

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Невинномысский технологический институт (филиал)

Методические указания к выполнению лабораторных работ
по дисциплине
«Информационные технологии в профессиональной
деятельности»
для студентов направления

Невинномысск, 2019

Настоящие методические указания предназначены для студентов направления 15.03.04 Автоматизация технологических процессов и производств. Они разработаны в соответствии с федеральным государственным образовательным стандартом.

В методических указаниях определены требования к содержанию лабораторной работы, даны варианты заданий.

Составитель
Отв. редактор

канд. экон. наук, доцент А.И Колдаев
канд. техн. наук А.А. Евдокимов

СОДЕРЖАНИЕ

Введение	
№ 1. Кодирование информации методом Хаффмана	5
№ 2. Текстовый процессор Microsoft Word	9
№ 3. Текстовый процессор Microsoft Word	21
№ 4. Издательская система Adobe Acrobat Professional	32
№ 5. Табличный процессор Microsoft Excel	34
№ 6. Табличный процессор Microsoft Excel	54
№ 7. СУБД Microsoft Access	67
№ 8. СУБД Microsoft Access	70
№ 9. Основы объектно-ориентированного программирования на языке DELPHI	72
№ 10. Моделирование работы ИС	76
№ 11. Обработка массивов данных	80
№ 12. Поиск в массиве заданного элемента	83
№ 13. Сортировка данных в массиве	86
№ 14. Работа с файлами	89
№ 15. Разработка БД	95
№ 16. Программирование БД	97
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	114

ВВЕДЕНИЕ

Использование лабораторного комплекса по дисциплине Информационные системы в профессиональной деятельности позволит активно воздействовать на образовательный процесс, что обеспечит его соответствие современным требованиям. Очевидно, что это решит далеко не все проблемы, связанные с качеством подготовки бакалавров. Однако функционирование данного комплекса позволит перейти от традиционных образовательных форм к новым технологиям.

Методические указания по циклу лабораторных работ занятий по дисциплине Информационные системы в профессиональной деятельности являются практической частью комплекса, выполняющие основную задачу – практическое обучение. Предлагаемые лабораторные работы способствуют формированию практических навыков и умений в области информационных технологий. Студенты научатся применять информационные технологии при решении функциональных задач в различных предметных областях, а также при разработке и проектировании информационных систем. Получат представление об областях применения информационных технологий и их перспективах.

Лабораторная работа № 1 КОДИРОВАНИЕ ИНФОРМАЦИИ МЕТОДОМ ХАФФМАНА

Цель работы: научиться кодировать информацию методом Хаффмана.

Теоретические сведения

Построение кодов Хаффмана рассмотрим на следующем примере: пусть имеется первичный алфавит A , состоящий из шести знаков $a_1 \dots a_6$ с вероятностями появления в сообщении, соответственно, $0,3; 0,2; 0,2; 0,15; 0,1; 0,05$. Создадим новый вспомогательный алфавит A_1 , объединив два знака с наименьшими вероятностями (a_5 и a_6) и заменив их одним знаком (например, $a^{(1)}$); вероятность нового знака будет равна сумме вероятностей тех, что в него вошли, т.е. $0,15$; остальные знаки исходного алфавита включим в новый без изменений; общее число знаков в новом алфавите, очевидно, будет на 1 меньше, чем в исходном. Аналогичным образом продолжим создавать новые алфавиты, пока в последнем не останется два знака; ясно, что число таких шагов будет равно $N - 2$, где N – число знаков исходного алфавита (в нашем случае $N = 6$, следовательно, необходимо построить 4 вспомогательных алфавита). В промежуточных алфавитах каждый раз будем переупорядочивать знаки по убыванию вероятностей. Вся процедура построения представим в виде таблицы:

Шаг	Вероятности				
	Исходный алфавит	Промежуточные алфавиты			
		$A^{(1)}$	$A^{(2)}$	$A^{(3)}$	$A^{(4)}$
1	0,3	→ 0,3	→ 0,3	→ 0,4	→ 0,6
2	0,2	→ 0,2	→ 0,3	→ 0,3	→ 0,4
3	0,2	→ 0,2	→ 0,2	→ 0,3	
4	0,15	→ 0,15	→ 0,2		
5	0,1	→ 0,15			
6	0,05				

Теперь в обратном направлении поведем процедуру кодирования. Двум знакам последнего алфавита присвоим коды 0 и 1 (какому – роли не играет; условимся, что верхний знак будет иметь код 0 , а нижний – 1). В нашем примере знак $a_1^{(4)}$ алфавита $A^{(4)}$, имеющий вероятность $0,6$, получит код 0 , а $a_2^{(4)}$ с вероятностью $0,4$ – код 1 . В алфавите $A^{(3)}$ знак $a_1^{(3)}$ с вероятностью $0,4$ сохранит свой код (1); коды знаков $a_2^{(3)}$ и $a_3^{(3)}$, объединенных знаком $a_1^{(4)}$ с вероятностью $0,6$, будут уже двузначным: их первой цифрой станет код связанного с ними знака (т.е. 0), а вторая цифра – как условились – у верхнего 0 , у нижнего – 1 ; таким образом, $a_2^{(3)}$ будет

иметь код 00 , а $a_3^{(3)}$ – код 01 . Полностью процедура кодирования представлена в следующей таблице:

№ знака	Вероятности									
	Исходный алфавит		Промежуточные алфавиты							
			A ⁽¹⁾		A ⁽²⁾		A ⁽³⁾		A ⁽⁴⁾	
1	0,3	00	0,3	00	0,3	00	0,4	1	0,6	0
2	0,2	10	0,2	10	0,3	01	0,3	00	0,4	1
3	0,2	11	0,2	11	0,2	10	0,3	01		
4	0,15	010	0,15	010	0,2	11				
5	0,1	0110	0,15	011						
6	0,05	0111								

Средняя длина кода при этом оказывается:

$$K^{(2)} = 0,3 \cdot 2 + 0,2 \cdot 2 + 0,2 \cdot 2 + 0,15 \cdot 3 + 0,1 \cdot 4 + 0,05 \cdot 4 = 2,45$$

Код Хаффмана важен в теоретическом отношении, поскольку можно доказать, что он является *самым экономичным* из всех возможных, т.е. *ни для какого метода алфавитного кодирования длина кода не может оказаться меньше, чем код Хаффмана*

Задание: закодировать методом Хаффмана алфавит.

Вариант № 1

знак	q	y	g	d	k	h	j	l
вероятность появления в тексте	0,03	0,05	0,005	0,002	0,0069	0,5	0,005	0,4011

Вариант № 2

знак	a	d	r	t	b	y	k	l
вероятность появления в тексте	0,2	0,036	0,01	0,23	0,004	0,5	0,001	0,019

Вариант № 3

знак	1	2	3	4	5	6	7	8
вероятность появления в тексте	0,36	0,2	0,02	0,05	0,003	0,25	0,06	0,057

Вариант № 4

знак	b	g	y	t	r	d	w	t
вероятность появления в тексте	0,25	0,03	0,23	0,003	0,02	0,3	0,036	0,131

Вариант № 5

знак	0	1	2	3	4	5	6	7
вероятность появления в тексте	0,05	0,36	0,02	0,001	0,003	0,069	0,23	0,267

Вариант № 6

знак	r	t	b	f	s	e	t	y
вероятность появления в тексте	0,02	0,036	0,05	0,26	0,04	0,06	0,09	0,444

Вариант № 7

знак	e	t	y	h	d	w	o	g
вероятность появления в тексте	0,12	0,05	0,36	0,23	0,06	0,059	0,02	0,101

Вариант № 8

знак	x	f	r	y	j	m	i	o
вероятность появления в тексте	0,04	0,056	0,56	0,23	0,005	0,006	0,002	0,101

Вариант № 9

знак	q	f	i	d	k	h	j	l
вероятность появления в тексте	0,02	0,096	0,02	0,04	0,36	0,21	0,15	0,104

Вариант № 10

знак	e	o	p	d	k	h	j	l
вероятность появления в тексте	0,5	0,05	0,09	0,2	0,09	0,001	0,015	0,054

Лабораторная работа № 2 ТЕКСТОВЫЙ ПРОЦЕССОР Microsoft Word

Цель работы: научиться создавать титульные листы документов, набирать и форматировать текст, научиться формировать списки разного вида. Изучить принцип работы Microsoft Equation. Научиться создавать с его помощью формулы.

Теоретические сведения

Создание и редактирование документов с помощью компьютера, т. е. овладение офисными информационными технологиями становится в информационном обществе социально необходимым умением. Современные информационные технологии позволяют включать в состав документа любые объекты (графику, звук, анимацию, видео). Текстовый процессор является инструментарием информационной технологии автоматизированного офиса.

Текстовый процессор – программные средства, предназначенные для создания, редактирования и форматирования простых и комплексных текстовых документов.

Создаваемые текстовые документы разделяют на две группы – *простые* и *комплексные*. Первые представляют собой форматированный текст, а вторые содержат, кроме текста, объекты другой природы (чертежи, рисунки, формулы, таблицы, объекты мультимедиа и др.).

Форматирование текста осуществляется средствами меню **Формат** или панели Форматирование.

Изменение шрифтового оформления

Для изменения шрифта выделенного текста или чисел нажмите кнопку **Шрифт**. Из списка **Шрифт** выберите имя нужного шрифта.

Изменение формата символов

С помощью панели форматирования можно изменить шрифт выделенного фрагмента текста, а также размер символов. Если какой-либо шрифт определенного размера используется чаще других, сделайте его используемым по умолчанию. Для этого нажмите кнопку **По умолчанию** в окне диалога **Шрифт** из меню **Формат**.

С помощью кнопок на панели форматирования легко сделать выделенный текст полужирным, курсивным или подчеркнутым. Эти типы начертаний могут быть применены как по отдельности, так и в любом сочетании.

Чтобы получить возможность изменить другие параметры символов, выполните команду **Шрифт** из меню **Формат**. С помощью окна диалога

Шрифт можно сделать выделенный текст скрытым, зачеркнутым или изменить его цвет.

Иногда в начале абзаца вставляют буквицу – прописную букву большего размера. Для создания буквицы выделите букву, выполните команду **Буквица** из меню **Формат**, а затем выберите одно из положений буквицы, например **В тексте**. Чтобы удалить буквицу, выберите **Нет**.

Приложение WordArt представляет средства для художественного оформления текста. Чтобы воспользоваться приложениями WordArt, выполните команду **Объект** из меню **Вставка**, а затем на вкладке **Создать новый** выберите Microsoft WordArt.

Изменение размера символов

Для изменения размера шрифта выделенного текста или чисел нажмите кнопку **Размер шрифта**. Из списка **Размер шрифта** выберите размер. Доступные размеры зависят от принтера и выбранного шрифта.

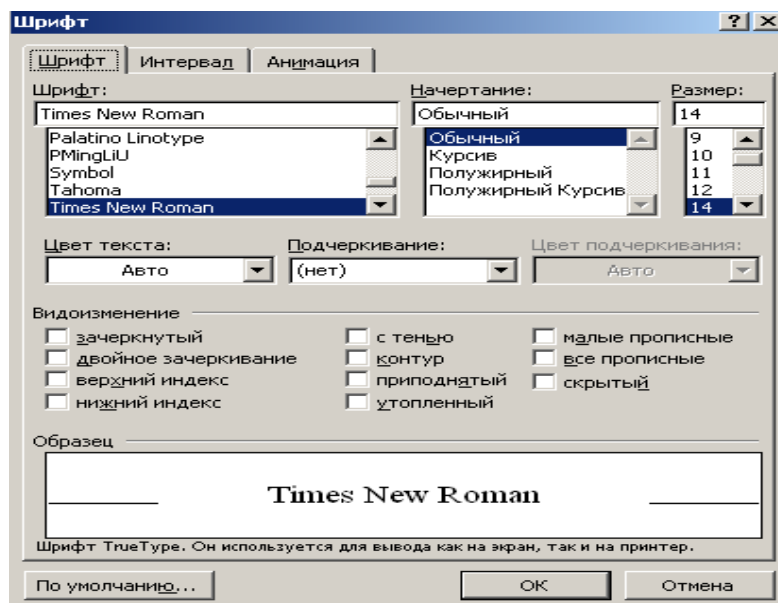


Рис. 2.1. Окно **Шрифт**

Верхние индексы

Чтобы оформить выделенный текст как верхний индекс, установите флажок **Верхний индекс** команды **Шрифт** из меню **Формат**. Размер и положение индекса определяются используемым шрифтом.

Нижние индексы

Чтобы оформить выделенный текст как нижний индекс, установите флажок **Нижний индекс** команды **Шрифт** из меню **Формат**. Размер и положение индекса определяются используемым шрифтом.

Зачеркивание символов

Чтобы зачеркнуть выделенный текст, установите флажок **Зачеркнутый** команды **Шрифт** из меню **Формат**.

Изменение межсимвольного интервала

Чтобы изменить расстояние между символами, укажите нужный параметр в поле **Интервал** вкладки **Интервал** команды **Шрифт** из меню **Формат**.

Преобразование букв в прописные

Чтобы выделенный текст отображался прописными буквами, установите флажок **Все прописные** команды **Шрифт** из меню **Формат**.

Преобразование букв в малые прописные

Чтобы выделенный текст отображался малыми прописными буквами, установите флажок **Малые прописные** команды **Шрифт** из меню **Формат**.

Изменение положения текста относительно опорной линии

Чтобы поднять или опустить выбранный текст относительно опорной линии, выберите значения **Вверх** или **Вниз** в поле **Смещение** на вкладке **Интервал** команды **Шрифт** из меню **Формат**.

Изменение формата абзаца

Чтобы изменить выравнивание текста внутри выделенного абзаца, нажмите соответствующую кнопку на панели форматирования. Существуют четыре способа выравнивания абзацев:

по левому краю – левый край ровный, а правый как получится;

по центру – весь текст выровнен по центру;

по правому краю – правый край ровный, а левый как получится;

по ширине – оба края ровные. Выравнивание по ширине в данной программе происходит за счет растягивания пробелов между символами.

Выравнивание выделенных абзацев (если абзац только один, достаточно установить в любое место этого абзаца курсор) осуществляется: при помощи меню **Формат** ⇒ **Абзац**, если на вкладке **Отступы** и интервалы выбрать нужную строку раскрывающегося списка **Выравнивание** или нажатием соответствующей кнопки панели инструментов.

Схемы на кнопках являются подсказками способов выравнивания абзацев.

Абзацные отступы и интервалы

Существует три вида абзацных отступов:

1. Отступ первой строки абзаца задается с помощью команды **Абзац** меню **Формат**.

2. **Отрицательные** отступы применяются для вынесения текста на левое или правое поле. Этот тип отступа задается путем ввода отрицательного числа в окне диалога **Абзац**.

3. Выступы первой строки абзаца применяются для оформления элементов библиографических указателей, словарей, а также маркированных и нумерованных списков.

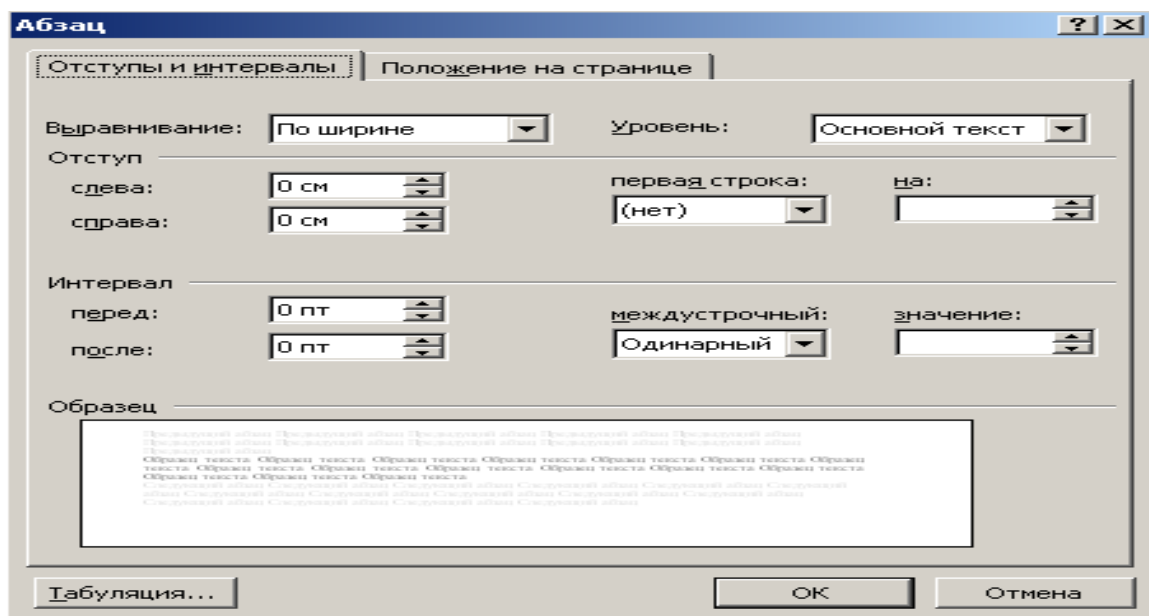


Рис. 2.2. Окно **Форматирование абзаца**

Установка отступов абзацев

Выберите абзацы, которые хотите снабдить отступом или отделить от правого или левого поля.

Чтобы сдвинуть абзац на одну позицию табуляции, нажмите кнопку **Увеличить отступ** на панели форматирования. Чтобы вернуть абзац к предыдущей позиции табуляции, нажмите кнопку **Уменьшить отступ** на панели форматирования.

Точные размеры отступа абзаца можно задать с помощью команды **Абзац** меню **Формат** и на вкладке **Отступы и интервалы** в поле ввода задать размер отступа.

Эту же операцию можно выполнить, передвигая мышью по горизонтальной линейке (треугольники и прямоугольник). Верхний треугольник соответствует положению начала первой строки абзаца, нижний – величине отступа слева. Если схватить и переместиться вместе, то будет выполняться отступ с учетом первой строки. В правой части линейки имеется только один треугольник, соответствующий отступу справа.



Создание абзацного выступа

- Выделите абзацы, которые требуется снабдить выступами. Выполните команду **Абзац** из меню **Формат**, а затем выберите вкладку **Отступы и интервалы**. В списке **Первая строка** выберите значение **Выступ**

Табуляция (установка красной строки)

- Выделите абзацы, в которых следует установить позиции табуляции.
- Нажмите кнопку слева от координатной линейки до тех пор, пока значок типа табуляции на ней не примет требуемый вид.
- Укажите на нижней границе координатной линейки место, в котором хотите задать позицию табуляции.
- Для точного задания позиций табуляции используйте команду **Табуляция** меню **Формат**.

Таблица 2.1

Междустрочные интервалы

Параметр	Описание
Одинарный интервал	Интервал, чуть-чуть превышающий высоту символов самого большого шрифта строки. Величина этого превышения зависит от используемого шрифта
Полуторный интервал	Интервал, превышающий одинарный в 1.5 раза. Так, если размер шрифта равен 10 пунктам, интервал будет равен приблизительно 15 пунктам
Двойной интервал	Интервал, превышающий одинарный в два раза. Так, если размер шрифта равен 10 пунктам, интервал будет равен приблизительно 20 пунктам
Минимум	Интервал, который подбирается автоматически при вставке шрифтов больших размеров или графики, которые никаким другим образом не могли бы уместиться в заданном интервале
Точно	Фиксированный интервал, который не меняется в зависимости от размера шрифта. Применение этого параметра делает все строки одинаковой высоты
Множитель	Интервал, размер которого увеличивается или уменьшается на заданное количество процентов в зависимости от указанного пользователем значения

Интервалы

Междустрочный интервал (табл. 2.1) определяет расстояние между строками текста по вертикали. По умолчанию используется интервал, равный

одной строке. Выбранный межстрочный интервал применяется ко всем строкам выделенного абзаца или абзаца, в котором находится текущая позиция курсора.

Нумерованные и маркированные списки

Нумерованные и маркированные списки

Выполните команду **Список** из меню **Формат**, а затем выберите нужный формат списка. Для изменения выбранного формата нажмите кнопку **Изменить**. Кроме того, в окне диалога **Изменение маркированного списка** можно выбрать другой символ маркера, например, галочку. Для этого нажмите кнопку **Маркер**.

Нумерованные и маркированные списки могут создаваться автоматически при вводе текста. Перед началом ввода текста нажмите кнопку **Нумерованный список** или **Маркированный список** на панели форматирования, а затем вводите текст. При каждом нажатии на клавишу **Enter**, новый абзац будет начинаться со следующего порядкового номера или маркера.

