModal	4096	Модальность системного уровня. Пока не будет закрыто это окно, никакие приложения не доступны

Например, в результате выполнения оператора

```
Ans = MsgBox(«Закончить?», vbYesNo + vbQuestion + _
vbDefaultButton1, «Пример окна MsgBox»)
```

появится окно вывода, как на рис. 5.7.

Чтобы определить, какая кнопка была нажата, значение переменной Ans анализируется с помощью оператора If: Например:

```
If Ans = vbYes then UserForm1.Hide
```

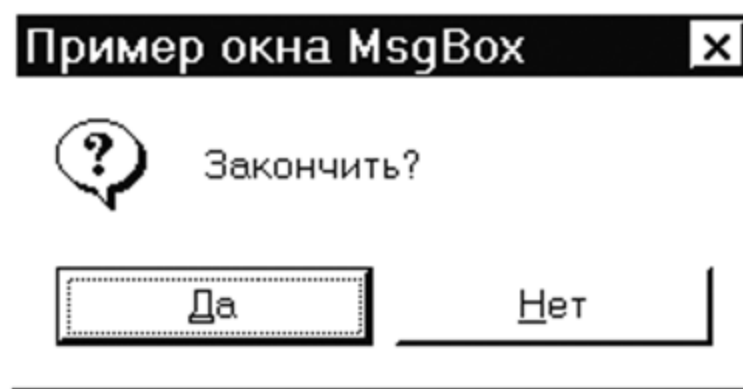


Рис. 5.7. Пример окна вывода

Окно Ввода

Окно ввода служит для ввода данных. Оно создается функцией InputBox(), имеющей следующий синтаксис:

```
InputBox(prompt[, title] [, default] [, xpos] [, ypos])
```

Окно содержит сообщение, указывающее, какие данные должен ввести пользователь, поле текста для ввода данных и две кнопки ОК и Отмена, которые используются для подтверждения или отмены ввода данных. Закончив ввод данных, пользователь должен щелкнуть на одной из кнопок. Если щелчок был сделан на кнопке ОК, то значением функции является текст, находящийся в поле ввода. Если щелчок был

сделан на кнопке Отмена, то значением функции является пустая строка, независимо от того, что ввел пользователь.

Параметры функции имеют следующий смысл:

- `prompt` – строка сообщения, которая будет напечатана в окне;
- `title` – строка, которая является заголовком окна. Если параметр не указан, то в качестве заголовка используется имя приложения;

- `default` – строка, помещаемая в текстовое поле. Если параметр не указан, то поле текста будет пустым;

- `xpos` – расстояние в твипах от левой границы экрана до левой границы окна. Если параметр не указан, окно центрируется по горизонтали;

- `ypos` – расстояние в твипах от верхней границы экрана до верхней границы окна.

Окно ввода может иметь еще два необязательных параметра (`helpfile` и `context`). О них можно прочитать в справочной системе.

Для преобразования введенной строки в другой тип данных используйте функции преобразования типов: `CCur()`, `CDate()`, `CInt()`, `CLng()`, `CSng()`, `CVar()` и др.

В качестве примера на рис. 5.8 приведено окно, созданное с помощью оператора:

```
A = InputBox(«Введите значение:», «Пример окна InputBox»)
```

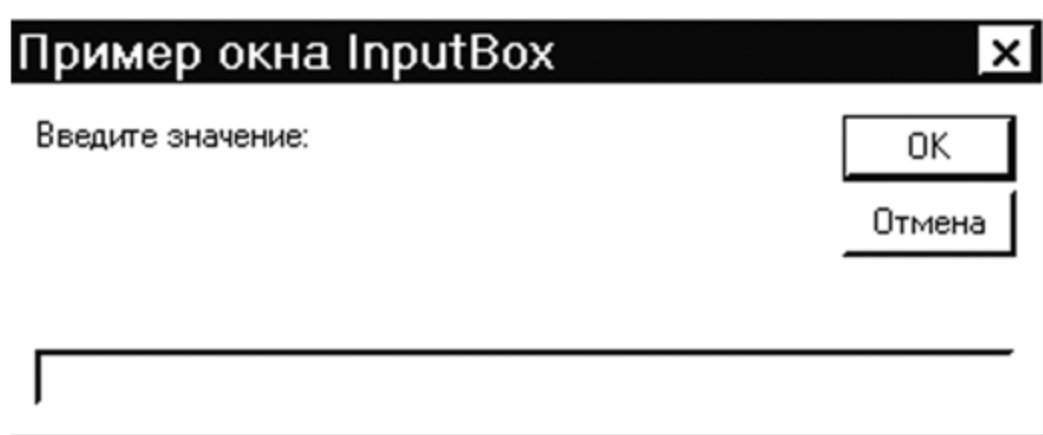


Рис. 5.8. Пример окна ввода

5.4. РЕДАКТИРОВАНИЕ МАКРОСОВ

Вернемся к макросу, созданному и измененному ранее, код которого приведен в листинге 8. Рассмотрим назначение каждой его инструкции.

Листинг 8.

```
Sub Уравн()
```

```
“
```

```
“ Уравн Макрос
```

```
“ Решение линейных уравнений Макрос записан 10.03.2008 (Мельников)
```

```
“
```

```
“ Сочетание клавиш: Ctrl+U
```

```
“
```

```
Dim rez As String
```

```
Dim A As String
```

```
Dim B As String
```

```
A = InputBox(«Введите адрес диапазона матрицы A»)
```

```
B = InputBox(«Укажите адрес диапазона вектора B»)
```

```
rez = InputBox(«укажите адрес диапазона для вывода результата»)
```

```
Range(rez).Select
```

```
Selection.FormulaArray = «=MMULT(MINVERSE(« & A & «),» & B & «)»
```

```
End Sub
```

Макрос представляет собой подпрограмму Sub. В подпрограмме декларированы три строковые переменные:

- Rez — для хранения адреса диапазона, в который будет выводиться результат;
- A — для хранения адреса диапазона, в котором записаны значения матрицы A;
- B — для хранения адреса диапазона, в котором содержатся элементы вектора B.