Нумерация автоматически обновляется при добавлении элементов в нумерованный список, удалении элементов из него или при изменении порядка элементов в списке. Для того чтобы сформатировать текст в виде списка, выделите абзацы, составляющие список, и нажмите кнопку **Нумерованный список** или **Маркированный список** на панели форматирования.

Создание многоуровневого списка

- Выберите команду **Список** в меню **Формат**.
- Выберите нужный стиль списка на вкладке **Многоуровневый**, а затем нажмите кнопку **Ок**.
- Введите список, завершая ввод каждого элемента нажатием клавиши **Enter**. Каждый элемент списка должен находиться в отдельном абзаце.
- Чтобы выбрать уровень нумерации элемента, выделите его, а затем нажмите кнопку **Увеличить отступ** или **Уменьшить отступ**.

В многоуровневом списке можно изменять уровень нумерации в процессе ввода текста. Чтобы понизить абзац на один уровень, нажмите в начале этого абзаца клавишу **Tab**. Чтобы повысить абзац на один уровень, нажмите клавиши **Shift+Tab**. После этого введите текст абзаца.

Можно преобразовать уже существующий список в многоуровневый. Для этого выделите список, выберите команду **Список** в меню **Формат**, затем – нужный стиль нумерации списка на вкладке **Многоуровневый**. После этого установите уровни нумерации для каждого из элементов списка.

Преобразование маркеров в номера и наоборот

- Выделите элементы списка с маркерами или нумерацией, которые следует изменить.
- Нажмите кнопку **Нумерованный список** или **Маркированный список** на панели форматирования.

Пример трехуровневого списка

1. ВВЕДЕНИЕ
2. ОСНОВЫ ТЕОРИИ ЛИНЕЙНЫХ НЕПРЕРЫВНЫХ СИСТЕМ
 - 2.1. Общая характеристика автоматических систем
 - 2.1.1. Основные понятия и определения
 - 2.1.2. Функциональная схема замкнутой автоматической системы
 - 2.1.3. Классификация автоматических систем
 - 2.2. Типовые звенья систем радиоавтоматики
 - 2.2.1. Безынерционное (усилительное, пропорциональное) звено
 - 2.2.2. Инерционное звено (апериодическое звено первого порядка)
 - 2.2.3. Интегрирующее звено
 - 2.3. Передаточные функции систем радиоавтоматики
 - 2.3.1. Обобщенные функциональная и структурная схемы радиотехнической следящей системы
 - 2.3.2. Правила преобразования структурных схем
 - 2.3.3. Передаточные функции замкнутой системы
 - 2.4. Устойчивость автоматических систем
 - 2.4.1. Общие требования к устойчивости систем
 - 2.4.2. Алгебраические критерии устойчивости
 - 2.4.3. Частотные критерии устойчивости
 - 2.5. Показатели качества систем радиоавтоматики
 - 2.5.1. Оценка качества автоматических систем в переходном режиме
 - 2.5.2. Точность автоматических систем при типовых воздействиях
 - 2.5.3. Точность автоматических систем при воздействии помех
3. ТИПОВЫЕ СИСТЕМЫ РАДИОАВТОМАТИКИ
 - 3.1. Системы автоматической регулировки усиления
 - 3.2. Системы автоматической подстройки частоты
 - 3.3. Системы фазовой автоподстройки частоты
4. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Создание формул

В качестве примера построения формул создадим формулу

$$d^3 = \sqrt[3]{\left(\frac{12}{\sqrt{a^2 + b^2}}\right)^5}$$

Последовательность действий вычисления представлена в табл. 2.2

Таблица 2.2


Часть формулы	Описание
$d^3 =$	d в кубе и знак равенства
$\sqrt[3]{}$	Кубический корень
$()$	Скобки, ограничивающие степени
$\frac{12}{\sqrt{}}$	Дробь, содержащая в знаменателе квадратный корень
$a^2 + b^2$	Выражение, включающее буквы греческого алфавита

Запуск Equation Editor

Microsoft Equation – средство, предназначенное для создания формульных объектов в самом текстовом процессоре. При необходимости вставленный объект можно редактировать непосредственно в поле документа.

Чтобы создать формулу поступайте, следующим образом:

Поместите курсор вставки в том месте, где вы хотите разместить свою формулу.

Выберите команду **Вставка ⇒ Объект**. Из списка **Тип объекта** выберите *Microsoft Equation*. Щелкните по кнопке 

Запустите Equation Editor. На экране появится меню Equation Editor и панель инструментов **Формула**, показанная на рис. 2.3



Рис. 2.3. Панель инструментов

Панель инструментов **Формула** состоит из двух рядов кнопок. Верхний ряд кнопок содержит палитры символов. Из этих палитр вы можете

выбирать символы, которые Equation Editor вставит в место расположения курсора вставки. Второй ряд содержит палитры шаблонов. Эти палитры позволяют выбирать распространенные математические структуры.

Построение образца формулы

Чтобы приступить к построению формулы, сделайте следующее:

1. Введите с клавиатуры букву d . Equation Editor автоматически вставляет введенную букву в поле. Сама буква набирается курсивом, так как обозначения переменных в математических формулах всегда имеют курсивное начертание.

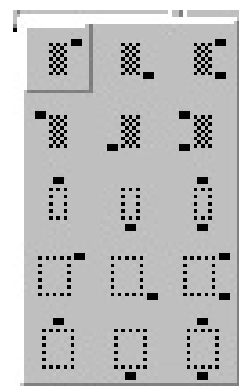


Рис. 2.4

2. Щелкните по третьей кнопке во втором ряду панели инструментов **Формула**, чтобы вывести на экран окно с доступными шаблонами степеней. Так как нам требуется возвести d в третью степень, выберите шаблон верхнего индекса (первый шаблон в первом ряду), как показано на рис. 2.4. Equation Editor разместит рядом с буквой d поле для показателя степени (он меньше по размерам, чем поле для обычной буквы, и поднят относительно нее).

3. Введите цифру 3 и нажмите клавишу **[Tab]**, чтобы выйти из поля верхнего индекса и вернуться в главное формульное поле. С помощью клавиши **[Tab]** осуществляется перемещение курсора вставки из одной части формулы в другую (например, из поля основания степени в поле показателя степени).

4. Теперь введите знак равенства. Таким образом мы закончили первую часть уравнения.

Размер любого элемента формулы можно менять с помощью команды **Размер \Rightarrow Определить**. В диалоговом окне **Размеры** вы можете указать новое значение размера для обычных символов, верхних и нижних индексов, специальных символов и любых определенных вами элементов.

Создание кубического корня

Чтобы выполнить построение кубического корня, сделайте следующее:

1. Щелкните по второй кнопке второго ряда панели инструментов **Формула**, чтобы вывести на экран список возможных конструкций с дробями и знаками корня. Нам из этого списка потребуется знак корня с указанием степени корня, который располагается во втором столбце четвертой строки, как показано на рис. 2.5. После вставки знака корня Equation Editor поместите курсор вставки в подкоренное выражение.

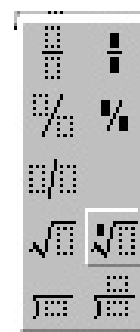


Рис.2.5

2. Нажмите клавишу **[Tab]**, чтобы переместиться в поле

показателя степени корня, и введите цифру 3.

Вставка формулы в подкоренное выражение

Если требуется ввести формулу под знак корня, используйте клавишу **[Tab]**, чтобы переместиться в нужное место уравнения. Для этого сделайте следующее:

1. Нажмите **[Shift]**, чтобы вернуться обратно в поле подкоренного выражения. Нажатие двух клавиш **[Shift]+[Tab]** перемещает курсор вставки по полям формулы в обратном порядке.

2. Находясь в поле под знаком корня, вставьте круглые скобки, выбрав первый шаблон из палитры скобок, показанной на рис. 2.6.

3. В поле, окруженное скобками, вставьте шаблон дроби. Для этого, поместив курсор вставки в поле, выберите шаблон дроби из той же палитры, как показано на рисунке справа.

4. В поле для числителя введите число 12.

Вместо вставки скобок в формулу с помощью шаблона скобок, вы можете просто ввести скобки с клавиатуры. Этот способ ничем не хуже вставки шаблона, – но только до тех пор, пока внутри скобок не располагается лишь один обычный символ, например (x). Если же элемент, окруженный скобками, превышает по высоте строчную букву, то вам придется прибегнуть к шаблону, поскольку вставленные с его помощью скобки будут автоматически увеличиваться по высоте в соответствии с высотой выражения, заключенного в них.

Построение формулы в знаменателе

Последний шаг в создании нашей формулы – создание знаменателя дроби. Как и раньше, используйте клавишу **[Tab]** для перемещения из числителя в знаменатель. Сделайте следующее:

1. Нажмите клавишу **[Tab]**, чтобы перейти в поле знаменателя.
2. Вставьте знак квадратного корня, выбрав его шаблон из палитры, как показано на рис. 2.7.
3. В поле подкоренного выражения введите a.
4. Добавьте поле для верхнего индекса и введите в этом поле цифру 2.
5. Нажмите клавишу **[Tab]** и наберите на клавиатуре + b.
6. Добавьте еще одно поле для верхнего индекса и введите в нем цифру 2.

Необходимо добавить показатель степени, относящийся к выражению в скобках. Для этого сделайте следующее:

1. Нажмите клавишу **[Tab]** четыре раза, чтобы вывести курсор вставки за пределы скобок.

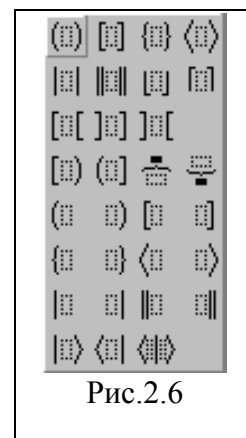


Рис.2.6

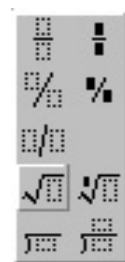


Рис.2.7

2. Добавьте поле для верхнего индекса и введите в нем цифру 5.
Формула готова.

3. Щелкните в любом месте в окне документа за пределами рамки с уравнением. Word уберет с экрана панель инструментов **Формула** и восстановит стандартное меню Word.

Порядок выполнения работы

1. Создать новый документ. Сохранить на своем диске в своей папке.

2. На первой странице своего документа сделать титульный лист, который содержит следующее: **ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**; Государственное образовательное учреждение высшего профессионального образования; Красноярский государственный технический университет; Лабораторные работы, по дисциплине «Информационные технологии»; выполнил, проверил, Красноярск – год. Сделать рамку.

3. Сделать следующий шаг: меню Вставка – Разрыв – на новую страницу.

4. Выполнить самостоятельно задание:

4.1. Набрать текст. Текст для работы выдает преподаватель.

4.2. Выделить четвертый и пятый абзацы.

4.3. Вызвать диалоговое окно **Абзац**.

4.5. Установить:

отступы слева и справа – 0 см;

первая строка – отступ на 0,7 см;

выравнивание – по ширине.

4.6. Выполнить форматирование эпиграфа:

выделить эпиграф (кроме строки с подписью);

вызвать диалоговое окно **Абзац**;

установить:

отступы слева – 9,5 см;

справа – 0 см;

первая строка: отступ – нет;

выравнивание: по левому краю.

4.7. Создать стиль на основе стиля **Основной** со следующими параметрами:

размер шрифта – 11 пт;

первая строка – красная с отступом 0,7 см;

отступы слева и справа – 0 см;

выравнивание – по ширине.

4.8. Дать стилю название – **Мой стиль**.

4.9. Применить созданный стиль для форматирования третьего абзаца.

4.10. Выделить первый и второй абзацы. Разделить выделенный текст на две колонки.

4.11. Изменить ширину колонок с помощью мыши.

4.12. Вставить буквицу в первом абзаце и в третьем абзаце.

5. Перед началом создания списков разного вида изучите подраздел теоретические сведения.

6. Самостоятельно сделать многоуровневый список, пример (см. подразд. теоретические сведения).

7. Перед созданием формул изучите подраздел теоретические сведения.

8. Наберите следующие формулы:

$$1) y \leq (a + kt^2)x, \quad 2) \hat{z} = z \pm r\delta, \quad 3) \vec{x} = x + \varepsilon \cdot c, \quad 4) \begin{cases} a \cdot x_1 + b \cdot x_2 = c \\ d \cdot x_1 + c \cdot x_2 = f \end{cases}$$

$$5) f'(x)_{x \in [0,1]} > 0 \Rightarrow f(0) < f(1), \quad 6) f(x) \rightarrow 0 \text{ при } x \rightarrow \infty, \quad 7) \sum_{i=1}^n \alpha_i x_i = S,$$

$$8) \frac{\partial f}{\partial x} = x^2 + \ln(x), \quad 9) \sqrt[5]{(x^3 + 17x)^4}, \quad 10) (x^2 + 3) / (x^2 + 2), \quad 11) \int_0^{\infty} f(x) dx = 6,$$

$$12) x_i \in \prod_{j=1}^n [\alpha_j, \beta_j], \quad 13) x \xrightarrow{r > 1} y \quad 14) \prod_{i=1}^m y_i = P, \quad 15) \Omega \subseteq \Theta \text{ I } \Psi,$$

$$16) \sum_{i=1}^n \sum_{j=1}^m x_i^3 + x_{ij} = S_j, \quad 17) \left. \begin{array}{l} \left[\alpha_0(t) \frac{\partial x}{\partial l} + \beta_0(t)x \right]_{l=0} = \eta_0(t); \\ \left[\alpha_1(t) \frac{\partial x}{\partial l} + \beta_1(t)x \right]_{l=L} = \eta_1(t). \end{array} \right\}$$

$$18) \|x\| = \left\| \sum_i x_i e_i \right\| \leq \alpha \sum_i |x_i| = \alpha \|x\|_1, \quad 19) \varphi(x) > 0 \text{ для всех } x \neq \emptyset;$$

$$20) \max_i |x_i| \leq \left(\sum_i |x_i|^2 \right)^{1/2} \leq \sqrt{n} \max_i |x_i|$$

$$21) \varphi(x + y) \leq \varphi(x) + \varphi(y)$$

$$22) u = - \frac{k}{\mu} (\nabla p + \rho g \nabla z)$$

$$23) - \operatorname{div}(\rho u) = \frac{\partial}{\partial t} (m\rho) + \tilde{q}.$$

9. Установите для 5-й формулы размер символов 14, размер крупного индекса – 12.

10. Установите для 16-й формулы размер символов 13, размер крупного индекса – 11, размер мелкого индекса – 9.

11. Записать выполненную работу в вашу рабочую папку.

Лабораторная работа № 3 ТЕКСТОВЫЙ ПРОЦЕССОР Microsoft Word

Цель работы: научиться создавать таблицы, вносить в них текстовую информацию, выполнять оформление таблицы и вычислять в них. Научиться создавать обычные и концевые сноски, гиперссылки, перекрестные ссылки. Научиться создавать предметный указатель и оглавление.

Теоретические сведения

Вычисления в таблице с помощью функций

Суммирование строки или столбца чисел

- Выделите ячейку, в которую следует поместить результат суммирования.

- В меню **Таблица** выберите команду **Формула**. Рабочее окно **Формула** изображено на рис. 2.8

- Если выбранная ячейка находится внизу столбца чисел, то предлагается формула = SUM(ABOVE). Если это правильно, то нажмите кнопку **ОК**.

Если выбранная ячейка находится в правом конце строки чисел, то предлагается формула = SUM(LEFT). Если это правильно, то нажмите кнопку **ОК**.

Ссылки на ячейки таблицы выглядят следующим образом: A1; B1; B2; и т. д., где буквами обозначаются столбцы, а цифрами – строки.

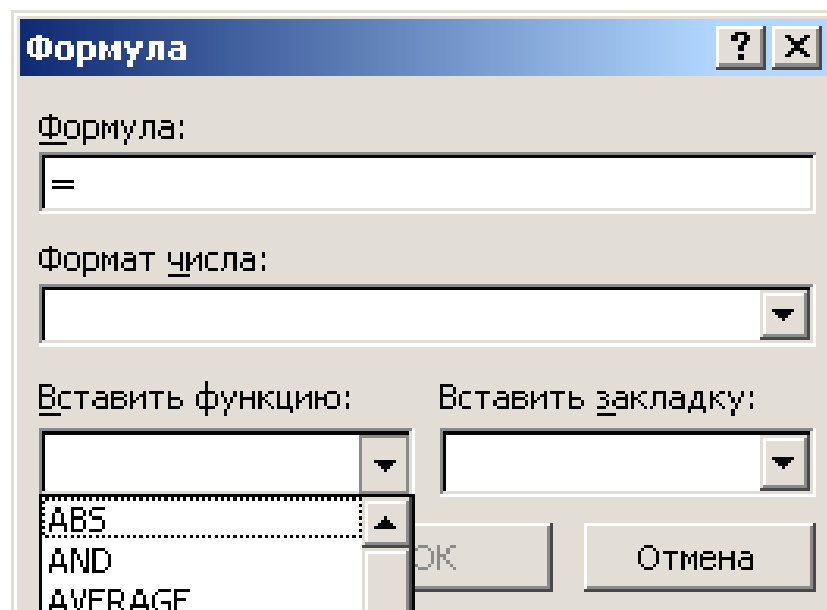


Рис. 2.8. Окно **Формула**

Таблица функций

Функция	Результат
ABS(x)	Абсолютное значение числа или формулы (без знака)
AND(x,y)	1 (истина), если оба логические выражения x и y истинны, или 0 (ложь), если хотя бы одно из них ложно
AVERAGE()	Среднее значений, включенных в список
COUNT()	Число элементов в списке
DEFINED(x)	1 (истина), если выражение x допустимо, или 0 (ложь), если оно не может быть вычислено
FALSE	0 (нуль)
INT(x)	Целая часть числа или значения формулы x
MIN()	Наименьшее значение в списке
MAX()	Наибольшее значение в списке
MOD(x,y)	Остаток от деления x на y
NOT(x)	0 (ложь), если логическое выражение x истинно, или 1 (истина), если оно ложно
OR(x,y)	1 (истина), если хотя бы одно из двух логических выражений x и y истинно, или 0 (ложь), если оба они ложны
PRODUCT()	Произведение значений, включенных в список. Например, функция $\{ = \text{PRODUCT}(1,3,7,9) \}$ возвращает значение 189
ROUND(x,y)	Значение x , округленное до указанного десятичного разряда (y). x может быть числом или значением формулы
SIGN(x)	Знак числа: 1 (если $x > 0$) или -1 (если $x < 0$)

SUM()	Сумма значений или формул, включенных в список
TRUE	1

Ввод сносок

Часто бывает необходимым пояснить значение какого-то слова. Для этого применяются сноски. Сноски бывают двух видов: расположенные в конце страницы и в конце документа – так называемые концевые сноски. Сноска состоит из двух частей: знака сноски и текста сноски. Текст сноски появляется в конце страницы или в конце документа. Для создания сноски используется команда **Сноска** из меню **Вставка**.

Сноски нумеруются автоматически, однако можно создавать собственные знаки сносок. Для просмотра текста сноски установите на знак сноски указатель и дважды нажмите кнопку мыши, а чтобы вернуться обратно, установите указатель на знак сноски, расположенной слева от текста сноски в окне сносок, и дважды нажмите кнопку мыши.

Для удаления сноски выделите ее знак и нажмите на клавишу **Delete**. Если сноски были автоматически пронумерованы, то после удаления сноски будет выполнена перенумерация.

Чтобы переместить сноску, выделите знак сноски и перетащите его на новое место с помощью мыши. Для копирования сноски сделайте эту операцию при нажатой клавише **Ctrl**.

Для отделения текста документа от текста сносок используется горизонтальная линия. Если сноски переходят на следующую страницу, печатается более длинная линия, называемая разделителем продолжения, к которой можно добавить текст, такой как «**Продолжение на следующей странице**». Допускается изменять разделители, снабжая их границами, текстом и рисунками. Формат разделителя сносок может быть изменен путем добавления обрамления, текста или графического объекта. В обычном режиме просмотра выполните команду **Сноски** из меню **Вид**. В списке **Сноски**, находящемся в верхней части области сносок, выберите разделитель, который нужно изменить, а затем отредактируйте его в области сносок.

Для изменения знака сноски выберите команду **Сноска** в меню **Вставка**. Установите флажок **Сноска** или **Концевая сноска**. В поле **Символ** укажите нужный знак.

Концевые сноски обычно используются для больших документов по объему ссылок и пояснений и размещаются в конце документа. Чтобы создать концевую сноску выполните команду **Сноска** из меню **Вставка**, а затем выберите команду **Концевая сноска**. В одном документе можно

использовать обычные и концевые сноски. Кроме того, обычные сноски могут быть преобразованы в концевые и наоборот.