Значения перечисленным переменным присваиваются посредством окон ввода InputBox, которые поочередно открываются после запуска макроса.

Инструкция Range(rez).Select выполняет выделение области ячеек рабочего листа, в которую будет записана формула для вычисления и помещен результат.

Инструкция Selection.FormulaArray = предписывает поместить формулу, записанную справа от знака «=», в выделенный диапазон.

Строковая константа «=MMULT(MINVERSE(« & A & «),» & B & «)» — формула Excel, помещаемая в ячейки рабочего листа.

5.5. СОЗДАНИЕ ФУНКЦИЙ РАБОЧЕГО ЛИСТА

Функции рабочего листа — это функции, которые могут быть помещены в ячейки рабочего листа с помощью инструмента **Мастер функций**. В проекте VBA такие функции создаются в отдельных модулях.

Для создания функции пользователя нужно выполнить следующие действия:

- активизировать редактор Visual Basic;
- если в проекте нет модуля, то создать его, выполнив команду меню редактора **Вставка/Модуль** (Insert/Module);
- выполнить команду меню редактора VB **Вставка/Процедура** (Insert/Procedure);
- в открывшемся диалоговом окне **Добавить процедуру** (Add procedure) установить переключатель *Функция* (Function);
- в поле *Имя* (Name) ввести имя функции;
- установить соответствующий переключатель *Область определения* (Scope);
- щелкнуть на ОК. После выполнения этих действий в окне модуля появится заготовка функции (заголовок и окончание), между которыми нужно поместить код тела функции;
- ввести список параметров функции, их типов данных, а также указать тип возвращаемого функцией значения;
- используя команду меню **Вид/Просмотр объектов** (View → Object Browser) или нажав клавишу <F2>, вызвать окно **Просмотр объектов** (Object Browser);
- раскрыть верхний левый список *Список проектов* и выбрать из него *VBA Project*. В окне **Классы** (Classes) отобразятся элементы текущего проекта;
- выбрать в этом окне модуль, в котором создана функция, — в окне **Компоненты** (Members) отобразятся элементы, которые содержатся в этом модуле;
- выделить в окне **Компоненты** (Members) элемент с именем созданной функции и включить контекстное меню;
- выполнить команду контекстного меню **Свойства** (Properties) — откроется окно **Параметры компонента** (Member Options);
- в поле *Описание* (Description) этого окна ввести текст краткого описания функции, если необходимо, то указать файл справки и идентификатор.

Приведем пример создания функции пользователя с линейным алгоритмом.

Пример 5.1. Требуется разработать функцию пользователя для расчета накопленной стоимости. Предусмотреть проверку корректности исходных данных.

Модель задачи

Накопленная стоимость по выданному кредиту (или по вкладу) вычисляется в соответствии с формулой

$$S = P(1 + iT),$$

где S – накопленное значение;
 P – величина кредита;
 i – номинальная процентная ставка;
 T – срок (в месяцах).

Разработка

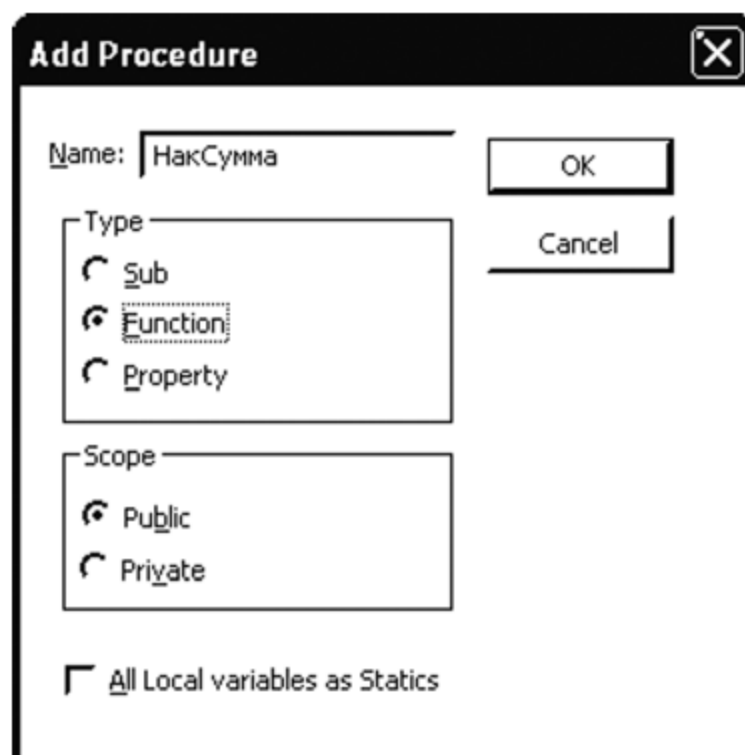
Прежде всего необходимо определить, какие данные являются исходными для решения задачи и какие у этих данных типы.

Для нашей задачи исходными данными являются величина кредита, номинальная процентная ставка и срок кредита. Удобно данные, их идентификаторы и типы представить в виде следующей таблицы.

№ п/п	Описание переменной	Идентификатор	Тип
1	Величина кредита	ИсходнаяСумма	Sting
2	Процентная ставка	Ставка	Sting
3	Срок	Срок	Sting
4	Имя функции	НакСумма	Double

Для создания функции выполним действия:

– выполним команду меню редактора Visual Basic **Вставка/Модуль** (Insert/Module), а затем **Вставка/Процедура** (Insert/Procedure), раскроется диалоговое окно **Добавление Процедуры** (Add Procedure);



— в поле имени появившегося окна введем имя функции — НакСумма, установим переключатель *Функция*. Щелкнем на ОК. В окне редактирования кода появится заготовка функции вида

```
Public Function НакСумма()  
End Function
```

— внутри скобок заголовка функции введем описания ее параметров (исходных данных) в соответствии с приведенной таблицей, а за скобками укажем тип значения, возвращаемого функцией

```
Public Function НакСумма(ИсходнаяСумма As String, _  
Ставка As String, Срок As String) As Double
```

— под строкой заголовка функции поместим код инструкций, приведенный в листинге 9.

Листинг 9.

“Вычисление накопленного значения исходной суммы по формуле

“ $S=P(1+iT)$,

“ *S* — накопленное значение

“ *P* — величина ссуды

“ *i* — номинальная процентная ставка

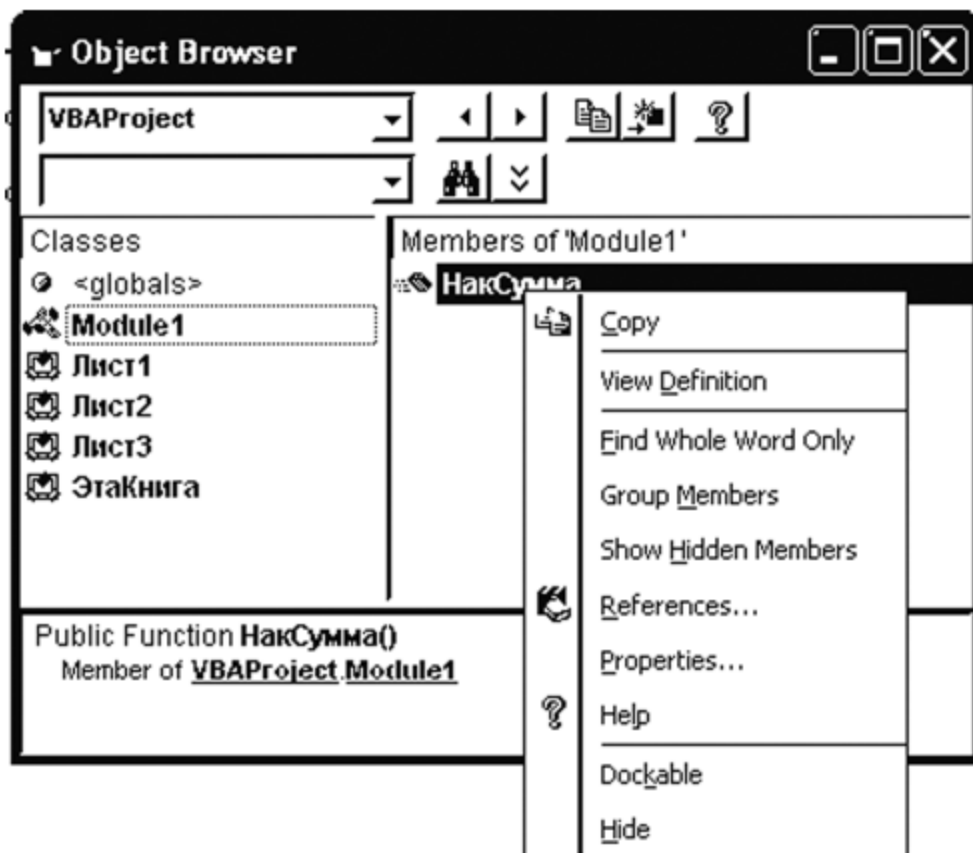
“ *T* — срок в месяцах

НакСумма = ИсходнаяСумма * (1 + Ставка * Срок / 12)

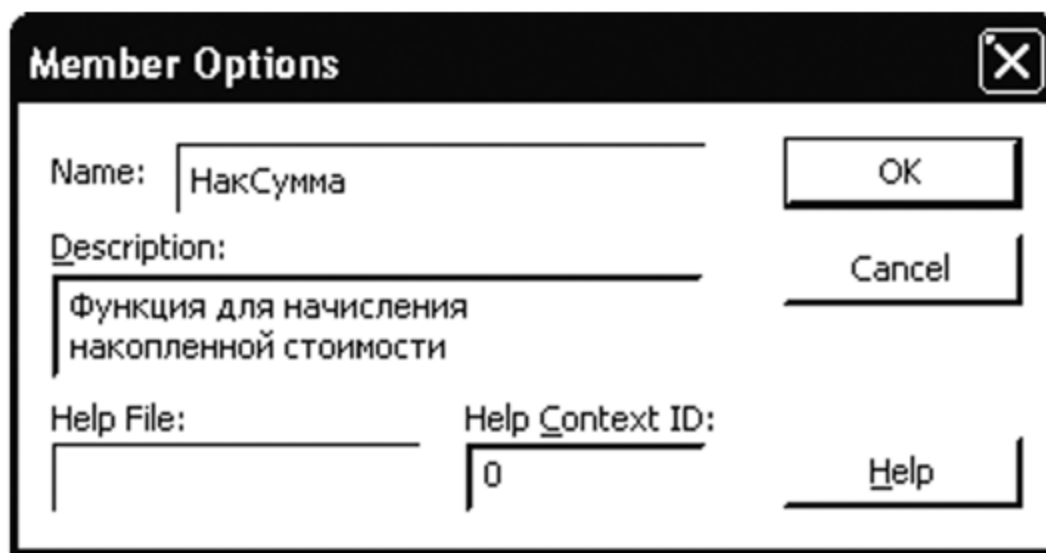
End If

Создадим описание к разработанной функции, для чего:

— выполним команду меню **Вид/Просмотр** объектов (View/Object Browser), откроется диалоговое окно **Object Browser**;



- в списке *Список проектов* и выберем *VBA Project* – в поле *Классы (Classes)* отобразятся элементы текущего проекта;
- выберем в окне **Классы** модуль, в котором записан код функции – в окне **Компоненты (Members)** отобразятся элементы, которые содержатся в этом модуле;
- выделим в окне **Компоненты (Members)** элемент с именем созданной функции и включим контекстное меню;
- выполним команду контекстного меню **Свойства (Properties)** – откроется окно **Параметры компонента (Member Options)**;



– в поле *Описание (Description)* окна **Параметры компонента (Member Options)** введем текст краткого описания функции, если необходимо, то указать файл справки и идентификатор.