Ввод перекрестных ссылок, названий

Перекрестные ссылки

Чтобы сослаться на элемент, расположенный в другом месте документа, используется перекрестная ссылка. Перекрестные ссылки создаются для заголовков, стандартных стилей, списков, закладок и названий. Чтобы создать перекрестную ссылку, выполните команду **Перекрестная ссылка** из меню **Вставка**.

Для создания перекрестной ссылки:

- Введите в документ вступительный текст, с которого будет начинаться перекрестная ссылка, например: «**Для получения подробных сведений см**».

- В меню **Вставка** выберите команду **Перекрестная ссылка**.

- В поле **Тип ссылки** выберите тип элемента, на который требуется сослаться, например, заголовок раздела или таблицы.

- В поле **Вставить ссылку на** выберите данные, которые хотите вставить в документ, например, Текст заголовка.

- В поле **Для какого заголовка** выберите тип элемента, на который надо сослаться. Так, чтобы выбрать значение заголовка в поле **Тип ссылки** для документа, имеющего шесть заголовков, выберите именно тот заголовок, который требуется. Нажмите кнопку **Вставить**.

Названия

Word может автоматически добавлять нумерованные названия при вставке рисунков, таблиц, диаграмм Excel и других объектов. Например, каждой вставляемой в документ таблице могут быть добавлены названия: **Таблица 1**, **Таблица 2** и т. д. Нумерованные названия могут быть добавлены также к уже существующим элементам документа. После того, как элементу добавлено название, на него можно сослаться в перекрестной ссылке. Для вставки названия выделите элемент, которому необходимо добавить название, а затем выполните команду **Название** из меню **Вставка**.

Для автоматического добавления названия к таблице, рисунку, формуле и другим элементам во время их вставки:

1. Выберите элементы, название к которым должно добавляться автоматически при их вставке в документ.

2. В меню **Вставка** выберите команду **Название**.

3. Нажмите кнопку **Автоназвание**.

4. Установите нужные параметры.

Для добавления названия к уже созданным таблице, рисунку, формуле и другим элементам:

1. Выберите элемент, к которому нужно добавить название.
2. В меню **Вставка** выберите команду **Название**.
3. Установите нужные параметры.

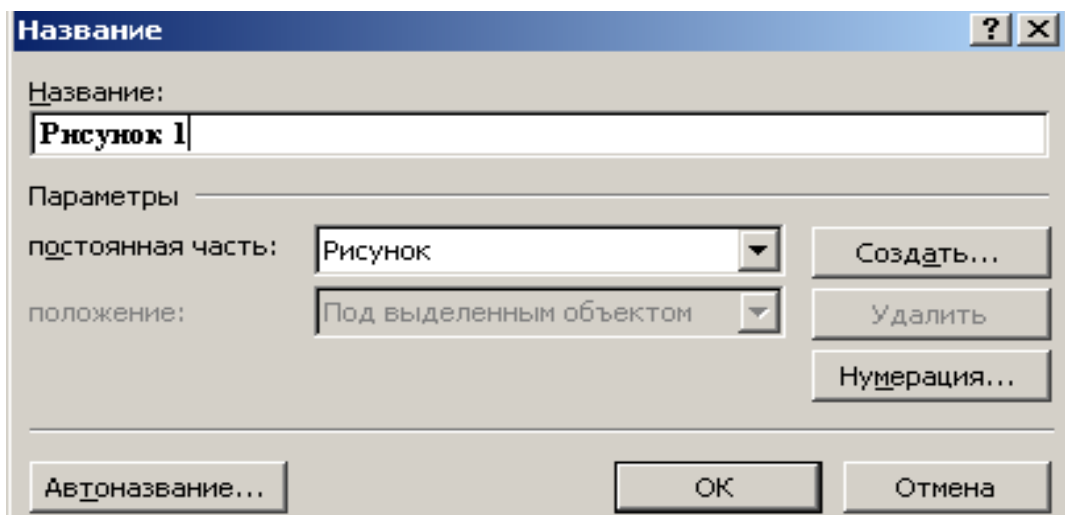


Рис. 2.9. Окно **Название**

Предметный указатель

Предметные указатели позволяют быстро найти нужные слова или предложения в документе по номерам страниц.

Создание предметного указателя

Чтобы создать предметный указатель, надо отметить текст или элементы для включения в указатель, выбрать формат, после чего собрать указатель. Допускается создавать перекрестные ссылки, словарь для автоматического отбора текста в документе, а также изменять или удалять существующие элементы предметного указателя.

Для разработки и сборки указателя:

- Создайте элементы указателя в документе.
- Установите курсор в место, куда хотите вставить готовый указатель.
- В меню **Вставка** выберите команду **Оглавление и указатели**, после чего выберите вкладку **Указатель**.
- Выберите вариант встроенного форматирования указателя в поле **Форматы**.

Для использования помеченного текста в указателе:

- Выберите в документе слово или предложение, которое следует сделать элементом указателя. Чтобы в качестве элемента указателя ввести

любой текст, поместите курсор в место, куда требуется вставить элемент указателя.

- Нажмите клавиши **Alt+Shift+X**.
- Чтобы создать основной элемент указателя, отредактируйте имеющийся или введите новый текст в поле **Основной элемент**. Чтобы создать дополнительный элемент указателя, введите его в поле **Дополнительный**. Чтобы добавить элемент третьего уровня, введите текст дополнительного элемента указателя в поле **Дополнительный**, поставьте двоеточие, а затем введите текст элемента третьего уровня.

- Нажмите кнопку **Пометить**. Чтобы отметить все вхождения текста в документе, нажмите кнопку **Пометить все**.

- Продолжайте отмечать нужный текст. До окончания выбора элементов указателя окно диалога **Определение элемента указателя** остается открытым.

Команда, выполняемая по нажатию кнопки **Пометить все**, отмечает первое вхождение в каждом абзаце искомого текста с учетом регистра букв.

Все отмеченные элементы указателя вставляются в виде полей {XE} в формате скрытого текста. Если поле {XE} не видно, нажмите кнопку **Непечатаемые символы**.

Для создания перекрестной ссылки в предметном указателе:

- Выберите текст для использования в качестве элемента указателя.
- Нажмите клавиши **Alt+Shift+X**.
- В группе **Параметры** выберите значение **Перекрестная ссылка**, после чего введите текст, который хотите использовать в качестве перекрестной ссылки на элемент указателя. Слово **Смотри** вставлено в поле **Перекрестная ссылка** по умолчанию. Это слово можно удалить или добавить к нему дополнительный текст.

- Нажмите кнопку **Пометить**.

Для того чтобы создать перекрестную ссылку, нужно:

- В меню **Вставка** выполнить команду **Название**, выбрать из списка нужное название объекта документа или создать свое;
- Вернуться в то место документа, откуда будет происходить ссылка;
- В меню **Вставка** выполнить команду **Перекрестная ссылка**;
- В окне **Тип ссылок** выбрать свое название, поставить нужные параметры. Нажать кнопку **Вставить**.

Для того чтобы создать гиперссылку, нужно:

- В меню **Вставка** выполнить команду **Гиперссылка**;
- Ввести имя файла или нажать кнопку **Файл** и выбрать нужное имя файла.
- Нажать кнопку **Ок**.

Создание элемента предметного указателя, содержащего ссылку на диапазон страниц

Для создания элемента предметного указателя, содержащего диапазон страниц:

□ Выберите диапазон текста, на который должен ссылаться элемент указателя, а затем выполните команду **Закладка меню Правка**.

□ Введите имя в поле **Имя закладки**, а затем нажмите кнопку **Добавить**.

□ Введите имя в поле **Имя закладки**, а затем нажмите кнопку **Добавить**.

□ Поместите курсор в текст документа, отмеченный закладкой.

□ Нажмите клавиши **Alt+Shift+X**. Введите элемент указателя для отмеченного текста в поле **Основной элемент**. Выберите значение диапазона страниц. В поле **Закладка** введите или выберите имя закладки для диапазона страниц, указанное на шаге 2. Нажмите кнопку **Пометить**.

Пример предметного указателя

А		Б	
абзац	4	байт	5
автозавершение	10	база данных	15
автозамена	7	безопасность	14
архиватор	15	базовые ИТ	54
атрибуты	58	базовый процесс	47
автоформа	15	броузер	54
алгоритм	58	буфер	47

Оглавление

Оглавление содержит список заданных заголовков документа с соответствующими номерами страниц.

Программа MS Word включает несколько вариантов встроенного форматирования и позволяет осуществлять предварительный просмотр оглавления до его создания.

Создание оглавления

Оглавление создается с помощью стилей, которыми оформлены заголовки документа. Можно использовать встроенные или создать собственные стили заголовков.

Для разработки и создания оглавления с использованием встроенных стилей заголовков:

□ Наложите встроенные стили **Заголовок 1 – Заголовок 9** на заголовки, которые собираетесь отобразить в оглавлении.

□ Поместите курсор в место, куда хотите вставить оглавление.

В меню **Вставка** выберите команду **Оглавление и указатели**, после чего выберите вкладку **Оглавление**. Поместите курсор в место, куда хотите вставить указатель оглавления.

Пример оглавления вашего лабораторного практикума

Титульный лист	1
Лабораторная работа № 1	2
СОЗДАНИЕ ТИТУЛЬНОГО ЛИСТА	
ФОРМАТИРОВАНИЕ ТЕКСТА	
СОЗДАНИЕ НУМЕРОВАННОГО И МНОГОУРОВНЕВОГО СПИСКА	
СОЗДАНИЕ ФОРМУЛ	
Лабораторная работа № 2	10
ВЫЧИСЛЕНИЯ В ТАБЛИЦЕ	
СОЗДАНИЕ ДИАГРАММЫ	
СОЗДАНИЕ ОБЫЧНЫХ И КОНЦЕВЫХ СНОСОК, ГИПЕРССЫЛОК И ПЕРЕКРЕСТНЫХ ССЫЛОК	
СОЗДАНИЕ ПРЕДМЕТНОГО УКАЗАТЕЛЯ И ОГЛАВЛЕНИЯ	

Порядок выполнения работы


Для создания таблицы расписания занятий (рис. 2.10) выполните следующие шаги.

1. Вставьте таблицу с требуемым числом ячеек.
2. Столбец с нумерацией лент выровняйте влево и при помощи мыши установите необходимую ширину. Все остальные ячейки выровняйте по центру.
3. Дни недели и номера лент выделите полужирным шрифтом. Форматирование шрифта можно произвести и после набора текста.


№	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
1	Высшая математика	Физическая культура	Физика	ОБЖ	Химия	Английский язык
2	Физика	История	Информатика	Физика	МХК	Экология
3	Русский язык	Высшая математика	Информационные технологии	Высшая математика	Информатика	Физическая культура
4	Английский язык			Химия	Русский язык	

Рис. 2.10. Таблица расписаний занятий


4. Заполните остальные ячейки, придерживаясь следующего требования: **на первом уроке** должны быть «короткие» названия уроков, **вписывающиеся в одну строчку** (например, физика) и «длинные» – **в две строчки** (например, физическая культура), **на втором уроке** все предметы должны помещаться **в одну строчку**, остальные уроки произвольно.


5. Выполните **Просмотр Файл⇒Предварительный просмотр**, используйте «лупу» (щелчок мышью) для увеличения масштаба просмотра (или используйте кнопку **Предварительный просмотр**  на панели инструментов). Видно, что таблица не разлинована.

6. Закройте **Просмотр** (кнопка **Заккрыть**).

7. Выделите таблицу и выполните **обрамление**. Для этого нажмите кнопку **Таблицы и граница**  на панели инструментов. После этого появляется окно *Таблицы и граница*, в котором можно выбрать формат таблицы и параметры ее оформления. Выделяя таблицу, следите за тем, чтобы в выделение не попал маркер абзаца, следующего за таблицей.

Нажмите кнопку **Обрамление** и в появившемся списке нажмите кнопку **Все границы** (рис. 2.11).

8. Выполните **Просмотр**. Для этого нажмите кнопку  на панели инструментов. В результате вы увидите, что **обрамление** всей таблицы выполнено линиями одной толщины.

9. На данном примере можно попробовать способы вставки и удаления строк и столбцов. Выделив строку (столбец), щелкните мышью по кнопке  (только для вставки) или воспользуйтесь командой **Таблица⇒Вставить (удалить) строки (столбцы)**.

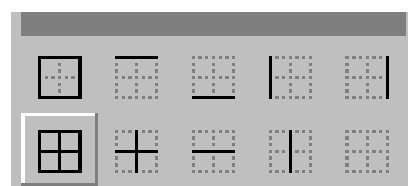




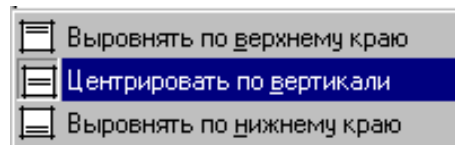


Рис. 3.11. Выбор границ
обрамления таблицы

Примечание: Для **обрамления** таблицы можно воспользоваться кнопкой **Обрамление**  на панели инструментов. Если нажать на стрелку, расположенную слева от этой кнопки, то появляется меню. Для того чтобы нарисовать границу клетки (или группы клеток), надо выделить эту клетку (или группу клеток) и нажать кнопку с необходимой линией. Например, если линия должна находиться сверху клетки, то надо нажать кнопку , если линия должна быть слева – кнопку . Для выбора типа и толщины линии необходимо воспользоваться командами, представленными в окне *Таблицы и граница*.


10. Расширьте первую строку. Подведите курсор мыши к нижней границы первой строки и, когда он приобретет вид курсора изменения размера (две стрелки вверх и вниз и две горизонтальные линии), нажмите левую кнопку мыши и, удерживая ее, растяните первую строку на 1,5 см вниз. В результате строка станет широкой, а названия дней недели будут расположены вверху строки.



11. Для размещения названия дней недели по центру клеток выделите первую строку и нажмите кнопку **Таблицы и граница**  или выполните команду контекстного меню **Выравнивание**⇒**Центрирование по вертикали** (рис. 2.12).

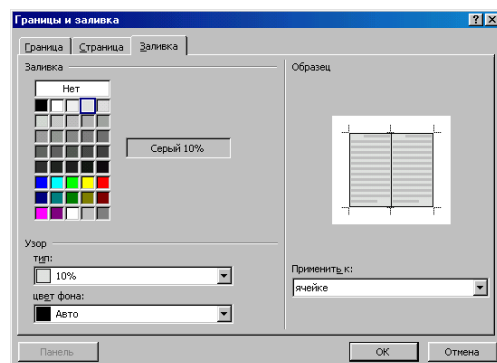


№	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
1						
2						
3						
4						

Рис. 2.12. Пример таблицы, которая должна получиться.

12. Разместим номера уроков по центру клетки. Для размещения содержимого столбца по центру необходимо выделить столбец и нажать кнопку **По центру**  на панели инструментов. Для размещения номеров по центру строки выполните команду контекстного меню **Выравнивание**⇒**Центрирование по вертикали**.

13. Для того чтобы установить требуемый вид границы оформления можно воспользоваться командами окна **Таблицы и оформления**. Выделим границу таблицы, Для этого выделите таблицу и нажмите кнопку **Таблицы и граница** . В списке *Типа линий* выберите *Двойную линию*, а в списке *Толщины линий* установите толщину 1,5 пт. После этого выберите границу оформления **Внешние границы** .



14. Для фонового оформления выделите нужные ячейки и выполните команду **Формат**⇒**Границы и заливка...** В появившемся окне **Граница и заливка** перейдите на страницу **Заливка** и выберите необходимый тип *Узора* или цвет *Заливки*. Выполните *Просмотр*, чтобы остаться довольными результатами своей работы. Самостоятельно выбирая различные значения формата таблицы, поэкспериментируйте с созданной таблицей (рис. 2.13).

15. Самостоятельно создайте таблицу и выполните вычисления с помощью функций (табл. 2.3). В меню **Таблица** выбрать команду **Формула** (рис. 2.8). Задание выдает преподаватель.

16. Создайте диаграмму на основе полученной таблицы.

Выделите часть таблицы или всю таблицу. В меню **Вставка** выполните команду **Рисунок – Диаграмма**. Выберите параметры форматирования диаграммы. Перейдите обратно в документ.

	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
1						
2						
3						
4						

Рис. 2.13. Пример таблицы

17. На следующей странице своего документа наберите текст.

18. Создайте 5 обычных и 5 концевых сносок (прочитать подраздел теоретические сведения).

19. Создайте 2 гиперссылки и 4 перекрестных ссылки (прочитать подраздел теоретические сведения).

20. Перед началом создания предметного указателя изучите подраздел теоретические сведения.

21. Создайте оглавление своего документа (прочитать подраздел теоретические сведения).

Лабораторная работа № 4

ИЗДАТЕЛЬСКАЯ СИСТЕМА Adobe Acrobat Professional

Цель работы: научиться оформлять документ в PDF-формат. Освоить технологию создания ссылок в Adobe Acrobat Professional.

Теоретические сведения

Формат PDF является стандартом для международного распространения электронных документов. Это универсальный файловый формат, который обеспечивает сохранение всех шрифтов, форматирование и графику любого исходного документа, независимо от используемых приложений и платформы. Документами в формате PDF в большинстве случаев, являются книги, статьи, учебные материалы, отчеты, предназначенные для архивации, пересылки и обмена (но не редактирования и обработки).

Формат PDF разработан фирмой Adobe, а специализированный пакет Adobe Reader этой фирмы обеспечивает возможность просмотра документа PDF. Пакет этот свободен и доступен на сайте www.adobe.com. Кроме того, он встроен в браузер Internet Explorer. Соответственно для просмотра документа PDF можно установить и использовать пакет Adobe Reader или использовать браузер Internet Explorer.

Ориентация документов PDF на архивацию, обмен и просмотр (а не на подготовку, редактирование и обработку) обеспечивает возможность целенаправленной оптимизации формата, направленной на компактное и безопасное представление документа.

Документы PDF компактны, их файлы в несколько раз меньше файлов исходных документов. Более того они поддерживают большинство PDF поддерживает большое количество алгоритмов компрессии данных.

PDF поддерживает безопасность. Создатель PDF-файла может задать несколько опций безопасности. Возможно заблокировать файл так, что он может быть открыт только после ввода пароля. Также возможно запретить изменение файла и/или его печать

PDF-файлы машинно-независимы. Это означает, что документ можно увидеть точно таким же, каким он был создан, вне зависимости от того, на каком компьютере и в какой операционной среде он создавался.

Требования к оформлению документов в PDF-формате установлены в стандарте предприятия СТБ 4.2.10-2006. Требования стандарта обеспечивают унификацию и единство внутреннего представления документа в *pdf*. Стандарт описывает оформление объектов и регламентирует настройку свойств электронных ресурсов при реализации их в формате *pdf*.

Порядок выполнения работы:

1. Загрузить программный продукт **Adobe Acrobat Professional**
2. Выбрать **Open** (Открыть) в меню File документ с расширением. **PDF**
3. Нажать на кнопку **Hand Tool** (Инструмент Рука)
4. Перейти на определенную страницу
5. Нажать правую кнопку мыши
6. Выбрать **Add Bookmark** (Создать закладку)
7. Переименовать закладку, появившуюся в левом окне на определенное название страницы
8. Перейти на следующую страницу и повторить пункты 6 –7 до завершения содержания документа.