Проверим работу функции на примере, вызывая ее с помощью Мастера функций Excel.

Правильный результат – 1100 (для $P = 1000$, $i = 10\%$ и $T = 12$ мес.).

Пример 5.2. Требуется доработать функцию пользователя, разработанную в предыдущем примере, таким образом, чтобы она проверяла корректность исходных данных. Данные корректны, если они являются числовыми.

Решение

Для этой цели используем оператор `If ... Then`.

Откроем окно редактирования кода для модуля, содержащего функцию `НакСумма`.

Дополним ее операторами проверки условий корректности. Поместим инструкции для вывода сообщений пользователю, если данные некорректны. Код функции после доработки приведен в листинге 10.

Листинг 10.

“Вычисление накопленного значения исходной суммы по формуле

“ $S = P(1 + iT)$,

“ S — накопленное значение

“ P — величина ссуды

“ i — номинальная процентная ставка

“ T — срок в месяцах

“ Проверка корректности данных

If (IsNumeric(ИсходнаяСумма) And IsNumeric(Ставка) And _

IsNumeric(Срок)) = False Then

MsgBox «Ошибка в исходных данных», vbInformation, _

«Вычисление накопленной суммы»

НакСумма = 0

Exit Function

Else

НакСумма = ИсходнаяСумма * (1 + Ставка * Срок / 12)

End If

Здесь применена встроенная функция IsNumeric, которая возвращает значение True, если ее аргумент является числом, и False, если аргумент не число.

Проверим работу функции для корректных и некорректных данных, вызывая ее с помощью мастера функций Excel.

Пример 5.3. Требуется разработать функцию пользователя для расчета накопленной суммы по методу простых процентов при неравномерных непериодических платежах.

Модель

Расчет накопленной стоимости при неравномерных непериодических платежах рассчитывается по формуле

$$S = \sum_{k=1}^{n-1} p_k i_k dt_k,$$

где S — накопленная сумма на счете;

n — количество платежей (периодов);

i_k — номинальная ставка за период;

dt_k — продолжительность периода, за который начисляются проценты;

p_k — сумма k -го платежа.

Исходными данными для решения задачи являются величины суммы платежей, даты их выплаты и номинальная процентная ставка. Вычисляемыми данными являются количество платежей и продолжительность периода.

Решение

Определим имена переменных и типы данных.

Переменная	Идентификатор	Тип данных	Примечание
Величина платежа	Платеж	Variant	Параметр функции, передающий значения диапазона ячеек
Дата платежа	Дата	Variant	Параметр функции, передающий значения диапазона ячеек
Номинальная ставка	Ставка	Single	Параметр функции
К	К	Integer	Вычисляемое значение
	i	Integer	Счетчик цикла
Dt	Дельта	Integer	Вычисляемое значение как разность между двумя смежными датами
Текущая сумма	Summa	Double	Вспомогательная переменная
Функция	НАКСУММА	Double	Имя функции (возвращаемое значение)

Замечание. Если параметр функции передает значения массива ячеек, то он должен иметь тип Variant.

Для создания функции выполним действия:

- создадим модуль;
- поместим в созданный модуль код подпрограммы — функции

(листинг 11).

Листинг 11.

```
Public Function НАКСУММА(Ставка As Single, _
Платеж As Variant, _
Дата As Variant) As Double
Dim K As Integer
Dim i As Integer
Dim Summa As Double "Текущая сумма вклада
Dim Дельта As Integer "Разность в днях между
'двумя платежами
Summa = Платеж(1)
K = Платеж.Count "общее количество платежей
For i = 2 To K
Дельта = Дата(i) - Дата(i - 1)
Summa = Summa + Summa * Ставка * Дельта / 365 + Платеж(i)
Next
НАКСУММА = Summa
End Function
```

Для проверки работоспособности функции решим задачу.

Пример 5.4. Клиент сделал вклад на текущий счет в банке в сумме 1000 руб. под 60% годовых. Через 3, 6 и 9 месяцев он вложил еще по 1000 руб. В конце учетного года клиент закрыл счет. Какую сумму он получил при закрытии счета?

Решение

На рабочем листе Excel подготовим таблицу с исходными данными и выполним расчет, применив созданную функцию.

Платеж	Дата платежа		
1000 руб.	01.01.2001		
1000 руб.	01.04.2001		
1000 руб.	01.07.2001		
1000 руб.	01.10.2001		Правильный результат
0	31.12.2001	Накопленная сумма	5 740,95 руб.

Пример 5.4. Требуется создать функцию для рабочего листа, выполняющую транспонирование исходной матрицы.

Решение

Суть транспонирования заключается в создании новой матрицы, которая отличается от исходной тем, что строки исходной матрицы в результирующей отображаются столбцами, а столбцы — строками.

Определим имена переменных и типы данных:

Переменная	Идентификатор	Тип данных	Примечание
Имя функции	МТРАНС	Variant	Возвращаемое значение
Матрица	Матрица	Variant	Параметр функции
Рабочий массив	<i>t</i>	Double	
Счетчики циклов	<i>n, m</i>	integer	

Код функции приведен в листинге 12.

Листинг 12.

Public Function МТРАНС(Матрица As Variant) As Variant

“ Транспонирует исходную матрицу

Dim n As Integer

Dim m As Integer

Dim t() As Double

‘ определение размерности исходной матрицы

n = Матрица.Rows.Count

m = Матрица.Columns.Count

‘ переопределение массива

```

ReDim t(1 To m, 1 To n)
For i = 1 To m
For j = 1 To n
'замена строк и столбцов
t(i, j) = Матрица(j, i)
Next
Next
МТРАНС = t
End Function

```

5.6. СОЗДАНИЕ ПРИЛОЖЕНИЙ

Visual Basic for Application предоставляет хорошие возможности для создания приложений пользователя, которые позволяют автоматизировать решение задач. Например, в главе 1 были рассмотрены технологии решения систем линейных уравнений методом обратной матрицы и методом наименьших квадратов, вычисления определенных интегралов. Решение этих задач связано с созданием относительно сложных формул. Если такие задачи решаются достаточно часто, то, конечно, следует автоматизировать эти процессы, создав соответствующие пользовательские приложения. Рассмотрим технологии создания подобных приложений на примерах. Изучив приведенные примеры, пользователь может самостоятельно создавать приложения для решения различных задач.

Пример 5.6. Требуется создать приложение для решения систем линейных уравнений методом обратной матрицы.

Разработка

Исходными данными для решения поставленной задачи являются матрица коэффициентов системы A и вектор свободных членов B .

Для ввода этих данных в программу вычислений применим диалоговое окно **Форма пользователя (User Form)**.

Создадим в редакторе Visual Basic форму и поместим в нее элементы управления, как показано на рисунке.



Элементы управления, размещенные в форме, имеют следующие характеристики:

Элемент	Назначение	Имя	Свойства	Обрабатываемое событие
UserForm	Диалоговое окно	LU	Caption – Метод обратной матрицы	Нет
RefEdit	Для ввода адреса матрицы A	A	Нет	Нет
RefEdit	Для ввода адреса матрицы B	B	Нет	Нет
RefEdit	Для ввода адреса матрицы X	X	Нет	Нет
CommandButton	Исполнительный	OK	Caption – Вычислить	Click Вызывает процедуру вычисления
CommandButton	Исполнительный	Can	Caption	Click

Для исполнительных элементов в процедуры обработки событий введем соответствующий код (листинг 13).

Листинг 13.

```
Private Sub OK_Click()
' Решение системы методом обратной матрицы
Range(X.Value).Select 'Выделяем диапазон для результата
' Помещаем в выделенную область рабочего листа формулу:
Selection.FormulaArray = «=MMULT(MINVERSE(« & A.Value & «),» &
B.Value & «)»
End Sub
Private Sub Can_Click()
' Завершаем работу приложения
End
End Sub
```

Для удобства запуска приложения создадим макрос:

```
Sub ОбратнаяМатрица()
" Макрос записан 24.03.2008 (Мельников)
'Включаем форму пользователя
LU.Show
End Sub
```

Проверим функционирование приложения на задаче.

Пример 5.7. Требуется создать приложение для решения систем линейных уравнений методом наименьших квадратов.

Разработка

Создадим диалоговое окно, как изображено на рисунке.



Элементы управления, размещенные в форме, имеют следующие характеристики:

Элемент	Назначение	Имя	Свойства	Обрабатываемое событие
UserForm	Диалоговое окно	Squar	Caption – Решение линейных уравнений наименьшими квадратами	Нет
RefEdit	Для ввода адреса матрицы А	matr	Нет	Нет
RefEdit	Для ввода адреса матрицы В	vect	Нет	Нет
RefEdit	Для ввода адреса матрицы X	solv	Нет	Нет
CommandButton	Исполнительный ОК	OK	Caption – ОК	Click Вызывает процедуру вычисления
CommandButton	Исполнительный Can	Can	Caption – Cancel	Click

В процедуры обработки событий кнопок введем код (листинг 14).

Листинг 14.

```
Private Sub OK_Click()
```

```
‘Выделяем диапазон ячеек рабочего листа для размещения формулы  
Range(solv.Value).Select
```

```
‘ Формируем и формулу и помещаем ее в выделенный диапазон
```

```
Selection.FormulaArray = _
```

```
«=MMULT(MINVERSE(MMULT(TRANSPOSE(« & matr.Value & «),» &  
matr.Value & _
```

```

«)),MMULT(TRANSPOSE(« & matr.Value & «),» & vect.Value & «))»
End Sub
Private Sub CommandButton2_Click()
' Завершаем работу приложения
End
End Sub

```

Создадим макрос для запуска приложения:

```

Sub leastSquar ()
'Макрос выполняет запуск приложения
" Решение систем линейных уравнений методом наименьших квадратов
" leastSquar
" Макрос записан 24.03.2008 (Мельников)
Squar.Show
End Sub

```

Проверим работу приложения на задаче.

Пример 5.8. Требуется разработать универсальную программу для решения систем линейных уравнений. Программа самостоятельно должна выбирать способ решения в зависимости от того, является матрица A квадратной или нет.

Разработка

Создадим диалоговое окно.



Элементы управления, размещенные в форме, имеют следующие характеристики:

Элемент	Назначение	Имя	Свойства	Обрабатываемое событие
UserForm	Диалоговое окно	LU	Caption – Системы линейных уравнений	Нет

Элемент	Назначение	Имя	Свойства	Обрабатываемое событие
RefEdit	Для ввода адреса матрицы А	А	Нет	Нет
RefEdit	Для ввода адреса матрицы В	В	Нет	Нет
RefEdit	Для ввода адреса матрицы Х	Х	Нет	Нет
CommandButton	Исполнительный	OK	Caption – Вычислить	Click Вызывает процедуру вычисления
CommandButton	Исполнительный	Can	Caption – Cancel	Click

В процедуры обработки событий кнопок введем код инструкций (листинг 15)