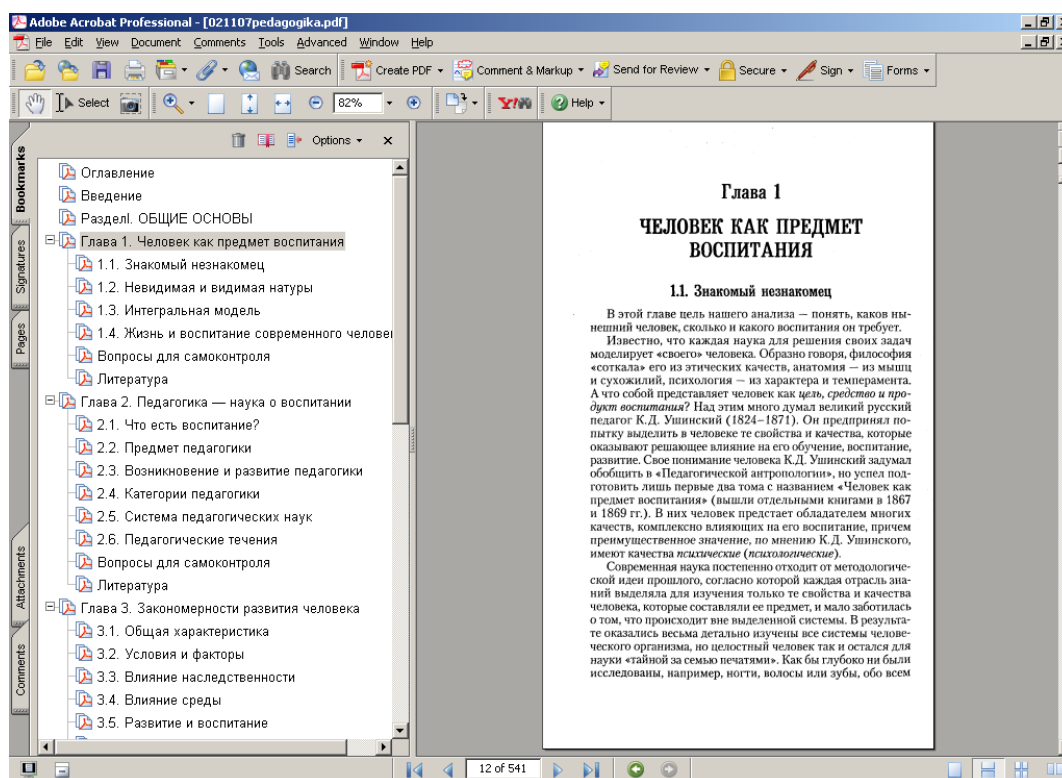


Рис. 2.14. Пример документа оформленного в PDF-формат

Современные информационные технологии позволяют включать в состав документа любые мультимедийные объекты (графику, звук, анимацию, видео), а также создать фотоархив, отсканировав старые фотографии и поместив их в упорядоченном виде в компьютерный фотоальбом, в процессе обучения вы можете подготовить реферат с иллюстрациями, компьютерную презентацию. Умение работать с мультимедиа документами, создавать компьютерные презентации становится важным в информационном обществе.

Лабораторная работа № 5 ТАБЛИЧНЫЙ ПРОЦЕССОР MICROSOFT EXCEL

Цель работы: научиться редактировать текст; выполнять простейшие вычисления; знать типы данных допустимые операции над ними; научиться копировать формулы и текст, делать вставки, удаление строк и столбцов, изменять размеры строк и столбцов, знать эргономическое оформление таблиц (автоформат). Создавать собственный стиль.

Теоретические сведения

Для представления данных в удобном виде используют таблицы. Компьютер позволяет представлять их в электронной форме, а это дает возможность не только отображать, но и обрабатывать данные. Класс программ, используемых для этой цели, называются электронными таблицами. Особенность электронных таблиц заключается в возможности применения формул для описания связи между значениями различных ячеек.

ЭЛЕКТРОННАЯ ТАБЛИЦА (ЭТ, рабочий лист) разбита на столбцы и строки, на пересечении которых находятся клетки. ЭТ имеет обрамление (верхняя строка и левый столбец) с идентификаторами (именами) строк и столбцов, представленных на экране.

РАБОЧАЯ КНИГА состоит из рабочих листов. Количество листов в рабочей книге ограничивается только объемом памяти (максимальное число листов книги, открываемой по умолчанию, – 255). При запуске на экране появится рабочий лист (Лист 1), а в нижней части экрана указывается, что на очереди Лист 2, Лист 3 и т. д. Максимальный размер листа: 16384 строки, 256 столбцов.

СТОЛБЦЫ обозначены одно- и двухсимвольными буквами латинского алфавита: А, В, С, ... Z, АА, АВ, ... AZ, и т.д.

СТРОКИ обозначаются цифрами.

ЯЧЕЙКА (клетка) образована пересечением строки и столбца, она имеет свой собственный адрес (или имя), составленный из буквы столбца и номера строки. Оказавшись, например, в левом верхнем углу – на пересечении столбца А и строки 1, – вы можете сказать, что находитесь в ячейке А1. Сделав один шаг вправо, окажетесь в ячейке В1. В окне **Имя** можно увидеть какая ячейка выбрана (является активной).

Клетка – это основной объект хранения данных и характеризуется: *адресом*. Адрес может быть:

ОТНОСИТЕЛЬНЫМ – изменяется при копировании содержимого клетки. Пример относительного адреса А2.

АБСОЛЮТНЫМ – не изменяется ни при каких перемещениях содержимого клетки. Пример: \$А\$2 – абсолютный адрес клетки А2;

СОДЕРЖИМЫМ – клетка может быть пустой или содержать данные. Данные записываются в клетку в виде строки текста или формулы;

ФОРМАТОМ – отображения на экране содержимого клетки. Можно задавать произвольную ширину отображения столбца клеток, расположение данных в клетках, менять количество знаков в числах, а также производить их масштабирование, дополнять числа специальными знаками типа \$, % и др.;

СТАТУСОМ – клетка может быть защищена или не защищена от записи новых данных.

БЛОК (ИНТЕРВАЛ) клеток – прямоугольная область ЭТ, задающаяся адресами верхней левой и правой нижней клеток области. Разделителем адресов угловых клеток служит двоеточие (например, A1:B15).

Объектами действия команд могут являться клетка, столбец, строка, диапазон столбцов либо строк или блок клеток.

Главное меню

Программа имеет девять ниспадающих меню. Каждое из этих меню обозначено единственным словом па линейке главного меню:

1. **Файл.** Сохраняет результаты вашей работы; находит ранее созданные файлы; посылает материал на распечатку.

2. **Правка.** Переносит, копирует и удаляет любые фрагменты текста; производит поиск слов или предложений.

3. **Вид.** Позволяет просмотреть результаты вашей работы различными способами. Можно вызвать на экран или убрать панели инструментов.

4. **Вставка.** Позволяет дополнительно внести специальную информацию. Добавлять строки, столбцы, диаграммы и др.

5. **Формат.** Позволяет оформить вашу работу более привлекательно, изменяя ширину, высоту ячеек, шрифт рамки и др.

6. **Сервис.** Выполняет особые задачи типа проверки правописания. В этом режиме вы также можете изменить параметры программных установок.

7. **Данные.** Позволяет сортировать и фильтровать данные по критерию, создавать сводные таблицы.

8. **Окно.** Производит переключение из одного окна в другое либо видоизменяет уже открытые окна.

9. **Справка.**

Правила работы с ячейками в Excel

Набирая буквы или цифры, вы видите в ячейке именно их (табл. 3.4). А вот если начнете ввод данных со знака равенства, то программа будет обрабатывать эти данные как формулу. Она запомнит эту формулу, но в ячейке покажет лишь результат своих вычислений. Так, напечатав =2+2 и нажав клавишу Enter, в ячейке вы увидите 4.

Таблица 2.4

Типы данных и результаты операций с ними

Тип данных	Значение	Результат обработки
Числа в любом формате	458 \$156,74 5%	Число
Буквы и цифры	Скидка 5 % 16 штук Расход,	Текст
Дата или время в любом виде	1/20/05 13:30 10 ноября	Дата /Время
Любое выражение, начинающееся со знака равенства	=5+25 =B3*0,5 =СУММ(B4:B15)	Формула

В табл. 2.4 приведены некоторые типы данных, которые можно ввести в ячейку Excel, и указано, что программа проделает с ними.

Ввод данных

Включите компьютер. Выберите иконку Excel. Введите данные в соответствующие клетки, как показано на рис. 2.15. При этом пользуйтесь следующими правилами:

- данные вводятся всегда в активную клетку, поэтому постоянно следите за ее положением. Перемещение активной клетки (АК) осуществляется с помощью стрелок или с помощью «мыши». Зафиксируйте указатель мыши на клетке А3 – клетка станет активной;

- введите данные, убедитесь в правильности набора и лишь после этого нажмите Enter. Данные запишутся в АК;

- чтобы убедиться в правильности ввода данных, необходимо установить АК в проверяемую клетку и в строке «формула» Вы увидите содержимое клетки;

- формульное выражение вводится только латинским алфавитом;
- неправильно заданную формулу система распознает как текст. В строке редактирования, текст вводится начиная с позиции курсора ввода, который можно перемещать с помощью клавиш управления курсором на клавиатуре.

На практике более удобно позиционировать курсор ввода с помощью мыши. Для этого необходимо установить указатель мыши в нужной позиции в тексте и зафиксировать его. Чтобы завершить ввод, сохранив введенные данные, используют кнопку Enter в строке формул или клавишу ENTER.

Чтобы отменить внесенные изменения и восстановить прежнее значение ячейки, используют кнопку **Отмена** в строке формул или клавишу ESC.

	A	B	C	D
1	Курс \$	30		
2	Наименование	Цена за ед..\$	Кол. продан.	Доход в руб.
3	Стол	375	7	=B\$1*B3*C3
4	Стул	48	28	=B\$1*B4*C4
5	Итого			=СУММ(D3:D4)

Рис.2.15. Пример электронной таблицы

Редактирование данных

Для исправления неправильно введенных данных следует установить курсор АК в редактируемую клетку. Далее необходимо либо ввести новые данные, либо, зафиксировав курсор мыши в строке формул (или нажать F2), исправить ошибку.

При этом можно использовать клавиши:

для перемещения курсора по таблице – стрелки или курсор мыши;

для стирания символа над курсором – клавишу Del;

для стирания символа, стоящего перед курсором – клавишу BackSpace;

для переключения режимов ВСТАВКА/ЗАМЕНА – Ins.

Если ввод данных или команда не получается, то нажмите клавишу Esc. Тем самым вы переведете систему в исходное состояние.

Как выделить ячейки интервала

Прежде, чем выполнить те или иные операции с помощью Excel – форматировать ячейку, удалять строки или столбцы либо переносить интервал с одного места на другое, необходимо сначала выделить требуемый объект. Проще это сделать с помощью мыши.

Работая с одной ячейкой, действуйте следующим образом:

- Чтобы выделить ячейку, укажите на нее и щелкните мышью.

- Чтобы поместить точку ввода в ячейку, укажите и дважды щелкните мышью.

Чтобы выделить символы в ячейке, дважды щелкните в ней, а затем проведите указателем мыши по этим символам.

Чтобы выделить слово в ячейке, вначале выделите ее, а затем дважды щелкните на слове (это можно делать также с адресами ячеек, аргументами формул и другими объектами, не являющимися словами в обычном понимании).

Если вы собираетесь работать более чем с одной ячейкой, в частности с **интервалом**, выполните следующие действия:

Чтобы выделить интервал, щелкните мышью в ячейке в одном углу интервала и выполните потягивание мышью до противоположного угла.

Чтобы выделить группу несвязанных ячеек, выберите первую ячейку группы и, удерживая клавишу Ctrl, выделите следующую ячейку. Клавишу Ctrl следует удерживать нажатой до тех пор, пока не будут выбраны все намеченные клетки. Чтобы выделить целый столбец (строку), просто щелкните мышью на букве (числе) его (ее) заголовка. Чтобы выделить несколько строк или столбцов, выделите первую строку (столбец) и, удерживая кнопку мыши, выполните протягивание через требуемые строки (столбцы).

Правила задания имен ячеек, интервалов

Можно использовать в имени до 255 символов, но все же лучше называть их покороче.

Первым символом должна быть буква или символ подчеркивания. Нельзя, например, называть ячейку «1йКварталПродаж», а вот «К1Продаж» вполне подойдет.

Остальными символами могут быть буквы, цифры, точки или символы подчеркивания, все прочие знаки препинания запрещены.

Пробелы в имени запрещены. Нельзя, например, называть ячейку «Прибыль за март», а «Прибыль_за_март» подойдет.

Имя ячейки или интервала не должно выглядеть как адрес ячейки. Например, нельзя ячейке H4 дать имя Q3.

Вставка (удаление) ячеек, строк и столбцов

Наилучший способ вставить или удалить какой-либо объект – это воспользоваться контекстным меню, вызываемым нажатием правой кнопки мыши (курсор мыши должен находиться на ячейке, строке, столбце или выделенной области). Вставка строк производится выше указанного объекта, а вставка столбцов – слева от объекта. Вызовите на экран контекстное меню и в нем выберите команду **Вставить** или **Удалить**. На экране появится диалоговое окно, с помощью которого, можно указать, каким образом вы можете переделать свой рабочий лист.

Вырезание, копирование и вставка данных

Перенести или скопировать **выделенный** фрагмент за пределами текущего окна удобнее всего с помощью **правой** кнопки мыши, нажатием на которую вызывается контекстное меню Excel. Вырезание, копирование интервала и его вставка выполняются следующим образом:

1. укажите курсором мыши на выделенный интервал, щелкните правой кнопкой мыши и выберите команду **Вырезать** или **Копировать** из контекстного меню. По щелчку мышью на одной из этих команд меню

граница выделенного фрагмента начнет двигаться подобно «бегущим огонькам»;

2. укажите курсором мыши на ту ячейку, куда хотите вставить свой фрагмент, и щелкните правой кнопкой;

3. выберите из контекстного меню команду **Вставить**, чтобы поместить фрагмент на выбранное место (или нажмите клавишу Esc, если решили отказаться от вставки).

Как производить вычисления

Простые формулы позволяют складывать, вычитать, перемножать и делить числа, хранящиеся в других ячейках. Например, в своем рабочем листе можно поместить все полученные доходы в один столбец, а затем создать формулу, которая вычислит сумму всех этих значений и поместит ее в ячейку внизу столбца. С этого момента стоит вам только изменить одно из значений в столбце, как нижняя строка будет тут же автоматически скорректирована.

Если вы собираетесь использовать постоянную величину во всем рабочем листе, поместите ее значение в одну из ячеек своей таблицы. Теперь можно просто ссылаться в других формулах на эту ячейку, вместо того чтобы каждый раз заново впечатывать одну и ту же величину. К примеру, вы можете внести текущий курс \$ в ячейку B1. И теперь, вместо того чтобы впечатывать это значение в каждую ячейку, где вычисляются доход от продажи, вы всего лишь вставляете адрес ячейки B1. А когда курс меняется, вы просто изменяете содержимое B1 и все ваши расчеты будут автоматически скорректированы.

ВЕЛИЧИНА – это любая информация, которая не изменяется, если только вы сами не решите ее изменить. Это может быть число (42), дата (4/9/05) или даже фрагмент текста («Наименование изделия»).

ФОРМУЛА – это сочетание величин, адресов ячеек, знаков математических действий (операторов), а также специальных функций.

При вводе формул вы должны следовать следующим принятым в Excel правилам:

формула **обязательно** начинается со знака равенства;

она может содержать- постоянные величины, знаки математических действий, функции Excel, адреса ячеек и их интервалов, и даже другие формулы, причем все это в любых сочетаниях;

в формулах можно использовать следующие знаки математических действий: сложения (+), вычитания (-), умножения (*), деления (/), вычисления процента (%) или возведения в степень;

для управления порядком действий пользуйтесь скобками. Excel выполняет умножение и деление прежде, чем сложение и вычитание;

длина формулы не более 1024 символа.

Чтобы автоматически превратить обыкновенную дробь в десятичную, введите сначала ноль и пробел, в противном случае Excel превратит некоторые дроби в даты (например, 5/8 станут 8 мая), а другие – в метки текста. Но если вы введете в ячейку 0 5/8, Excel сохранит это значение как 0,625 и отобразит его на экране.

Оформление рабочих листов

При вводе данных в ячейку Excel всегда проверяет нет ли каких-либо указаний на то, как следует отображать содержимое этой ячейки. Если Excel не находит никаких указаний, он анализирует данные, чтобы представить их в наиболее удобном для пользователя виде. Преобразования, которые Excel произведет с оформлением символом в ячейке, называются *форматированием*.

Вы можете пользоваться инструкциями по форматированию, чтобы точно указать Excel, в каком виде следует показывать символы в данной ячейке или интервале.

На панели инструментов форматирования есть «быстрые» кнопки. Но для полного представления о средствах форматирования, выделите фрагмент таблицы, щелкните правой кнопкой мыши и выберите команды **Формат ячеек**. Перед вами появится диалоговое окно, в верхней части которого расположены шесть ярлычков. Краткие справки по каждому из них:

ЧИСЛО задает, сколько десятичных знаков следует показывать для любого числа в данной ячейке, следует ли добавить знак доллара или процента, является ли число датой и т. п.

ВЫРАВНИВАНИЕ определяет, каким образом располагается содержимое внутри ячейки. Обычно числа печатают у правого края ячейки, текст – у левого, а заголовки центрируют.

ШРИФТ определяет размер, форму и толщину линий каждого символа внутри ячейки. Единицей измерения для шрифта служит «пункт»: чем больше пунктов, тем крупнее символ.

РАМКА. С помощью этого ярлычка вы можете обвести (разделить) какие-либо части таблицы линиями.

ВИД. С помощью этого ярлычка задается цвет и тонирование, что позволяет вам привлечь внимание к какой-либо части своей таблицы, например, к заголовку или итоговой строке. Не следует забывать, что на бумаге вы увидите этот цвет только если вы имеете цветной принтер.

ЗАЩИТА позволит установить блокировку, что сделает невозможным изменить содержимое данной ячейки.

Форматы чисел

Текстовые данные по умолчанию выравниваются по левому краю ячейки, а числа – по правому. Обратившись к меню **Число** диалогового меню

Формат ячеек, выберите **Категорию** из списка слева, **Код** формата из списка справа и оцените вид образца в области **Пример** в нижней части диалогового окна. Если Вас устраивает данный вид числа, щелкните на кнопке **Ок**. Форматы отображения чисел приведены в табл. 2.5.

Таблица 2.5

Некоторые форматы отображения чисел

Типы числовых данных	Формат	Вид числа на экране
Любые	Основной	-34.7 34
Числа для отчета о прибылях и убытках	Число или Бухгалтерский	34724 (34724)
Финансовые данные	Денежный	\$34724 \$365,40
Процентные соотношения	Процент	67% 3,5%
Даты и время	Дата или Время	9/11/72 09-Я.Н-1995 12:03:12

Выделение слов и чисел

Используя различный шрифт, можно сразу привлечь внимание к определенным местам вашей таблицы.

Чтобы выбрать новый шрифт для ячейки или интервала, нужно просто выделить ячейку (интервал) и выбрать шрифт из ниспадающего списка **Шрифт** на панели инструментов форматирования.

Увеличить размер букв можно с помощью ниспадающего списка **Размер шрифта**.

Чтобы украсить свой текст полужирным начертанием, курсивом или подчеркнуть его, пользуйтесь «быстрыми» кнопками на панели инструментов.

В любом случае, когда вы не представляете, как именно выглядит данный шрифт, воспользуйтесь меню **Шрифт** диалогового окна **Формат ячеек**.

Если у вас длинные строки, которые не помещаются в ячейку, можно дать указание Excel переносить текст на вторую или третью строку. Для этого вызовите диалоговое окно **Формат ячеек** выберите ярлычок **Выравнивание** и вставьте маркер в контрольный индикатор **Переносить по словам**. Теперь текст не будет исчезать из виду, достигнув края ячейки, а начнет заполнять следующую строку в той же самой ячейке.

Для заголовка нескольких столбцов (рис. 2.16) наберите текст в ячейке A1, затем выделите область столбцов которые вы хотите объединить заголовком (A1:D2), вызовите диалоговое окно **Формат ячеек**, щелкните мышью на ярлычке **Выравнивание**, выберите **Центрировать по выделению**, **Ок**.

	А	В	С
1	Цвет	Наименование	
2	черный	стол	шкаф
3	белый	стул	

Рис. 2.16. Пример заголовка

Excel позволяет изменить «ориентацию» заголовка и развернуть ячейку так, чтобы она могла служить узким заголовком столбца (по вертикали).

Как скопировать форматирование

Существуют удобные способы копирования форматов из одной ячейки в другую. Опция **Копировать формат** работает подобно кисти. «Погружаясь» в первую ячейку, кисть «вбирает» в себя все элементы форматирования, содержащиеся в этой ячейке. Затем Вы «проводите» этой кистью вдоль другой ячейки или целого интервала ячеек, и они приобретают то же самое форматирование, что и исходная ячейка. Для этого:

- 1) выберите исходную ячейку, формат которой хотите скопировать;
- 2) щелкните мышью на кнопке **Копировать формат** (если Вы собираетесь копировать формат более чем в одном месте, то щелкните на кнопке дважды, чтобы зафиксировать её). Начав перемещать указатель по рабочему листу, Вы заметите, что теперь рядом с обычным указателем Excel находится символ кисти;
- 3) щелкните мышью в ячейке, в которую хотите скопировать форматирование. Чтобы внести новый формат в целый интервал, "проведите" кистью через весь интервал, удерживая кнопку мыши нажатой;
- 4) если Вы зафиксировали кнопку **Копировать формат**, то нажмите клавишу Esc или вновь щелкните на самой кнопке, чтобы вернуть указатель к прежнему виду.

Изменение размеров строк и столбцов

Вначале в рабочем листе все строки имеют одну и ту же высоту, а все столбцы – одну и ту же ширину. Но в процессе работы эти размеры можно изменить.

Чтобы изменить размер столбца или строки, поместите указатель мыши на тонкую линию между заголовками строк или столбцов, и указатель превратится в черную черточку с двусторонней стрелкой (рис. 2.17). Нажав кнопку мыши, «передвигайте» границу строки или столбца пока не получите желаемый размер, а затем отпустите кнопку.

Чтобы автоматически отрегулировать размер строки или столбца, выделите их, затем выберите пункты меню **Формат, Строка, Подгон высоты** или **Формат, Столбец, Подгон ширины**.



Рис.2.17. Изменение размера столбца

Автоматически отрегулировать размер так же можно дважды щелкнув мышью на правой границе заголовка столбца или на нижней границе заголовка строки.

Автоформат

Команда *Автоформат* способна отформатировать целый рабочий лист. С ее помощью можно украсить даже самые невзрачные рабочие листы. Правила обращения :

1. выделите интервал, иначе Excel попытается сам угадать, какую часть своего рабочего листа Вы хотели бы автоматически отформатировать;
2. выберите пункты меню **Формат, Автоформат**, чтобы вызвать на экране диалоговое окно.

Выберите один из встроенных форматов таблицы. Просматривая список (слева), можно оцепить вид образца в окне предварительного просмотра справа.

Щелкните мышью па кнопке **Параметры**, если необходимо изменить размеры столбцов, шрифтов и т. п.

Щелкните па кнопке ОК и получайте отформатированную таблицу.

Печать документов

На стандартной панели инструментов Excel есть кнопка, нажав которую можно автоматически распечатать рабочий лист. Рабочий лист может не поместиться на странице или возникнуть другие проблемы.

Для того, чтобы получить качественную распечатку:

1. Щелкните мышью па кнопке **Просмотр печати**, и Вы увидите в точности то, что получится на распечатке.

В режиме Просмотр печати действуют режимы:

«**Увеличение**» – переключение из обычного режима просмотра на просмотр «под увеличением».

«Следующая»/ «Предыдущая» – для перехода со страницы на страницу. На индикаторе в левом нижнем углу указано сколько страниц потребуется для распечатки выбранного фрагмента.

«Поля» – позволяет увидеть на экране тонкие пунктирные линии из точек, обозначающие границы полей страницы, а также колонтитулы. С помощью мыши можно «перетаскивать» эти линии, чтобы отрегулировать поля, а также ширину столбцов.

«Закорыть» – возвращает в режим редактирования рабочего листа.

2. С помощью кнопки **Настройка** можно отрегулировать размеры полей; поместить выделенный фрагмент в нужное место на странице и вставить верхний и нижний колонтитулы, распечатать рабочий лист на определенном количестве страниц.

Работа с подсказкой

В любой из программ пакета Office очень просто получить быстрый ответ на большую часть вопросов. Стоит только открыть меню “?” и щелкнуть мышью на пункте **Вызов справки**. В вкладке **Поиск** просто начните набирать название темы или понятие с клавиатуры и, по мере того как Вы будете печатать, в ниже расположенном списке будут отображаться темы, начинающиеся со слов, которые Вы уже успели набрать. Когда вы нашли то, что искали, щелкните мышью на этом слове или словосочетании в списке, затем выберите нужный раздел и нажмите на кнопку **Вывести**.

В системе помощи пакета Office используются следующие основные элементы:

Кнопки. Они позволяют «перескочить» к другой родственной теме в программе помощи. Некоторые кнопки похожи на пиктограммы. Чтобы вернуться обратно, щелкните по кнопке с надписью **Назад**.

Подчеркнутый текст. Выделяет слово, содержащиеся в глоссарии. Щелкните мышью на слове - появится краткое определение; щелкните где-либо еще - и окно подсказки вновь исчезнет.

Создание пользовательского формата

Для создания собственного формата:

1. выделите ячейку, которую нужно отформатировать;
2. выберите в меню **Формат** команду **Ячейки** или **Формат ячеек** из контекстного меню;
3. выберите ярлычок вкладки **Число**;
4. выберите формат из списка форматов (который наиболее подходит вам);
5. в поле ввода **Код** измените код, чтобы создать новый формат.

При создании формата используйте таблицу специальных символов (табл. 2.6)

Символы для создания формата

Символ	Значение
#	Цифровой шаблон. Как Вы знаете, число состоит из целой и дробной части. Целую часть числа Excel оставляет без изменения. Дробную часть числа Excel «урезает» до количество знаков # в формате, при этом всё число округляется
0 (ноль)	Цифровой шаблон. Справедливы те же правила что и для знака #, но кроме этого, если в дробной части или в целой части числа меньше цифр, чем указано в формате, то добавляются нули до нужного количества цифр.
пробел	Разделитель тысяч (см. пример №2).
\ (слеш)	Отображает следующий за слешем символ формата.
[Синий]	В квадратных скобках задается цвет числа, если он отличается от стандартного.
«текст»	Отображает любой текст заключенный в кавычки.
_(подчерк)	Ставит один пробел.
< > =	Знаки условий. Условия заключаются в квадратные скобки (см. пример №3).

Пример №1 формат:

0,00;[Синий]-# # #0,00

Для всех неотрицательных чисел будет использоваться формат стоящий до точки с запятой, отрицательные числа будут записаны синим цветом, используя формат стоящий после точки с запятой.

Пример №2 формат:

\ руб.

При наборе числа 768900, в ячейке высветится 769. так как между значками # и \ стоит пробел, что означает отображать только тысячи.

Пример №3 формат:

[>=4]0" Дней";[>=2]0" Дня";0" День"

Выводиться различный текст в зависимости от числа. Если выполняется первое условие, то используется первый фрагмент, иначе проверяется второе условие и т.д.

Стили форматирования

Excel позволяет собирать коллекцию Ваших любимых форматов ячеек и хранить их в виде стилей, которые можно повторно использовать.

- Чтобы *создать стиль* выделите ячейку с тем **форматированием**, которое хотите сохранить, а затем выберите пункт меню «**Формат**»,

«Стиль». Наберите имя для Вашего стиля в окне **Имя стиля** и щелкните на кнопке **ОК**.

• Чтобы *выбрать* уже существующий стиль выберите «**Формат**», «**Стиль**» из ниспадающего списка имен стилей выбираете нижний и щелкаете на кнопке **ОК**.

Стили можно применять лишь в той рабочей книге, в которой Вы их создали. Для того, чтобы их использовать в другой рабочей книге Вам придется: открыть ту рабочую книгу, стили которой будут скопированы. Щелкните мышью на кнопке **Объединить** и выберите имя этой рабочей книги из появившегося списка. Щелкните на кнопке **ОК** - и стили будут доступны также и в текущей рабочей книге.

Варианты заданий

Данная лабораторная работа состоит из двух частей. Сначала выполняется первая часть заданий по вариантам, затем – вторая тоже по вариантам.

Вариант № 1

Курс \$				
Груз	Объем груза	Тариф \$	Расстояние	Стоимость перевозки
Итого				

Требуется:

1. Создать и заполнить ЭТ (не менее семи записей). Колонки «Груз», «Объем груза», «Расстояние», «Тариф» заполнить произвольно.
2. Стоимость перевозки вычисляется по формуле: $\text{Курс} * \text{Тариф} * \text{Объем} * \text{Расстояние}$.
3. Вставить графу «Заказчик» между графами «Объем груза» и «Тариф», заполнить эту графу произвольно.
4. Подсчитать **Итого** по графам «Объем груза», «Стоимость перевозки».
5. Выделить наименования граф цветом и изменить шрифт.
6. Поменять ориентацию ячейки «Груз» на вертикальную.
7. Записать ЭТ в вашу рабочую папку.

Вариант № 2

Показатель	Единица измерения	Выполнение плана		
		по плану	фактически	в %
Продукция в постоянных ценах	млн. руб.			
Среднесписочное число рабочих	чел.			
Выработка на одного рабочего	тыс. руб.			
Общее выполнение плана по объему производства				

Требуется:

1. Создать и заполнить ЭТ. Колонки «Выполнение плана по плану», «Выполнение плана фактически» заполнить произвольно.
2. Вычислить выполнение плана в %. Подсчитать общее выполнение плана по объему производства по формуле: (Среднесписочное число рабочих* выработку на одного рабочего)/100.
3. Подсчитать выработка на одного рабочего по плану и фактически.
4. Добавить показатель «Выпуск продукции, в тоннах» и заполнить её.
5. Выделить наименования граф цветом и изменить шрифт.
6. Поменять ориентацию ячейки «Единица измерения» на вертикальную.
7. Записать ЭТ в вашу рабочую папку.

Вариант № 3

Наименование изделия	Стоимость выпуска одного изделия	Количество выпущенных изделий	Продано единиц изделий	Остаток нереализованной продукции	
				в товарном выражении	в денежном выражении
Сумма					

Требуется:

1. Создать и заполнить ЭТ (не менее семи записей). Колонки «Стоимость выпуска», «Количество выпущенных изделий», «Продано единиц изделий» заполнить произвольно.

2. Подсчитать «Остаток нереализованной продукции в товарном выражении» по формуле (Количество выпущенных изделий) - (количество проданных изделий)

3. Подсчитать «Остаток нереализованной продукции в денежном выражении» по формуле: (Количество выпущенных изделий) - (количество проданных изделий)*(стоимость выпуска одного изделия)

4. Подсчитать «Сумму» по графам «Количество выпущенных изделий» и «Остаток нереализованной продукции».

5. Вставить графу «Стоимость выпуска изделий» между графами «Количество выпущенных изделий», «Продано единиц изделий». Вычислить стоимость выпуска изделий по формуле Стоимость выпуска одного изделия * Количество выпущенных изделий.

6. Выделить наименования граф цветом и изменить шрифт.

7. Поменять ориентацию ячейки «Продано единиц изделий» на вертикальную.

8. Записать ЭТ в вашу рабочую папку.

Вариант № 4

Наименование товара	Стоимость ед. товара	Принято товара, т	Списано, т	Остаток продукции на складе	
				в денежном выражении	в товарном выражении
Итого					

Требуется:

1. Создать и заполнить ЭТ (не менее семи записей). Колонки «Наименование товара», «Стоимость ед. товара», «Принято товара», «Списано» заполнить произвольно.

2. Подсчитать «Остаток продукции на складе в товарном выражении» по формуле: (количество принятого товара) - (количество списанного товара)

3. Подсчитать «Остаток продукции на складе в денежном выражении» по формуле: ((стоимость ед. товара)*(количество принятого товара))-((количество списанного товара)*(стоимость ед. товара))

4. Подсчитать «Итого» по всем графам.

5. Вставить графу «Номер склада», перед графой «Наименование товара».

6. Выделить наименования граф цветом и изменить шрифт.

7. Поменять ориентацию ячейки «Списано» на вертикальную.
8. Записать ЭТ в вашу рабочую папку.

Вариант № 5

Наименование товара	Стоимость, тыс.	Транспорт, тыс.	Цена реализации	Прибыль
ИТОГО				

Требуется:

1. Создать и заполнить ЭТ (не менее семи записей). Колонки «Наименование товара», «Стоимость», «Цена реализации».
2. Издержки на транспорт составляют 25% от стоимости товара.
3. Вставить графу «Склады» между графами «Транспорт». «Цена реализации». Стоимость аренды склада составляет 15% от стоимости товара.
4. Подсчитать «Прибыль» по формуле: (цена реализации) - (стоимость товара + затраты).
5. Подсчитать «Итого» по графам: «Стоимость». «Прибыль».
6. Выделить наименования граф цветом и изменить шрифт.
7. Поменять ориентацию всех ячеек на вертикальную.
8. Записать ЭТ в вашу рабочую папку.

Вариант № 6

Номер образца	Показания спидометра		Удельный расход топлива, на 1 км.	Количество топлива
	утром	вечером		
ИТОГО				

Требуется:

1. Создать и заполнить ЭТ (не менее семи записей). Колонки «Номер образца». «Удельный расход топлива», «Показания спидометра» заполнить произвольно.
2. Вставить графу «Километраж» между графами «Удельный расход топлива», «Количество топлива».
3. Подсчитать количество потребляемого топлива для каждого образца.

4. Подсчитать «Километраж» для каждого образца.
5. Подсчитать «Итого» по графам: «Километраж», Количество топлива».
6. Выделить наименования графы цветом и изменить шрифт.
7. Поменять ориентацию ячейки «Количество топлива» на вертикальную.
8. Записать ЭТ в вашу рабочую папку.

Вариант № 7

Вычисление №	Значения		Уравнения	
	X	Y	A	B

Требуется:

1. Создать и заполнить ЭТ (не менее семи записей).
2. Вставить графу «Z» между графами «X», «Y». Заполнить графы «Значения» произвольно.
3. Вычислить значения уравнений.
4. Вставить в графу «A» значения, вычисленные по формуле $\sin(x/y-z)$
5. Вставить в графу «B» значения, вычисленные по формуле $\sin(x-y)/\cos(z)$.
6. Подсчитать сумму значений для «X», «Y».
7. Выделить наименования граф цветом.
8. Изменить шрифт.
9. Поменять ориентацию ячейки «Значения» на вертикальную.
10. Записать ЭТ в вашу рабочую папку.

Вариант № 8

Номера предприя тий	Объем всей продукции, млн. руб., f_j	Доля товаров народного потребления, % x_j	Объем выпуска товаров народного потребления. млн. руб., z_j
Итого			
Средняя доля предметов потребления в продукции предприятий, %			

Требуется:

1. Создать и заполнить ЭТ (не менее семи записей). Колонки «Объем всей продукции», «Объем выпуска товаров народного потребления» заполнить произвольно.
2. Подсчитать долю товаров народного потребления в %.
3. Подсчитать «Итог» по графам: «Объем всей продукции», «Объем выпуска товаров народного потребления».
4. Средняя доля предметов народного потребления в продукции предприятий считается по формуле:

$$\frac{\sum_{j=1}^7 z_j}{\sum_{j=1}^7 f_j} * 100\%$$

5. Над таблицей подписать – «Объем и структура промышленной продукции».
6. Выделить наименования графы цветом и изменить шрифт.
7. Поменять ориентацию ячейки Номера предприятий на вертикальную.
8. Записать ЭТ в вашу рабочую папку.

Общие требования для всех вариантов второй части работы:

1. Открыть ЭТ, созданную Вами при выполнении первой части лабораторной работы № 5.
2. Убрать стандартную сетку.
3. Добавить (создать собственную панель инструментов) на которой должны быть пиктограммы оформления, пиктограмма стиля.
4. При создании панели используете Мастер подсказок.
5. Выделите цветной (толстой) линией одну из граф.

Вариант № 1

Объем груза
56 тонн
-65 тонн

1) Создайте собственный формат (см. пример) отображения чисел для графы «Объем груза» (причем, если число отрицательное оно должно выделяться цветом). Используя Мастер подсказок, самостоятельно найдите нужную подсказку.

2) Сохраните формат одной из ячеек графы «Объем груза» в виде стиля.

Вариант № 2

Среднесписочное число рабочих	2500 чел	450 чел
----------------------------------	-------------	------------

1) Создайте собственный формат (см. пример) отображения чисел для графы «Выполнение плана» (причем, если число больше 1000 оно должно выделяться цветом). Используя Мастер подсказок, самостоятельно найдите нужную подсказку.

2) Сохраните (формат одной из ячеек графы «Выполнение плана» и виде стиля.

Вариант № 3

Продано единиц изделий
204 шт.
50 шт.

1) Создайте собственный формат (см. пример) отображения чисел для графы «Продано единиц изделий» (причем, (если число отрицательное, оно должно выделяться цветом). Используя Мастер подсказок, самостоятельно найдите нужную подсказку.

2) Сохраните формат одной из ячеек графы «Продано единиц изделий» в виде стиля.

Вариант № 4

Списано
34 тонн
0 тонн

1) Создайте собственный формат (см. пример) отображения чисел для графы «Списано» (причем, если число больше 200, оно должно выделяться синим цветом). Используя Мастер подсказок, самостоятельно найдите нужную подсказку.

2) Сохраните формат одной из ячеек графы «Списано» и виде стиля.

Вариант № 5

Прибыль
56 тыс.
4 тыс.

1) Создайте собственный формат (см. пример) отображения чисел для графы «Прибыль» (причем, если число меньше 50 тыс., оно должно выделяться красным цветом). Используя Мастер подсказок, самостоятельно найдите нужную подсказку.

2) Сохраните формат одной из ячеек графы «Прибыль» и виде стиля.

Вариант № 6

Показания спидометра	
Утром	Вечером
17,00 тыс. км.	17,87 тыс. км.

- | | |
|---------------|---------------|
| 2,00 тыс. км. | 3,43 тыс. км. |
|---------------|---------------|
- 1) Создайте собственный формат (см. пример) отображения чисел для графы «Показания спидометра» (причем, если число больше 20 тыс., оно должно выделяться красным цветом). Используя Мастер подсказок, самостоятельно найдите нужную подсказку.
- 2) Сохраните формат одной из ячеек графы «Показания спидометра» в виде стиля.

Вариант № 7

Вычисление №
004
023

- 1) Создайте; собственный формат (см. пример) отображения чисел для графы «Вычисление». Используя Мастер подсказок, самостоятельно найдите нужную подсказку.
- 2) Сохраните формат одной из ячеек графы «Вычисление» в виде стиля.

Вариант № 8

Объем всей продукции млн. руб., f_i
34 млн. руб.
4 млн. руб.

- 1) Создайте собственный формат (см. пример) отображения чисел для графы «Объем всей продукции». Используя Мастер подсказок, самостоятельно найдите нужную подсказку.
- 2) Сохраните формат одной из ячеек графы «Объем всей продукции» в виде стиля.

Лабораторная работа № 6 ТАБЛИЧНЫЙ ПРОЦЕССОР MICROSOFT EXCEL

Цель работы: научиться использовать простые формулы, составлять простые условия, при работе использовать табличные формулы, составлять сложные условия.

Теоретические сведения

Общие сведения об использовании функций

Вы можете использовать встроенные функции Microsoft Excel для выполнения стандартных вычислений. Значения, которые используются для вычисления функций, называются *аргументами*. Значения, которые функции возвращают в качестве ответа, называются *результатами*.

Если функция появляется в самом начале формулы, ей должен предшествовать знак равенства, как и во всякой другой формуле. Последовательность, в которой должны располагаться используемые в формуле символы, называется *синтаксисом функции*. Все функции используют одинаковые основные правила синтаксиса.

Скобки позволяют определить, где начинается и где заканчивается список аргументов. Помните о том, что нужны обе скобки, и при этом не следует вставлять пробелов ни перед ними, ни после них. В качестве аргументов можно использовать числа, текст, логические значения, массивы, значения ошибок или ссылки. Задаваемые аргументы должны иметь допустимые для данного аргумента значения.

Некоторые функции могут иметь необязательные аргументы, которые могут отсутствовать при вычислении значений функции.

Аргументы могут быть как константами, так и формулами. В свою очередь эти формулы могут содержать другие функции. Функции, являющиеся аргументами других функций, называются *вложенными функциями*. В формулах можно использовать до семи уровней вложенности функций. Мастер Функций позволяет автоматизировать процесс создания формулы.

1. чтобы вставить функцию и самом начале формулы, выделите ячейку, в которой Вы хотите набрать формулу. Чтобы вставить функцию в другом месте формулы, установите курсор там, где нужно вставить функцию;

2. в меню **Вставить** выберите команду **Функция**; Быстрый вызов: Кнопка Мастер Функций или Shift + F3. Microsoft Excel отображает диалоговое окно Мастера Функций, в котором выводит список типов или логических групп функций и для каждого элемента этого списка выводится список соответствующих функций, включая пользовательские;

3. выберите категорию функции;

4. выберите имя функции. Если требуется более полная информация о функции и ее использовании нажмите F1;

5. нажмите кнопку Шаг; Появится диалоговое окно следующего шага Мастера Функций. В этом окне выводятся названия аргументов в функции рядом с полями ввода, используемыми для заполнения аргументов;

6. введите значения аргументов; Между именем аргумента и его полем ввода расположена кнопка Мастера Функций. Щелчок по этой кнопке приведет к еще одному вызову Мастера Функций, что позволяет ввести вложенные функции в качестве аргумента;

7. нажмите на кнопку **Закончить**. Мастер Функций помещает законченную функцию в формулу.

Для получения результата вычисления формулы нажмите Enter.

Математические и тригонометрические функции

Пример использования некоторых функций.