Листинг 15.

```
Private Sub OK_Click()
```

```
Dim n As Integer “ количество строк в матрице А
```

```
Dim m As Integer “ количество столбцов в матрице А
```

```
‘ Вычисляем количество строк матрицы А
```

```
n = Range(A.Value).Rows.Count
```

```
‘ Вычисляем количество столбцов матрицы А
```

```
m = Range(A.Value).Columns.Count
```

```
If m = n Then “ Матрица А квадратная ?
```

```
‘ Вычисление методом обратной матрицы
```

```
Range(X.Value).Select
```

```
Selection.FormulaArray = «=MMULT(MINVERSE(« & A.Value & «),» & B.Value & «)»
```

```
Else
```

```
‘ Вычисление методом наименьших квадратов
```

```
Range(X.Value).Select
```

```
Selection.FormulaArray = _
```

```
«=MMULT(MINVERSE(MMULT(TRANSPOSE(« & A.Value & «),» & A.Value & _
```

```
«)),MMULT(TRANSPOSE(« & A.Value & «),» & B.Value & «))»
```

```
End If
```

```
End Sub
```

```
Private Sub Can_Click()
```

```
‘ Завершаем работу приложения
```

```
End
```

```
End Sub
```

Для удобства запуска приложения создадим макрос, как в предыдущих примерах.

Проверим корректность функционирования приложения на задачах.

Пример 5.9. Требуется разработать приложение для вычисления производной функции в заданной точке. Функция задается в виде формулы, введенной в ячейку рабочего листа со ссылкой на ячейку, содержащей аргумент.

Разработка

Создадим форму диалогового окна, как на рисунке. В форме размещены три элемента RefEdit и две кнопки.



Элементы управления, размещенные в форме, имеют следующие характеристики:

Элемент	Назначение	Имя	Свойства	Обрабатываемое событие
UserForm	Диалоговое окно	proizv	Caption – Системы линейных уравнений	Нет
RefEdit	Для ввода адреса ячейки, содержащей формулу	fun	Нет	Нет
RefEdit	Для ввода адреса аргумента	val	Нет	Нет
RefEdit	Для ввода адреса результата	solv	Нет	Нет
CommandButton	Исполнительный	OK	Caption – OK	Click Вызывает процедуру вычисления
CommandButton	Исполнительный	Can	Caption – Cancel	Click

В процедуры обработки событий исполнительных кнопок введем код (листинг 16).

Листинг 16.

```
Private Sub OK_Click()  
    ' Процедура вычисления  
    Dim delta As Double  
    Dim f1 As Double  
    delta = 0.00000000001 ' Константа, задающая приращение аргумента  
    ' Выделяем ячейку для размещения результата вычислений  
    Range(solv.Value).Select  
    ' Помещаем в ячейку формулу для вычисления значения функции в точке  $X_k$   
    ActiveCell.Formula = «=» & fun.Value  
    ' Запоминаем вычисленное значение функции в точке  $x_k$   
    f1 = ActiveCell.Value  
    ' Помещаем в ячейку формулу для вычисления значения функции в точке  $X_{k+1}$   
    Range(val.Value).Value = Range(val.Value).Value + delta  
    ' Помещаем в ячейку вычисленное значение производной  
    ActiveCell.Value = (ActiveCell.Value - f1) / delta  
End Sub  
Private Sub CommandButton2_Click()  
    ' Процедура завершения  
End  
End Sub
```

Создадим макрос для запуска приложения:

```
Sub Произв()  
    " Вычисляет производную функции в точке  
    " Макрос записан 24.03.2008 (Мельников)  
    proizv.Show  
End Sub
```

Проверим функционирование приложения и корректность вычислений на задаче.

Пример 5.10. Требуется разработать приложение для вычисления определенного интеграла.

Можно было заметить, что технология точного вычисления определенного интеграла с использованием циклических ссылок MS Excel достаточно сложна. Поэтому целесообразно создать приложение для автоматизации решения этой задачи.

Разработка

Исходными данными для вычисления определенного интеграла являются:

- формула подынтегральной функции;
- адрес ячейки, на которую ссылается формула (аргумент функции);
- значения нижнего и верхнего пределов интегрирования.

Создадим форму пользователя, выполняющую роль диалогового окна, подобную приведенной на рисунке.



Элементы управления, размещенные в форме, имеют следующие характеристики:

Элемент	Назначение	Имя	Свойства	Обрабатываемое событие
UserForm	Диалоговое окно	integral	Caption – Вычисление определенного интеграла	Нет
RefEdit	Для ввода адреса ячейки, содержащей формулу	fun	Нет	Нет
RefEdit	Для ввода адреса аргумента	arg	Нет	Нет
RefEdit	Для ввода адреса результата	rez	Нет	Нет
TextBox	Поле для ввода величины нижнего предела интегрирования	NP	Нет	Нет
TextBox	Поле для ввода величины верхнего предела интегрирования	VP	Нет	Нет

Элемент	Назначение	Имя	Свойства	Обрабатываемое событие
CommandButton	Исполнительный	OK	Caption – OK	Click Вызывает процедуру вычисления
CommandButton	Исполнительный	Can	Caption – Cancel	Click

Для вычисления интеграла используем формулу трапеций. В процедуры обработки событий введем соответствующий код инструкций (листинг 17).

Листинг 17.

```
Private Sub OK_Click()
' Процедура вычисления
Dim delta As Double ' Переменная, хранящая шаг интегрирования
Dim f1 As Double ' Переменная для хранения значения функции на предыдущем шаге
Dim s As Double ' Переменная для накопления сумм площадей элементарных фигур
Dim c As Integer ' Переменная для хранения количества шагов
delta = 0.001 ' Константа величины приращения
' Вычисляем количество шагов интегрирования
c = CInt((CDBl(VP.Text) – CDBl(NP.Text)) / delta) – 1
' В ячейку, содержащую аргумент функции, вводим величину нижнего предела
Range(arg.Value).Value = NP.Value
' Выделяем ячейку с формулой функции
Range(fun.Value).Select
s = 0
« Выполняем цикл
For i = 1 To c
' Запоминаем значение функции
f1 = ActiveCell.Value
' Вычисляем значение функции на следующем шаге, вычисляем
«площадь элементарной фигуры и накапливаем сумму
Range(arg.Value).Value = Range(arg.Value).Value + delta
s = s + f1 * delta + (Range(fun.Value).Value – f1) * delta / 2
Next
' В ячейку рабочего листа выводим результат
```



```

Range(rez.Value).Value = s
End
End Sub
Private Sub CommandButton2_Click()
' Процедура завершения
End
End Sub
Sub Интеграл()
' Макрос для запуска приложения
" Интеграл макрос
" Макрос записан 24.03.2008 (Мельников)
' Вызываем диалоговое окно
Integral.Show
End Sub