СУММЕСЛИ (интервал; критерий; интервал_суммирования)

Суммирует ячейки удовлетворяющие критерию.

Интервал – это интервал ячеек, в котором производится поиск ячеек удовлетворяющих заданному критерию.

Критерий – это условие, которое может задаваться в форме числа, выражения или текста. Например, критерий может быть выражен как 32, "32", ">32", "яблоки".

Интервал суммирования – это фактические ячейки для суммирования. Ячейки из интервала (интервал_суммирования) суммируются, только если соответствующие им ячейки в аргументе интервал удовлетворяют критерию. Если интервал_суммирования опущен, то суммируются ячейки из аргумента Интервал.

Таблица 2.7

Пример таблицы вида

	А	В
1	<i>стоимость</i>	<i>комиссионный сбор</i>
2	100	7 руб.
3	200	14 руб.
4	300	21 руб.
5	400	28 руб.
6	150	128 руб.
7	90	280 руб.

Данные для вычисления по формуле СУММЕСЛИ() (табл. 2.7).
Запишем формулу в ячейку В6 :

=СУММЕСЛИ(А2:А7;">160";В2:В7)

Получим 63 руб. – сумма комиссионных сборов с товаров чья стоимость больше 160.

СУММ (число1;число2;...)

Суммирует аргументы.

Число 1, Число2,... – это от 1 до 30 аргументов, которые суммируются.

- При суммировании учитываются числа, логические значения и текстовые представления чисел. Аргументы, которые являются текстами не преобразуемыми в числа, вызывают ошибки. См. первый пример.

- Если аргумент является массивом или ссылкой, то учитываются только числа. Пустые ячейки, логические значения, тексты и значения ошибок в массиве или ссылке игнорируются. См. второй пример.

Пример №1:

=СУММ(3;2;"5";ИСТИНА)

равняется 11. так как текстовые значения преобразуются в числа, а логическое значение ИСТИНА преобразуется в число 1.

Пример №2: Имеется таблица вида (табл. 2.8)

Таблица 2.8

Пример таблицы вида

	А	В
1	"5"	ИСТИНА
2	10	20
3	25	
4	30	15
5	"8"	12

=СУММ(А1:В2;2) равняется 22, так как не числовые значения в ссылке не преобразуются.

=СУММ(А1:А3) равно 35

=СУММ(А1:В3) равно 55

=СУММ(А1:А3;2) равно 37

=СУММ(А1:А5;2) равно 67

=СУММ(А1:А5) равно 65

ПРОИЗВЕД(число1;число2;...)

Перемножает числа, заданные в качестве аргументов, и возвращает их произведение.

Число 1, Число 2,... – это от 1 до 30 перемножаемых аргументов.

•Учитываются числа, логические значения и текстовые представления чисел. Аргументы, которые являются текстами не преобразуемыми в числа, вызывают ошибки.

•Если аргумент является массивом или ссылкой, то учитываются только числа. Пустые ячейки, логические значения, тексты и значения ошибок в массиве или ссылке игнорируются.

Таблица 2.9

Пример таблицы вида

	А	В	С
1	3	15	30

Данные для вычисления по формуле **ПРОИЗВЕД ()** (табл. 2.9).

=ПРОИЗВЕД(А1:С1) равняется 1350

=ПРОИЗВЕД(А1:С1;2) равняется 2700

СЧЕТЕСЛИ(интервал; критерий)

Подсчитывает количество не пустых ячеек, удовлетворяющих заданному критерию внутри интервала.

Описание аргументов аналогично функции СУММЕСЛИ.

Таблица 2.10

Информация о поступающих в вуз

	А	В	С
1	Ф.И.О.	Школа №	Ср. оценка аттестата
2	Иванов И. И.	1	3,8
3	Федоров П.Е.	20	5
4	Персик И.Н.	10	4,6
5	Семенова Е.Н.	20	4

Подсчитаем количество выпускников школы №20 (табл. 2.10).

=СЧЕТЕСЛИ(А2:В5;20) равняется 2

Подсчитаем количество выпускников с средним баллом больше 4.

=СЧЕТЕСЛИ(С2:С5;">4") равняется 2

Подсчитаем количество выпускников с средним баллом 3.

=СЧЕТЕСЛИ(С2:С5;"=3") равняется 0

СЛЧИС()

Возвращает случайное число между 0 и 1. Новое случайное число возвращается каждый раз, когда рабочий лист вычисляется.

Замечания:

- Чтобы получить случайное вещественное число между А и В, можно использовать следующую формулу: =СЛЧИС()*(b-a)+a

- Если Вы хотите использовать функцию СЛЧИС для генерации случайного числа, но не хотите, чтобы это число менялось каждый раз при вычислении значения ячейки, то Вы можете ввести =СЛЧИС () в строку формул и нажать F9 для фиксации значения случайного числа.

Пример:

=СЛЧИС()*100

Для генерации случайного числа в пределах от 0 до 100.

ОКРУГЛ(число; число разрядов)

Округляет число до указанного количества десятичных разрядов.

Число - это округляемое число или аргумент

Число_разрядов - это количество десятичных разрядов, до которого нужно округлить число.

Пример:

=ОКРУГЛ(2,15;1) равняется 2.2

=ОКРУГЛ(2.15;0) равняется 2

=ОКРУГЛ(B1;0) равняется 2, если в ячейке B1 число 2,15

ЦЕЛОЕ(число)

Округляет число до ближайшего меньшего целого.

Число - это округляемый аргумент.

Пример:

=ЦЕЛОЕ(2,78) равняется 2

Статистические функции

МАКС(число1;число2;...)

МИН(число1;число2;...)

Возвращает максимальное (минимальное) значение из списка аргументов.

Число1, Число2,... – это от 1 до 30 аргументов, среди которых ищется максимальное (минимальное) значение.

- Аргументами могут быть числа, логические значения и текстовые представления чисел. Аргументы, которые являются текстами не преобразуемыми в числа, вызывают ошибки.

- Если аргумент является массивом или ссылкой, то учитываются только числа. Пустые ячейки, логические значения, тексты и значения ошибок в массиве или ссылке игнорируются.

- Если аргументы не содержат чисел, то функции МАКС (МИН) возвращает ноль.

Таблица 2.11

Пример таблицы вида

	А	В
1	"5"	ИСТИНА
2	10	2
3	25	

=МИН(А1:А3) результат 10

=МИН(А1:В3) результат 2

=МИН(ИСТИНА;4;"3") результат 1

=МАКС(А1:В3;100) результат 100

Логические функции

ЕСЛИ(лог-ое_выраж-;знач_истина;знач_ложь)

Выполняет логическую проверку.

Лог_выражение – это любое значение или выражение, которое при вычислении дает значение ИСТИНА или ЛОЖЬ.

Знач_если_истина – это значение, которое возвращается, если лог_выражение имеет значение ИСТИНА,

Знач_если_ложь – это значение, которое возвращается, если лог_выражение имеет значение ЛОЖЬ.

Пример №1: Рассмотрим табл. 2.10

Подсчитаем количество выпускников школы №20. Запишем формулу =ЕСЛИ(В2=20;1;0) и ячейку D2, затем скопируйте формулу на интервал D3:D5, (напротив каждой записи будут стоять 0 - не выпускник школы №20 или 1 - выпускник школы №20) и в ячейке D6 запишите формулу =СУММ(D2:D5). Ответом служит число 2, стоящее в ячейке D6.

Пример №2: Рассмотрим табл. 2.10

Подсчитаем количество выпускников школы №20, имеющих средний балл аттестата больше 4. Запишем формулу

=ЕСЛИ(В2=20;ЕСЛИ(С2>4;1;0);0) в ячейку E2, затем скопируйте формулу на интервал E3:E5, (напротив каждой записи будут стоять 0 - не выпускник школы №20 или 1 - выпускник школы №20) и в ячейке D6 запишите формулу =СУММ(E2:E5). Ответом служит число 1, стоящее в ячейке E6.

Замечания: Если Вам нужно, чтобы условие проверялось для нескольких ячеек, то чтобы не добавлять дополнительный столбец условий воспользуйтесь, табличной формулой.

И(логическое1;логическое2;...)

ИЛИ (логическое1 ;логическое2;...)

Возвращает значение ИСТИНА, если все аргументы имеют значение ИСТИНА, во всех других случаях возвращает значение ЛОЖЬ.

Логическое1, логическое2,.. – это 1 до 30 проверяемых условий, которые могут иметь значение либо ИСТИНА, либо ЛОЖЬ.

•Аргументы должны быть логическими значениями, массивами или ссылками которые содержат логические значения. Если указанный интервал не содержит логических значений, то И возвращает значение ошибки #ЗНАЧ!

•Если аргумент, который является ссылкой или массивом, содержит тексты или пустые ячейки, то такие значения игнорируются.

Пример №1.

=И(ИСТИНА;ИСТИНА) равняется ИСТИНА

=И(ИСТИНА;ЛОЖЬ) равняется ЛОЖЬ

Если интервал В1:В3 содержит значения ИСТИНА, ЛОЖЬ, ИСТИНА, то =И(В1:В3) равняется ЛОЖЬ. Если в ячейке В4 содержится число в пределах от 50 до 70, то =И(В4>50;В3<70) результат ИСТИНА.

Проверим, если в ячейке А2 записано "яблоки", а в ячейке В2 их стоимость меньше 6, выведем их стоимость в ячейке С2, иначе "таких яблок нет". Запишем формулу в ячейке С2 :

=ЕСЛИ(И(А2="яблоки";В2<6);В2;"таких яблок нет")

Пример №2: Запишем формулу в ячейку С3:

=ЕСЛИ(ИЛИ(А2="яблоки";А2="груши");В2;"это другой продукт")

Если в ячейке А2 записано "яблоки" или "груши", то в С3 будет выведена цена из ячейки В2, иначе фраза "это другой продукт".

НЕ(логическое_значение)

Меняет на противоположное логическое значение своего аргумента.

Логическое_значение – это значение или выражение, которое при вычислении дает ИСТИНА или ЛОЖЬ.

Пример:

=НЕ(ЛОЖЬ) равняется ИСТИНА

Ссылки

ВПР(искомое_значение;таблица_массив;номер_столбца;диапазон_просмотра)

Искомое_значение – значение которое должно быть найдено в первом столбце_таблицы массива. Аргументами могут быть данные: текст, числа, логические значения.

Таблица_массив – область рабочего листа с информацией для поиска и вывода результата.

Номер_столбца – номер столбца из которого берется значение, относительно столбца в котором производится поиск,

Диапазон_просмотра – (данный параметр необязательный, по умолчанию принимает значение ИСТИНА) признак, определяющий точность сопоставления искомого значения с значениями в первом столбце таблицы массива. Параметр имеет два значения: ИСТИНА – сопоставление приближенное; ЛОЖЬ – сопоставление точное.

Пример.№1: Рассмотрим табл. 2.15

Найдем номер школы, которую закончил ученик по фамилии Персик И.В.
=ВПР("Персик И.Н.";A2:B5;2;ложь)

Ищем фамилию Персик И.Н. в первом столбце интервала A2:B5, результат берем из второго столбца интервала A2:B5. Сопоставление фамилий точное.

Табличные формулы

Пример №1: Рассмотрим табл. 2.10

Подсчитаем количество выпускников школы №20
{=СУММ(ЕСЛИ(B2:B5=20;1))}

При наборе табличной формулы руководствуйтесь правилами:

1. наберите формулу без фигурных скобок и не спешите нажимать Enter;
2. нажмите комбинацию клавиш <Ctrl> + <Shift> + <Enter>. Вы должны удерживать нажатыми клавиши <Ctrl> и <Shift> в тот момент когда нажимаете <Enter>, Вы получите фигурные скобки иначе формула будет введена как обычная формула.

Пример №2: Рассмотрим табл. 2.10

Подсчитаем количество выпускников школы №20, имеющих среднюю оценку аттестата >4 баллов.
{=СУММ(ЕСЛИ(B2:B5=20;ЕСЛИ(C2:C5>4;1;0);0))}

Пример №3: Имеется таблица вида:

Таблица 2.12

Пример таблицы вида

1	2	3
---	---	---

№ Отделения банка	Валюта	Объем продажи
1	\$	100
1	DM	300
2	\$	30
2	DM	0
3	DM	120

Подсчитаем объем продаж \$ по всем отделениям банков.
 {=СУММ(ЕСЛИ(B3:B7="\$";C3:C7;0))}

Выражение A3:A7="\$" имеет значение ИСТИНА для каждого вхождения символа \$ в указанном интервале. При значении ИСТИНА складываются значения ячеек из интервала B3:B7. Результат функции равен 130.

Подсчитаем объем продаж DM по всем отделениям банков.
 {=СУММ(ЕСЛИ(B3:B7=" DM ";C3:C7;0))}

Выражение A3:A7="DM" имеет значение ИСТИНА для каждого вхождения символа \$ в указанном интервале. При значении ИСТИНА складываются значения ячеек из интервала B3:B7. Результат функции равен 420.

Варианты заданий

Вариант № 1

	A	B	C	D
1	Отчетная ведомость			
2	Изделие	Стоимость (тыс.)	Цвет	Кол-во
3	<i>стол</i>		<i>черный</i>	
4	<i>стул</i>		<i>белый</i>	
...				
20	<i>шкаф</i>		<i>белый</i>	

Требуется: Создать и заполнить ЭТ (не менее 20 записей), колонки A, C заполнить произвольно.

1. Стоимость одного изделия вычислить по формуле: случайное число из диапазона 50..700.

2. Количество изделий вычислить по формуле: (стоимость изделия) - (случайное число в пределах от 1 до стоимости изделия).
3. Двумя различными способами подсчитать, сколько видов изделий «стол» выпущено.
4. Двумя различными способами подсчитать количество черных столов.
5. В столбце «Стоимость» используйте денежный формат.
6. Записать ЭТ в ваш рабочий каталог.

Вариант № 2

	А	В	С	Д	Е	Ф
1	СПРАВОЧНИК					
2	<i>Площадь.</i>	<i>Адрес</i>				<i>Стоимость квартиры</i>
3	<i>кв.м</i>	<i>улица</i>	<i>дом</i>	<i>№ кв.</i>	<i>этаж</i>	
4	40	Мира	32		2	
...						
20	85	Ленина	20		4	

Требуется: Создать и заполнить ЭТ (не менее 20 записей), колонки А, В, С, Е заполнить произвольно. Номер квартиры вычисляется по формуле: случайное число из диапазона 1..150.

1. Стоимость квартиры вычисляется по формуле: ((площадь квартиры)*(случайное число из диапазона 1..7))-(этаж)*(случайное число из диапазона 1..5)).

2. Двумя различными способами подсчитать сколько квартир с номером 33 на улице Мира и площадью >50 кв.м.

3. Двумя различными способами подсчитать общую площадь квартир на улицах Ленина и Мира.

4. В столбце «Стоимость квартиры» используйте денежный формат «млн.»

5. Записать ЭТ в ваш рабочий каталог.

Вариант № 3

	А	В	С	Д	Е
1	СПРАВОЧНИК				
2	мин. з/п	346			
3	<i>Ф.И.О.</i>	<i>День недели</i>	<i>Смена, 1/2</i>	<i>Кэф. 1/1.5</i>	<i>Зарботная плата</i>
4	Иванов И. Н.	Суббота	2		
...					
20	Петров М. Л.	четверг	1		

Требуется: Создать и заполнить ЭТ (не менее 20 записей), колонки А, В, С заполнить произвольно.

1. Расставить коэффициент в столбце D по правилу: человек, работающий в первую смену имеет коэффициент 1; человек, работающий во вторую смену имеет коэффициент 1,5.

2. Двумя различными способами подсчитать сколько человек работает по субботам во вторую смену.

3. Двумя различными способами подсчитать общую заработную плату всех работающих по субботам и пятницам во вторую смену.

4. В столбце «Заработная плата» используйте денежный формат «тыс».

5. Записать ЭТ в ваш рабочий каталог.

Вариант № 4

	А	В	С	Д	Е
1	Наличие товара на складе				
2	<i>Номер склада</i>	<i>Количество товара</i>	<i>Цена ед. товара</i>	<i>Стоимость товара, тыс.</i>	<i>Площадь склада</i>
3	34		7		500
...					
20	1		20		250

Требуется: Создать и заполнить ЭТ (не менее 20 записей), колонки А, С, Е заполнить произвольно. Количество товара вычисляется по формуле: случайное число из диапазона 100..250.

1. Стоимость товара вычислять по формуле: Количество товара * Цену ед. товара.

2. Двумя различными способами подсчитать па какую сумму товара на первом складе.

3. Двумя различными способами подсчитать количество товара, стоимость которого превышает 500 тыс.

4. В столбце «Количество товара» используйте формат «тонн».

5. Записать ЭТ в ваш рабочий каталог.

Вариант № 5

	А	В	С	Д	Е
1	Физические показатели группы.				
2	<i>Ф. И. О.</i>	<i>Группа</i>	<i>Пол, м/ж</i>	<i>Рост, см</i>	<i>Вес, кг.</i>
3	Иванов В. В.	21-1	м	190	103
...					
20	Сидоров А. П.	73-2	м	174	67

Требуется: Создать и заполнить ЭТ (не менее 20 записей), колонки А, В, С заполнить произвольно. Колонку "Рост" заполнить по формуле: случайное число из диапазона 150..250.

1. Колонку «Вес» заполнить по формуле: случайное число из диапазона 45..150.
2. Двумя различными способами подсчитать средний рост мужчин в группе X.
3. Двумя различными способами подсчитать процентное соотношение мужчин и женщин в группе X.
4. В столбце «Вес» используйте формат «кг».
5. Записать ЭТ в ваш рабочий каталог.

Вариант № 6

	А	В	С	Д	Е
1	Расписание				
2	<i>Номер рейса</i>	<i>Пункт отправления</i>	<i>Пункт назначения</i>	<i>Время</i>	
3				6.30	10.10
...					
20	555	Красноярск	Минск	5.20	7.54

Требуется: Создать и заполнить ЭТ (не менее 20 записей), колонки А, В, С заполнить произвольно.

1. Время вылета и время прилета вычисляется по формуле: случайное число из диапазона 0.00...24.00
2. Двумя различными способами подсчитать сколько утренних рейсов (4.00-11.00) вылетает в Москву.
3. Двумя различными способами подсчитать количество рейсов вылетающих в Норильск.
4. В столбце «Время» используйте формат «час».
5. Записать ЭТ в ваш рабочий каталог.

Вариант № 7

	А	В	С	Д
1	СПРАВОЧНИК			
2	<i>Автор</i>	<i>Название книги</i>	<i>Год издания</i>	<i>Тираж'</i>
3	Симонович	Всё об internet	2000	20000
...				
...				
20	Том Сван	Windows форматы файлов	1999	26000

Требуется: Создать и заполнить ЭТ (не менее 20 записей), колонки А, В, D заполнить произвольно.

1. Год издания вычисляется по формуле: случайное число из диапазона 1800...2005.
2. Двумя различными способами подсчитать сколько раз издавалась книга Y автора X.
3. Двумя различными способами подсчитать каков общий тираж книги Y автора X.
4. В столбце «Год издания» используйте формат «год».
5. Записать ЭТ в ваш рабочий каталог.

Вариант № 8

	А	В	С	Д
1	СТАТИСТИКА			
2	<i>Город</i>	<i>Месяц</i>	<i>Ср. температура</i>	<i>Кол. осадков</i>
3	Москва	Январь	-15.4	24,1
...				
20	Киев	Март	10,2	30

Требуется: Создать и заполнить ЭТ (не менее 20 записей), колонки А, В, С заполнить произвольно.

1. Количество осадков вычисляется по формуле: случайное число из диапазона 10..50.
2. Двумя различными способами подсчитать среднюю температуру в городе X за два месяца Январь. Февраль.
3. Двумя различными способами подсчитать среднее количество осадков в городе Y.
4. В столбце «Ср. температура» используйте формат «⁰С».
5. Записать ЭТ в ваш рабочий каталог.

Лабораторная работа № 7 СУБД MICROSOFT ACCESS СОЗДАНИЕ БАЗОВЫХ ТАБЛИЦ И МЕЖТАБЛИЧНЫХ СВЯЗЕЙ

Цель первой части работы: научиться создавать таблицы базы данных, задавать их структуру, выбирать типы полей и управлять их свойствами. Освоить приемы наполнения таблиц конкретным содержанием.

Руководитель малого предприятия, выполняющего сборку персональных компьютеров из готовых компонентов, заказал разработку базы данных, основанной на двух таблицах комплектующих. Одна таблица содержит данные, которые могут отображаться для клиентов при согласовании спецификации изделия, – в ней указаны розничные цены на компоненты. Вторая таблица предназначена для анализа результатов деятельности предприятия – в ней содержатся оптовые цены на компоненты и краткая информация о поставщиках (клиенты предприятия не имеют доступа к данным этой таблицы).

Порядок выполнения первой части работы

1. Запустите программу Microsoft Access (Пуск > Программы > Microsoft Access).
2. В окне Microsoft Access включите переключатель **Новая база данных** и щелкните на кнопке **Ок**.
3. В окне **Файл** новой базы данных выберите папку **Мои документы** и дайте файлу имя: **Комплектующие**. Убедитесь, что в качестве типа файла выбраны **Базы данных Microsoft Access**, и щелкните на кнопке **Создать**. Откроется окно новой базы **Комплектующие: база данных**.
4. Откройте панель **Таблицы**.
5. Дважды щелкните на значке **Создание таблицы в режиме конструктора** – откроется бланк создания структуры таблицы.
6. Для первой таблицы введите следующие поля (рис. 2.18).

Имя	Тип
Компоне	Текс
Модель	Текс
Основно	Числ
Цена	Числовой

Рис.2.18. Пример полей

Обратите внимание на то, что в данном случае поле Цена задано не денежным типом, а числовым. Данные, относящиеся к денежному типу, имеют размерность, выраженную в рублях (если работа выполняется с версией Microsoft Access, локализованной в России). Но стоимость компонентов вычислительной техники выразить в этой единице измерения не принято. Для сравнимости цен разных поставщиков обычно используют «условные единицы». В таких случаях удобно использовать поле числового типа, чтобы не перенастраивать всю СУБД.

7. Щелкните на поле Цена. В нижней части бланка задайте свойство Число десятичных знаков, равным 2.

8. Для связи с будущей таблицей поставщиков надо задать ключевое поле. Поскольку здесь ни одно поле явно не претендует на “уникальность”, используем комбинацию полей **Компонент** и **Модель**. Выделите оба поля в верхней части бланка (при нажатой клавише SHIFT). Щелчком правой кнопки мыши откройте контекстное меню и выберите в нем пункт **Ключевое поле**. Закройте окно **Конструктора**. При закрытии окна дайте таблице имя Комплектующие. Повторив действия пунктов 5 – 9, создайте таблицу Поставщики, в которую входят следующие поля (рис. 2.19).

Имя поля	Тип поля
Компонент	Текстовый
Модель	Текстовый
Цена оптовая	Числовой
Поставщик	Текстовый
Телефон	Текстовый
Адрес	Текстовый
Примечание	Поле МЕМО

Рис.2.19. Таблица поставщики

Обратите внимание на то, что поле номера телефона является текстовым, несмотря на то, что обычно номера телефонов записывают цифрами. Это связано с тем, что они не имеют числового содержания. Номера телефонов не сравнивают по величине, не вычитают из одного номера другой и т. д. Это типичное текстовое поле.

Ключевое поле можно не задавать – для текущей задачи оно не требуется.

9. В окне **Комплектующие: база данных** откройте по очереди созданные таблицы и наполните их экспериментальным содержанием (3 – 4 записи). Закончив работу, закройте таблицы и завершите работу с программой.

Цель второй части работы: изучить условия, необходимые для создания взаимосвязанных таблиц, и приемы их создания. Познакомиться с основными типами связей, образующихся между таблицами, и научиться редактировать параметры связи.

Порядок выполнения второй части работы

1. Запустите программу Microsoft Access 2000 (Пуск > Программы > Microsoft Access).

2. В окне Microsoft Access включите переключатель **Открыть базу данных**, выберите ранее созданную базу **Комплектующие** и щелкните на кнопке **Ок**.

3. В окне **Комплектующие: база данных** откройте панель **Таблицы**. Убедитесь, что на ней присутствуют значки ранее созданных таблиц **Комплектующие** и **Поставщики**.

4. Разыщите на панели инструментов кнопку **Схема данных**. Если есть сложности, найдите команду строки меню: **Сервис > Схема данных**. Воспользуйтесь любым из этих средств, чтобы открыть окно **Схема данных**. Одновременно с открытием этого окна открывается диалоговое окно **Добавление таблицы**, на вкладке **Таблицы** которого можно выбрать таблицы, между которыми создаются связи.

5. Щелчком на кнопке **Добавить** выберите таблицы **Комплектующие** и **Поставщики** в окне **Схема данных** откроются списки полей этих таблиц.

6. При нажатой клавише SHIFT выделите в таблице **Комплектующие** поля **Компонент** и **Модель**.

7. Перетащите эти поля на список полей таблицы **Поставщики**. При отпускании кнопки мыши автоматически откроется диалоговое окно **Изменение связей**.

8. На правой панели окна **Изменение связей** выберите поля **Компонент** и **Модель** таблицы **Поставщики**, включаемые в связь. Не устанавливайте флажок **Обеспечение целостности данных** — в данном упражнении это не требуется, но может препятствовать постановке учебных опытов с таблицами.

9. Закройте диалоговое окно **Изменение связей** и в окне **Схема данных** рассмотрите образовавшуюся связь. Убедитесь, что линию связи можно выделить щелчком левой кнопки мыши, а щелчком правой откроется контекстное меню, позволяющее разорвать связь или отредактировать ее.

10. Закройте окно **Схема данных**.

Лабораторная работа № 8
СУБД MICROSOFT ACCESS
СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОРКУ
СОЗДАНИЕ ЗАПРОСОВ «С ПАРАМЕТРОМ»

Цель первой части работы: научиться создавать запрос на выборку, основанный на связанных таблицах, формировать структуру запроса путем выбора базовых таблиц и полей, а также формулировать условие отбора.

В первой части работы мы создадим запрос на выборку жестких дисков, имеющих емкость не менее 8 Гбайт при цене менее 150 условных единиц. Результирующая таблица должна содержать также адрес поставщика и номер его телефона.

Порядок выполнения первой части работы

1. Запустите программу Microsoft Access (Пуск > Программы > Microsoft Access).
2. В окне Microsoft Access включите переключатель **Открыть базу данных**, выберите ранее созданную базу **Комплектующие** и щелкните на кнопке **Ок**.
3. В окне **Комплектующие: база данных** откройте панель **Запросы**. Дважды щелкните на значке **Создание запроса** в режиме **Конструктора** — откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно **Добавление таблицы**.
4. В окне **Добавление таблицы** выберите таблицу **Поставщики** и щелкните на кнопке **Добавить**. Закройте окно **Добавление таблицы**.
5. В списке полей таблицы **Поставщики** выберите поля, включаемые в результирующую таблицу: **Компонент**, **Модель**, **Цена оптовая**, **Поставщик**, **Телефон**. Выбор производите двойными щелчками на именах полей.
6. Задайте условие отбора для поля **Компонент**. В соответствующую строку введите: **Жесткий диск**. Из таблицы будут выбираться не все изделия, а только жесткие диски.
7. Задайте условие отбора для поля **Цена оптовая**. В соответствующую строку введите: **< 150**. Из таблицы будут выбираться только изделия, имеющие цену менее 150 условных единиц.
8. Нам надо задать условие отбора по основному потребительскому параметру — емкости жесткого диска. Однако в таблице **Поставщики** такого поля нет, но в ней есть поле **Модель**, которое однозначно определяет параметры изделия. Благодаря тому, что по полю **Модель** у нас установлена связь с таблицей **Комплектующие**, мы получаем возможность ввести в запрос поле **Основной параметр**, взяв его из другой таблицы. Добавьте список полей

таблицы Комплектующие в верхнюю часть бланка запроса по образцу. Для этого щелкните правой кнопкой мыши в верхней области бланка и в открывшемся контекстном меню выберите пункт **Добавить таблицу** — откроется уже знакомое нам окно **Добавление таблицы**. Выберите в нем таблицу **Комплектующие**.

9. Двойным щелчком на поле **Основной параметр** в списке полей таблицы **Комплектующие** введите это поле в бланк запроса по образцу.

10. В строке **Условие отбора столбца Основной параметр** введите условие **>8** (емкость диска более восьми гигабайт).

11. Закройте бланк запроса по образцу. При закрытии запроса введите его имя — **Выбор комплектующих**. В окне **Комплектующие: база данных** откройте только что созданный запрос и рассмотрите результирующую таблицу. Ее содержательность зависит от того, что было введено в таблицы **Комплектующие** и **Поставщики** при их наполнении в лабораторной работе № 6. Если ни одно изделие не соответствует условию отбора и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы и наполните их модельными данными, позволяющими проверить работу запроса.

Порядок выполнения второй части работы

Цель второй части работы: научиться формировать запросы «с параметром», в основе которых лежат запросы на выборку, у которых в поле **Условие отбора** записан заключенный в квадратные скобки текст, обращенный к пользователю.

Специальный тип запросов, называемый запросами «с параметром», позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этим приемом обеспечивается гибкость работы с базой.

Создадим простой запрос, позволяющий отбирать процессоры, предельную цену которых пользователь может задать сам при запуске запроса.

1. В окне **Microsoft Access** включите переключатель **Открыть базу данных**, выберите ранее созданную базу **Комплектующие** и щелкните на кнопке **Ок**.

2. В окне **Комплектующие: база данных** откройте панель **Запросы**. Дважды щелкните на значке **Создание запроса в режиме Конструктор** — откроется бланк запроса по образцу.

3. Согласно лабораторной работе №3 создайте запрос на выборку, основанный на таблице **Поставщики**, в который войдут следующие поля:

Компонент;

Модель;

Цена оптовая;

Поставщик;

Телефон.

4. В строке Условие отбора поля Компонент введите: Процессор.

5. Строку Условие отбора для поля Цена оптовая надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение. Текст, обращенный к пользователю, должен быть заключен в квадратные скобки. Если бы мы хотели отобрать процессоры, цена которых больше 100 единиц, мы бы написали: >100. Если бы нам были нужны процессоры дешевле 80 единиц, мы бы написали <80. Но если мы хотим дать пользователю возможность выбора, мы должны написать: < [Введите максимальную цену]. Закройте запрос. При закрытии сохраните его под именем Выбор комплектующих.

6. В окне **Комплектующие: база данных** откройте панель **Запросы** и запустите запрос **Выбор комплектующих** — на экране появится диалоговое окно **Введите значение параметра**.

7. Введите какое-либо число и щелкните на кнопке **Ок**. В зависимости от того, что реально содержится в таблице Поставщики, по результатам запроса будет сформирована результирующая таблица.

8. Закройте все объекты базы данных и программу Microsoft Access.

ПРОГРАММИРОВАНИЕ В СРЕДЕ DELPHI

Лабораторная работа № 9 ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ DELPHI

Цель работы: изучить основы среды разработчика, объекты их свойства и назначение, а также команды главного меню.

Теоретические сведения

Базовые программные средства относятся к инструментальной страте информационных технологий и включают в себя:

- операционные системы (ОС);
- языки программирования;
- программные среды;
- системы управления базами данных (СУБД).

Программные среды реализуют отдельные задачи и операции информационных технологий. Среда DELPHI является одним из средств разработки информационных систем.

Понятие объекта. Реальные объекты окружающего мира обладают тремя базовыми характеристиками: они имеют набор свойств, способны разными методами изменять эти свойства и реагировать на события, возникающие как в окружающем мире, так и внутри самого объекта. Именно в таком виде в языках программирования и реализовано понятие объекта как совокупности свойств, методов их обработки и событий, на которые данный объект может реагировать и которые приводят к изменению свойств объекта.

Класс. Объекты могут иметь идентичную структуру и отличаться только значениями свойств. В таких случаях в программе создается новый тип, основанный на единой структуре объекта. Он называется классом, а каждый конкретный объект, имеющий структуру этого класса, называется экземпляром класса.

Описание класса. Описание нового класса похоже на описание новой структуры данных, только к полям (свойствам) добавляются методы – программы.

```
class TMyClass
Item1: integer;
Item2: string;
function GetSum(n:integer): integer;
procedure initialize;
end;
```

При определении подпрограмм, принадлежащих конкретному классу, его методов в заголовке подпрограммы, перед ее названием, указывают, к какому классу она принадлежит. Название класса от названия метода отделяют специальные символы (точка в Delphi).

```
procedure TmyClass.Initialize;  
begin  
Item1:= 1;  
Item2:= “”; end;
```

Класс – это тип данных, такой же, как любой другой базовый или сложный тип. На его основе можно описывать конкретные объекты.

```
Var C1, C2: TmyClass;
```

Доступ к свойствам объектов и к их методам осуществляется так же, как к полям записей, через точку: C1.Item1:=5; C2.initialize;

Объединение данных с методами в одном типе называется инкапсуляцией.

Наследование. Важнейшая характеристика класса – возможность создания на его основе новых классов с наследованием всех его свойств и методов и добавлением собственных. Класс, не имеющий предшественника, называется базовым. Наследование позволяет создавать новые классы, повторно используя уже готовый исходный код и не тратя времени на его переписывание.

Полиморфизм. В большинстве случаев методы базового класса у классов-наследников приходится переопределять. Все переопределяемые методы по названию будут совпадать с методами базового объекта, однако компилятор по типу объекта распознает, какой конкретно метод надо использовать. Такое свойство объектов переопределять методы наследуемого класса называется полиморфизмом.

Визуальное программирование. Технологии объектного, событийного и структурного программирования сегодня объединены в RAD-системах, которые содержат множество готовых классов, представленных в виде визуальных компонентов, которые добавляются в программу одним щелчком мыши. Надо только спроектировать внешний вид окон своего приложения и определить обработку основных событий – какие операторы будут выполняться при нажатии на кнопки, при выборе пунктов меню или щелчках мышкой. Весь вспомогательный исходный код среда сгенерирует сама, позволяя полностью сосредоточиться только на реализации алгоритма.

Порядок выполнения работы

1. Постановка задачи. В среде Delphi разработать проект, содержащий визуальные интерфейсные компоненты для ввода или вывода данных. Ознакомиться с основными окнами Delphi, со свойствами объектов.

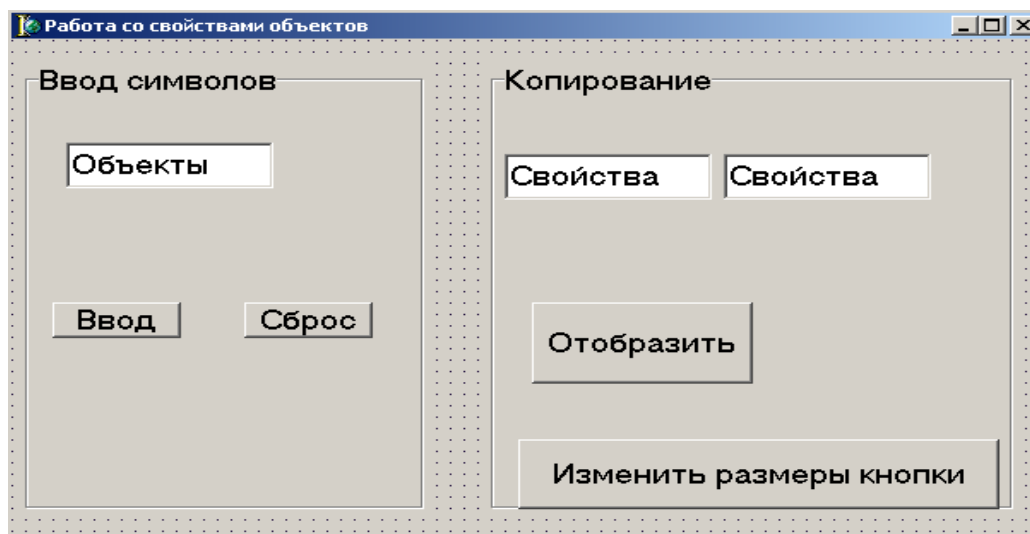


Рис. 2.20. Диалоговое окно программы «Знакомство со свойствами объектов»

2. Конструирование формы. На форме необходимо расположить компоненты типа Tbutton для кнопок *ввод*, *сброс*, *отобразить*, *изменить размеры кнопки*. Нанесем на кнопки надписи, используя свойство Caption. Для вывода данных используем компонент типа Tedit. Вид формы, который должен получиться, показан на рис. 2.20.

3. Написание программы. Написать программу при нажатии на кнопку *ввод*, в текстовое поле введите текст. При нажатии на кнопку **Сброс** текстовое поле очищается. При нажатии на кнопку **Отобразить** текст, который был напечатан в первом текстовом поле, копируется во второе. При нажатии на вторую кнопку меняются названия формы и первой кнопки, ее размеры (ширина и длина).

Содержание отчета

Программа с учетом основных рекомендаций.

Вопросы для самопроверки

1. Какие свойства отвечают за расположение объекта?
2. Значения объекта Enabled.
3. Какие параметры задаются в свойстве Font?
4. Какие свойства отвечают за размер объекта?
5. Допустимые значения свойства Name.
6. Какие программные среды вы знаете?
7. Какие средства разработки вы знаете?

Лабораторная работа № 10 МОДЕЛИРОВАНИЕ РАБОТЫ ИС (на примере калькулятора)

Цель работы: спроектировать модель поведения некоторого устройства, например калькулятора.

Порядок выполнения работы

1. Постановка задачи. В среде Delphi разработать проект, моделирующий работу калькулятора, содержащего кнопки цифр и операций, кнопку сброса, табло для набираемых данных и выдачи результата вычислений. Обеспечить вычисление выражения, содержащего несколько операций. Создать справку (сведения о программе и разработчиках).

2. Конструирование формы. На форме необходимо расположить множество компонентов типа TSpeedbutton одинаковых размеров для кнопок с цифрами и знаками операций (*, +, -, /, =, математические функции), кнопок сброса и десятичной запятой. Сначала вынесем один компонент типа TSpeedbutton, отредактируем его размеры и шрифт (свойство Font). Затем средствами редактирования среды Delphi (используя пункт меню Edit) размножим этот компонент в необходимом количестве и расставим на нужные места в форме. Нанесем на кнопки надписи – цифры, знаки операций и т. д., используя свойство Caption. Для вывода результата используем компонент типа Tedit. Полученный вид формы показан на рис. 2.21. В листинге 2 приведены фрагменты программы.

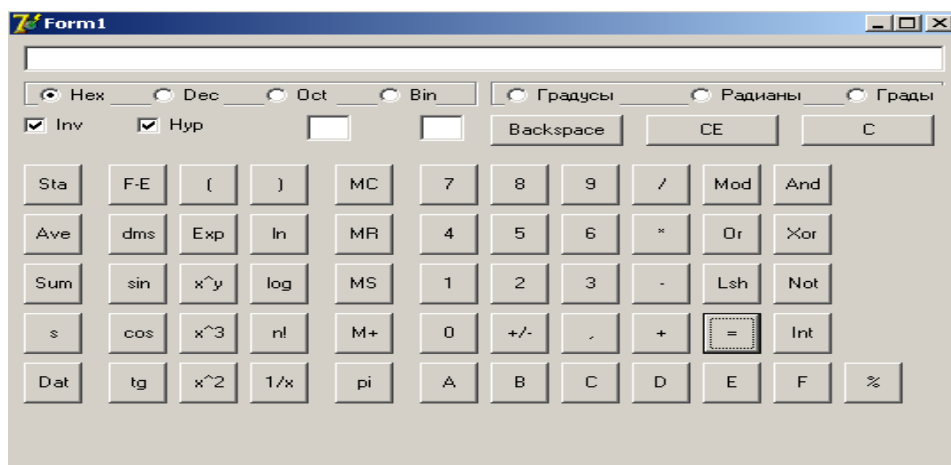


Рис. 2.21. Диалоговое окно программы «Калькулятор»

3. Написание программы «Калькулятор».