```

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какова технология создания макросов с помощью макроре-кордера?
2. Какие элементы может включать проект VBA?
3. Назовите базовые типы переменных Visual Basic и их назна-чение.
4. Каким образом объявляются переменные в программе?
5. Что такое цикл с известным числом повторений, какова его синтаксическая конструкция?
6. Какие операторы выбора решения вы знаете?

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задание 5.1. Разработайте функцию пользователя для вычисле-ния накопленной суммы при условии непрерывного начисления про-центов вкладу. Платежи неравномерные, непериодические. Функция должна учитывать европейский и американский стандарт исчисления количества дней в финансовом году. С помощью разработанной функ-ции вычислите сумму на счете клиента, если годовая ставка равна 12%, а платежи выполнялись в соответствии с приведенной таблицей.

Дата платежа	Сумма платежа, руб.
01.06.2005	10 000,00
01.09.2005	1 500,00
31.12.2005	700,00
05.03.2006	1 000,00
09.05.2006	750,00
31.05.2006	

Задание 5.2. Создайте приложение для вычисления матрицы коэффициентов прямых затрат (Модель Леонтьева), выполняющее анализ продуктивности по двум критериям. Предусмотрите вывод информации о продуктивности (непродуктивности).

ОТВЕТЫ К ЗАДАНИЯМ

Глава 1

1.1. $\begin{pmatrix} 8 & 12 \\ 20 & 5 \end{pmatrix}$.

1.2. 1) $\begin{pmatrix} 25 \\ 52 \\ 37 \\ 44 \end{pmatrix}$; 2) $\begin{pmatrix} 20 \\ 19 \\ 37 \\ 49 \end{pmatrix}$.

1.3. 1) $\begin{pmatrix} 60 \\ 150 \\ 90 \\ 120 \end{pmatrix}$; 2) $\begin{pmatrix} 120 \\ 180 \\ 360 \\ 540 \end{pmatrix}$.

1.4. 1) 570; 2) 60.

1.5. 1а) 32; 1б) $\begin{pmatrix} 2 & 3 & 5 \\ 4 & 1 & 3 \\ 7 & 2 & 0 \end{pmatrix}$; 2) $\begin{pmatrix} 0,396552 & 0,275862 & 0,017241 \\ 0,034483 & 0,241379 & -0,17241 \\ 0,224138 & 0,068966 & -0,12069 \end{pmatrix}$.

1.6. 1) $\begin{pmatrix} 1 \\ -1,4557 \\ 3,113924 \\ -2,02532 \end{pmatrix}$; 2) $\begin{pmatrix} -2,45113 \\ -0,69173 \\ 0,954887 \\ 2,067669 \end{pmatrix}$; 3) $\begin{pmatrix} 1,5 \\ 0 \\ -0,5 \end{pmatrix}$, $\begin{pmatrix} 2,4 \\ 1,9 \\ 1,9 \end{pmatrix}$.

1.7. а) $x = 0, y = 2$; б) $x = -1, y = 0, z = 1$;

1.8. Годовая производительность каждого предприятия:

$$A_{\text{год}} = \begin{pmatrix} 800 & 750 & 510 & 720 & 980 \\ 0 & 300 & 680 & 360 & 0 \\ 1600 & 2250 & 0 & 480 & 840 \\ 600 & 1500 & 1190 & 600 & 560 \end{pmatrix}.$$

Матрица затрат сырья на единицу изделия.

Вид изделия 1 2 3 4:

$$B = \begin{pmatrix} 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 8 \\ 4 & 6 & 5 & 6 \end{pmatrix}.$$

Дневной расход сырья на предприятиях:

$$BA = \begin{pmatrix} 55 & 126 & 53 & 62 & 58 \\ 68 & 165 & 85 & 89 & 77 \\ 74 & 167 & 78 & 92 & 82 \end{pmatrix}.$$

Годовая потребность каждого предприятия в каждом виде сырья:

$$BA_{\text{год}} = \begin{pmatrix} 11000 & 18900 & 9019 & 7440 & 8120 \\ 13600 & 24750 & 14450 & 10680 & 10780 \\ 14800 & 25050 & 13260 & 11040 & 11480 \end{pmatrix}.$$

Стоимость общего годового запаса сырья для каждого предприятия:

$$P = (2008000, 3496500, 1878500, 1494000, 1552600).$$

1.9. 1) $x = \begin{pmatrix} 152,6 \\ 135,8 \\ 92,5 \end{pmatrix}$; 2) $y = (135, 34, 35, 40)$.

1.10. $(1, 8, 27, 64, \dots, 1728)$, $(1, 0,5, 0,3333, \dots, 0,0833)$, $(0,5, 0,666667, \dots, 0,923077)$.

1.11. 1) 2,718282; 2) 0,5; 3) 4,99999E-06; 4) 2036,32.

1.12. 1) 0,875; 2) 7,016334393.

1.13. 1) 3007,26 руб.; 2) 107 738,32 руб.;

1.15. 1) 1; 2) 2,718282053.

1.16. 1) 2; 2) -3,14157, -1,57079, 1,571105, 3,141505; 3) 0,376002; 4) 0,322469; 5) 0,006784; 6) 3,837965; 7) 2,066903; 8) -0,7854, 2,356194; 9) 0,549306.

1.17. 1) $x_1 = 2,32$, $x_2 = -2,04$; 2) 2,387113242.

1.18. 1) $x = 1,2$; $y = 1,55$; 2) $x = 1,1$; $y = 1,67$.

1.19. 2,796226.

1.20. 0,87077.

1.21. Эластичность спроса -0,10747; эластичность предложения 0,525712.

1.22. 1) Приближенное решение 0,999196; точное решение 0,999999687; 2) 2994,006988; 3) 497160,6; 4) 168,7912544; 5) 892 (млрд руб.).

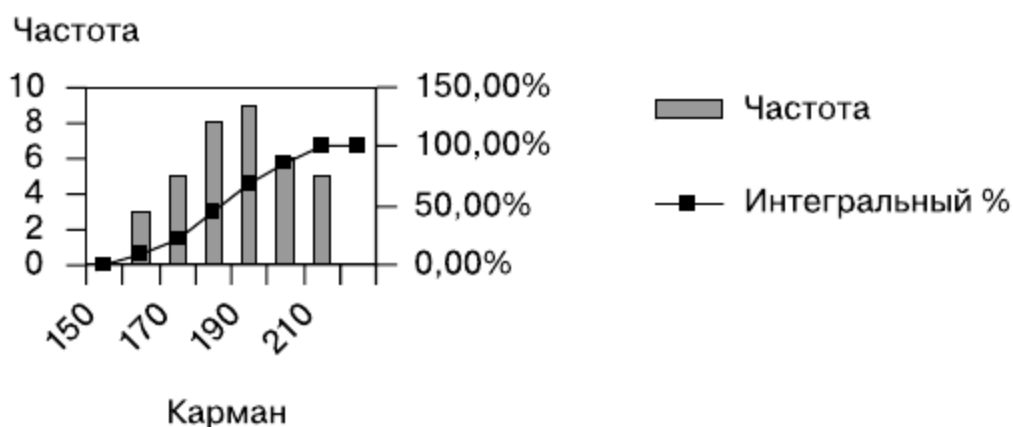
- 1.23. 1) $x_1 = 1,75; x_2 = 0; L(x)_{\max} = 5,25$; 2) $x_1 = 4; x_2 = 5; L(x)_{\max} = 23$;
 3) $A=2,25; B=5,5$; наибольший доход — 1775.

Глава 2

- 2.2. 1) Математическое ожидание — 2,975, стандартное отклонение — 2,54937; 2) 1; 3) 5,389136.
 2.3. 1) 281,25 тыс. ден. ед.; 2) 3,9 15,21.
 2.4. 0,0378.
 2.5. 1) 25,31174, 23,28826; 2) 0,908789; 3) 12; 4) 42; 5) 24,17%.
 2.6.

Карман	Частота	Интегральный процент
150	0	0,00
160	3	8,33
170	5	22,22
180	8	44,44
190	9	69,44
200	6	86,11
210	5	100,00
Еще	0	100,00

Гистограмма



- 2.7. Выходит за границы.
 2.8. (0, 2, 5, 2).
 2.9.

	Уровень инфляции (%)	Ставка рефинансирования	Курс (руб./дол.)
Уровень инфляции (%)	1		
Ставка рефинансирования (%)	0,992039776	1	
Курс (руб./дол.)	-0,913153161	-0,915305695	1

Глава 3

- 3.1. 212000.
- 3.2. 1) 229; 2) 212000.
- 3.3. 73 712 844,81 руб.

Глава 5

- 5.1. 15 409,99; 15 431,14.

ГЛОССАРИЙ

Диапазон ячеек рабочего листа — область, включающая смежные ячейки рабочего листа, которая имеет адрес или имя.

Генератор случайных чисел — инструмент, входящий в пакет анализа табличного процессора, предназначенный для генерации массивов чисел, распределенных по различным законам.

Компьютерная модель — модель, которая реализуется и исследуется на компьютере, с использованием соответствующих инструментов и технологий.

Консолидация — объединение данных из одной или нескольких таблиц с выполнением необходимых расчетов.

Макрос — программа на языке программирования, оформленная как процедура SUB, выполняющая заданную последовательность операций на рабочем листе.

Массив — набор данных одного типа.

Математическая модель — модель объекта или процесса, заданная аналитическими выражениями (математическими формулами)

Модель — прототип реального объекта либо процесса, который адекватно отражает только те свойства реального процесса или объекта, которые существенны для исследования или для решения задачи.

Поиск решения — инструмент, включенный в надстройки табличного процессора, который вычисляет заданное значение функции путем подбора значений ее аргументов.

Целевая функция — функция, которая оптимизируется путем подбора ее аргументов.

Linux — открытая операционная система, альтернативная ОС Microsoft Windows.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

- Васильков Ю.В., Василькова Н.Н.* Компьютерные технологии вычислений в математическом моделировании : учеб. пособие. М. : Финансы и статистика, 2002.
- Гельман В.Я.* Решение математических задач средствами Excel : практикум. СПб. : Питер, 2003.
- Ильченко А.Н.* Экономико-математические методы : учеб. пособие. М. : Финансы и статистика, 2006.
- Красс М.С., Чупрынов Б.П.* Математика для экономистов. СПб. : Питер, 2004 (Серия «Учебное пособие»).
- Кутуков В.Б.* Основы финансовой и страховой математики: Методы расчета кредитных, пенсионных, инвестиционных и страховых схем. М. : Дело, 1998.
- Практикум по эконометрике : учеб. пособие / И.И. Елисеева, С.В. Курешева и др. ; под ред. И.И. Елисеевой. М. : Финансы и статистика, 2006.
- Программирование в пакетах MS Office: учеб. пособие / С.В. Назаров, П.П. Мельников, Л.П. Смольников и др. ; под ред. С.В. Назарова. М. : Финансы и статистика, 2007.