Листинг1. «Калькулятор».

```
procedure TForm1.Button № Click(Sender: TObject);  
//цифры 0..9
```

```

begin
edit1.Text:=edit1.Text+Button №.Caption; end;
//десятичная запятая
procedure TForm1.Button44Click(Sender: TObject);
begin
If Pos(',',edit1.Text)<>0 then edit1.Text:=edit1.Text
else edit1.Text:=edit1.Text+',';
end;
//арифметические операции
procedure TForm1.Button45Click(Sender: TObject); //+
begin
a:=0; k:=1;
a:=strtofloat(edit1.Text);
edit1.Text:="";
end;
//кнопка равно
procedure TForm1.Button46Click(Sender: TObject);
begin
case k of
1:edit1.Text:=floattostr(a+strtofloat(edit1.Text));
2:edit1.Text:=floattostr(a-strtofloat(edit1.Text));
3:edit1.Text:=floattostr(a*strtofloat(edit1.Text));
...
10:edit1.Text:=inttostr(f mod strtoint(edit1.Text));
end;
end;
//*
procedure TForm1.Button59Click(Sender: TObject);
begin
a:=0;k:=3;
a:=strtofloat(edit1.Text);
edit1.Text:="";
end;
// Нахождение остатка от деления
procedure TForm1.Button13Click(Sender: TObject);
begin //остаток от деления
f:=0;k:=10;
f:=strtoint(edit1.text);
edit1.Text:="";
end;
//1/x
procedure TForm1.Button51Click(Sender: TObject);

```

```

Var n:real;
begin
n:=strtofloat(edit1.Text);
edit1.Text:=floattostr(1/n);
end;
    // Занесение отображаемого числа в память. Переменные описать в разделе
        описания глобальных переменных.
procedure TForm1.Button41Click(Sender: TObject);
begin
n:=n+strtofloat(edit1.Text);
edit2.Text:='M';
edit1.Text:=''; end;
    //Замена отображаемого числа числом, хранящимся в памяти.
procedure TForm1.Button19Click(Sender: TObject);
begin
edit1.Text:=floattostr(n);
end;
        // Тригонометрические функции
procedure TForm1.Button27Click(Sender: TObject);
begin t:=strtofloat(edit1.text);
edit1.Text:='';
if (CheckBox1.Checked=true)and(CheckBox2.Checked=true)
then edit1.Text:=floattostr(arcsinh(t))
else if (CheckBox1.Checked=false)and(CheckBox2.Checked=true)
then edit1.Text:=floattostr(sinh(t))
else if (CheckBox1.Checked=true)and(CheckBox2.Checked=False)
then
begin
if Abs(t)<=1 then
begin
t:=arcsin(t);
perevod1;edit1.Text:=floattostr(t);
end
else edit1.text:='Недопустимый аргумент функции';
end
else
begin
perevod;
edit1.Text:=floattostr(sin(t));
end;
CheckBox1.Checked:=False; CheckBox2.Checked:=False; end;

```

```

// Логарифм натуральный.
procedure TForm1.Button18Click(Sender: TObject);
begin
if CheckBox1.Checked=true
then edit1.Text:=floattostr(exp(strtfloat(edit1.Text)))
else edit1.Text:=floattostr(ln(strtfloat(edit1.Text)));
CheckBox1.Checked:=false;
end;

// Логарифм десятичный.
procedure TForm1.Button29Click(Sender: TObject);
begin
if CheckBox1.Checked=true
then edit1.Text:=floattostr(power(10,strtfloat(edit1.text)))
else edit1.text:=floattostr(log10(strtfloat(edit1.Text)));
CheckBox1.Checked:=false;
end;

// Факториал
procedure TForm1.Button40Click(Sender: TObject);
Var n,s:integer;
begin
n:=strtoint(edit1.Text);
edit1.Text:="";
s:=1;
While (n>0) do
begin
s:=s*n;
n:=n-1;
end;
edit1.Text:=inttostr(s);
end;

```

Содержание отчета

Программа с учетом основных рекомендаций.

Вопросы для самопроверки

1. Что является аргументом функции \sin ?
2. Какие системы исчисления используются в тригонометрических функциях?
3. Как преобразовать переменные разного типа?
4. Как задать математические функции?
5. Как реализовать процедуру вычисления факториала двумя способами?

Лабораторная работа № 11 ОБРАБОТКА МАССИВОВ ДАННЫХ

Цель работы: изучить основные понятия темы. Научиться решать задачи с таблицами.

Порядок выполнения работы

1. Постановка задачи. В среде Delphi разработать проект, содержащий визуальные интерфейсные компоненты для ввода или вывода данных – на примере матриц и векторов.

2. Конструирование формы:

указать формулировку задачи как заголовок формы (caption);

для задания таблиц (матриц и векторов) использовать компонент TStringGrid: подобрать размеры его ячеек, использовать для запуска вычислений кнопку (Tbutton); для заполнения результата использовать компоненты TLabel или TEdit (обеспечить размер ячейки, подходящий для представления результата);

для указания размеров таблицы использовать свойства Colcount (количество столбцов) и Rowcount (количество строк) компонента TStringGrid;

для компонента TStringGrid задать свойства FixedCol и FixedRow равными 0 (для диалоговой работы со всеми ячейками этого компонента). Если матрица, для вектора по одному из измерений нужно указать длину, равную 1;

Для обеспечения ввода и редактирования элементов таблицы задать в свойстве Options компонента TStringGrid режим goEditing.

Таблица 2.13

Значения свойств компонента TStringGrid1

Свойство	Значение
Colcount	5
Rowcount	1
FixedCol	0
DefaultRowHeight	24
Height	24
DefaultColWidth	64
Width	328
Options.goEditing	True
Options.AlwaysShowEditing	True
Options.goTabs	True

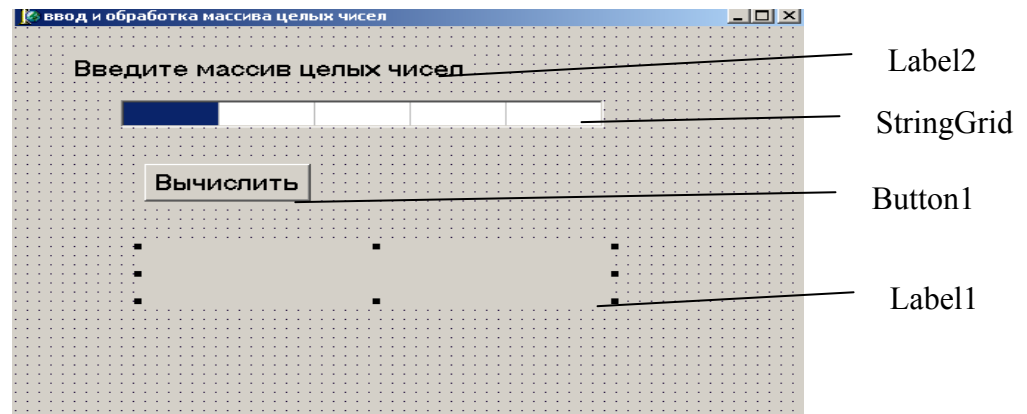


Рис. 2.22. Диалоговое окно программы «Ввод и обработка массива»

Листинг2. Ввод и обработка массива.

```

procedure TForm1.Button1Click(Sender: TObject);
var a:array[1..5] of integer; // массив
    summ:integer;           // сумма элементов
    sr:real;                // среднее арифметическое
    i:integer;
begin
    // ввод одномерного массива
    for i:=1 to 5 do
    a[i]:=strtoint(stringgrid1.Cells[i-1,0]);
    // обработка массива

    summ:=0;
    for i:=1 to 5 do
    summ:=summ+a[i];
    sr:=summ/5;

    // вывод результата
    label1.Caption:='Сумма элементов:' + inttostr(summ)+ #13+
    'среднее арифметическое:'+floattostr(sr); end; end.

```

Пример описания, ввода и вывода двумерного массива:

```

// двумерный массив, состоящий из элементов целого типа.
var a:array[1..5,1..5] of integer;

// ввод двумерного массива
for i:=1 to 5 do
for j:=1 to 5 do
a[i,j]:=strtoint(stringgrid1.Cells[i-1,j-1]);
// вывод двумерного массива
for i:=1 to 5 do

```

```
for j:=1 to 5 do  
stringgrid1.Cells[i-1,j-1]:=inttostr (a[i,j]);
```

Содержание отчета

Программа с учетом основных рекомендаций.

Вопросы для самопроверки

1. Что понимается под массивом?
2. Как обозначаются в программе элементы массива?
3. Как описывается одномерный массив в разделе переменных?
4. Как организуется в программе ввод одномерного массива?
5. Как организуется в программе вывод одномерного массива?
6. Как описывается двумерный массив в разделе переменных?
4. Как организуется в программе ввод двумерного массива?
5. Как организуется в программе вывод двумерного массива?

Задание

1. Дана матрица (5x5). Поменять местами 4-ю строку с 3-м столбцом
2. Записать +1 вместо максимального элемента массива, а -1 вместо минимального. Матрица (5x5)
3. Найти наибольший элемент главной диагонали матрицы (10x10). Вывести всю строку, в которой он находится
4. Найти номер наименьшего из положительных элементов матрицы (5x5)
5. Дана матрица (6x6). Поменять местами 6-й столбец с 1-м столбцом.
6. Дана матрица (5x5). Найти произведение элементов главной диагонали
7. Вычислить сумму элементов матрицы (6x6), расположенных над главной диагональю
8. Определить количество положительных и отрицательных элементов матрицы (10x10)
9. Вычислить произведение элементов матрицы (6x6), расположенных под главной диагональю
10. Сформировать матрицу (5x5). Элементы которой вычисляются по формуле $b[i,j]=i+\sin(3.14/j)$
11. Найти сумму наибольшего и наименьшего элементов массива вещественных чисел массива

Лабораторная работа № 12 ПОИСК В МАССИВЕ ЗАДАННОГО ЭЛЕМЕНТА

Цель работы: изучить основные методы поиска заданного элемента в массиве. Научиться находить нужный элемент в массиве разными методами.

Порядок проведения работы

1. Постановка задачи. Организовать в массиве поиск заданного элемента с помощью различных алгоритмов (алгоритм простого перебора, бинарный поиск, интерполяционный поиск).

2. Конструирование формы. Компонент StringGrid1 применяется для ввода элементов массива. Поле метки Label1 используется для вывода результатов поисков и протоколов. Протокол выводится, если установлен флажок **Выводить протокол**. Протокол содержит значения переменных *verh*, *niz*, *sred*. Эта информация, выводимая во время поиска, полезна для понимания сути алгоритма. Значок компонента CheckBox1 находится на вкладке Standard. Компоненты Groupbox1 и Radiobutton1 тоже находятся на вкладке Standard. В поле Edit1 вводится тот элемент, который надо найти.

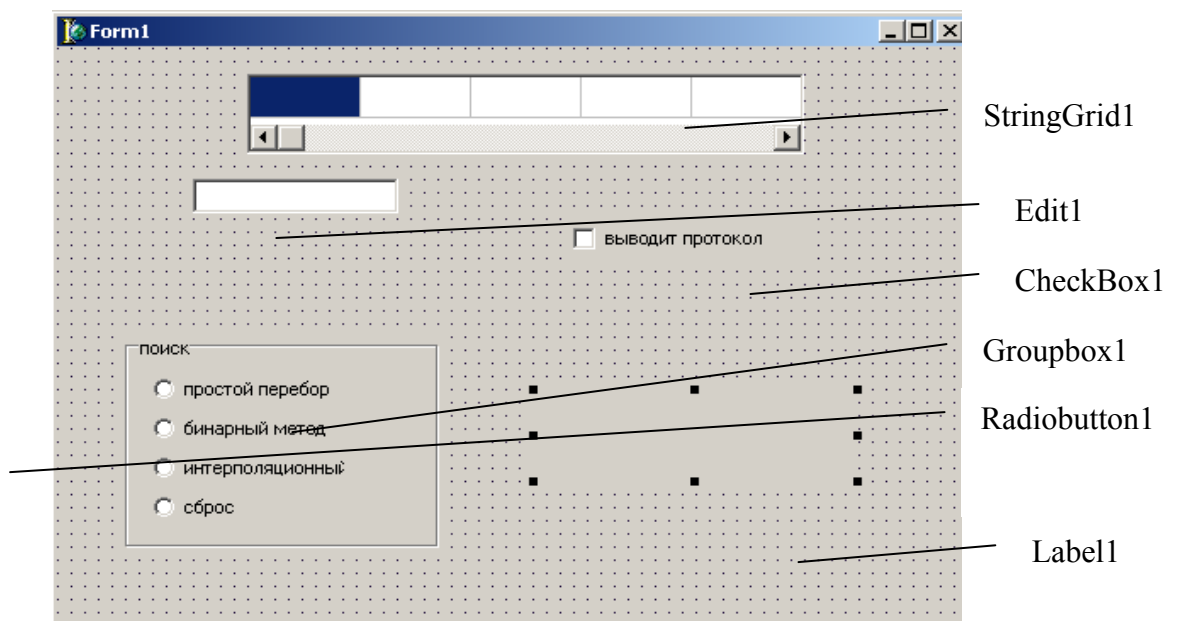


Рис. 2.23. Диалоговое окно программы «Три метода поиска в массиве»

3. Написание программы.

Алгоритм простого перебора

Поиск осуществляется последовательным сравнением элементов массива с образцом до тех пор, пока не будет найден элемент, равный образцу, или не будут проверены все элементы. Алгоритм простого перебора применяется, если элементы массива не упорядочены. Ниже приведен

фрагмент программы поиска в массиве целых чисел. Перебор элементов массива осуществляется инструкцией `repeat`, в теле которой инструкция `if` сравнивает текущий элемент массива с образцом и присваивает переменной `found` значение `true`, если текущий элемент и образец равны. Цикл завершается, если в массиве обнаружен элемент, равный образцу (в этом случае значение переменной `found` равно `true`), или если проверены все элементы массива. По завершении цикла по значению переменной `found` можно определить, успешен поиск или нет.

Листинг 3.1. Поиск простого перебора.

```
repeat
if a[i] =obr
then found:= true
else i:=i+1;
until (i>10) or (found=true);
```

Алгоритм бинарного поиска

Пусть есть упорядоченный по возрастанию массив целых чисел. Нужно определить, содержит ли этот массив некоторое число (образец).

Метод (алгоритм) бинарного поиска реализуется следующим образом:

1. Сначала образец сравнивается со средним (по номеру) элементом массива:

Если образец равен среднему элементу, то задача решена.

Если образец больше среднего элемента, то это значит, что искомый элемент расположен ниже среднего элемента (между элементами с номерами `sred+1` и `niz`) и за новое значение `verh` принимается `sred+1`, а значение `niz` не меняется.

Если образец меньше среднего элемента, то это значит, что искомый элемент расположен выше среднего элемента (между элементами с номерами `verh` и `sred-1`) и за новое значение `niz` принимается `sred-1`, а значение `verh` не меняется.

2. После того как определена часть массива, в которой может находиться искомый элемент, по формуле $(niz-verh)/2+verh$ вычисляется новое значение `sred` и поиск продолжается.

Алгоритм бинарного поиска заканчивает свою работу, если искомый элемент найден или если перед выполнением очередного цикла поиска обнаруживается, что значение `verh` больше, чем `niz`.

Листинг 3.2. Бинарный поиск.

```
repeat
sred:=trunc((niz-verh)/2)+verh;
if checkbox1.checked
then label1.Caption:= label1.Caption+inttostr(v)+#9
+inttostr(niz)+ #9+inttostr(sred)+#13;
```

```
n:=n+1;
if a[sred]=obr
then f:=true
else if obr<a[sred]then niz:=sred-1 else v:=sred+1;
until (verh>niz) or (f=true);
```

Алгоритм интерполяционного поиска

В данном алгоритме массив также разбивается на две части, но точка разбиения зависит от искомого значения. В этом случае используется метод интерполяции, т. е. предсказание результата по имеющимся данным, в качестве исходных данных выступают порядковые номера и значения элементов массива. Для равномерных списков данный алгоритм позволяет на каждом шаге отбросить большую часть массива, что приводит к более быстрой сходимости, чем в случае двоичного поиска.

Листинг 3.3. Интерполяционный поиск.

```
repeat
sred:=trunc(v+((obr-a[v])*(niz-v)/(a[niz]-a[v])));
if checkbox1.checked
then label1.Caption:= label1.Caption+inttostr(v)+#9
+inttostr(niz)+ #9+inttostr(sred)+#13;
n:=n+1;
if a[sred]=obr
then f:=true; exit;
else if obr<a[sred]then niz:=sred-1 else v:=sred+1;
until (verh>niz) or (f=true);
```

Содержание отчета

Программа с учетом основных рекомендаций.

Вопросы для самопроверки

1. Условия выхода из процедуры.
2. Двоичный поиск в верхней части массива.
3. Объяснить формулу sred для интерполяционного поиска.
4. Двоичный поиск в нижней части массива.
5. Преимущества и недостатки различных методов поиска.
6. Когда применяют метод бинарного поиска?
7. Каким образом реализуется метод бинарного поиска?
8. Когда алгоритм бинарного поиска заканчивает свою работу?
9. В чем суть метода простого перебора?

Лабораторная работа № 13 СОРТИРОВКА ДАННЫХ В МАССИВЕ

Цель работы: изучить основные методы сортировки элементов массива, научиться сортировать элементы массива с помощью разных алгоритмов сортировки.

Порядок проведения работы

1. Постановка задачи. Организовать сортировку элементов массива несколькими способами (метод прямого выбора, метод прямого обмена, метод вставки).

2. Конструирование формы. Компонент StringGrid1 применяется для ввода элементов массива. Поле метки Label1 используется для вывода результатов сортировки и протоколов. Значок компонента CheckBox находится на вкладке Standard. Компоненты Groupbox и Radiobutton тоже находятся на вкладке Standard. Пример формы изображен на рис. 2.24.

3. Написание программы. Под сортировкой массива понимается процесс перестановки элементов массива, целью которого является размещение элементов массива в определенном порядке.

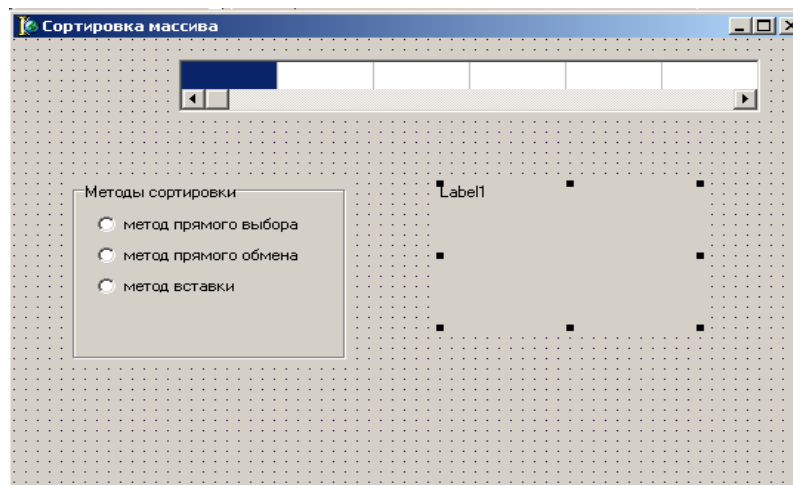


Рис. 2.24. Диалоговое окно программы «Методы сортировки элементов массива»

Алгоритм прямого выбора

Сортировка методом прямого выбора может быть представлена так:

Просматривая массив от первого элемента, найти минимальный элемент и поместить его на место первого элемента, а первый – на место минимального.

Просматривая массив от второго элемента, найти минимальный элемент и поместить на место второго элемента, а второй – на место минимального.

И так далее до предпоследнего элемента.

Процедура сортировки приведена в листинге 5.1.

Листинг 4.1. Сортировка массива простым выбором.

```
min:=i;
for j:=i+1 to 10 do
if a[j] < a[min]
then min:=j;
buf:=a[i];
a[i]:a[min];
a[min]:=buf;
```

Алгоритм методом обмена

В основе алгоритма лежит обмен соседних элементов массива. Каждый элемент массива, начиная с первого, сравнивается со следующим, и если он больше следующего, то элементы меняются местами. Таким образом, элементы с меньшим значением продвигаются к началу массива (всплывают), а элементы с большим значением – к концу массива (тонут). Поэтому данный метод сортировки обменом называют методом “пузырька”. Этот процесс повторяется столько раз, сколько элементов в массиве, минус единица. Процедура сортировки приведена в листинге 5.2

Листинг 4.2. Сортировка массива методом обмена.

```
repeat
for k:=1 to size-1 do
if a[k] < a[k+1]
then begin
buf:=a[k];
a[k]:a[k+1];
a[k+1]:=buf; end;
```

Алгоритм метода вставки

Создается новый массив. Элементы исходного списка помещаются на определенные позиции нового списка, при этом элементы, большие текущего значения, сдвигаются на одну позицию вниз; если таких элементов нет, выполнение процедуры прерывается.

Листинг 4.3 Метод вставки.

```
min:=1;
s:=min-1;
for i:=1 to 5 do
begin
next:=a[i];
for j:=min to s do
if (a[j]>=next)then break;
for k:=s downto j do
```

```
a[k+1]:=a[k];  
a[j]:=next;  
inc(s);  
end;
```

Сортировка методом подсчета

Создается массив (C), состоящий из целых чисел, элементы которого определяются значениями исходного массива. Если используется массив с предыдущего шага, его необходимо обнулить. Исходный список просматривается для подсчета количества определенных значений, эти данные сохраняются в массиве C. Число повторений значений в отсортированном массиве определяется элементами массива C.

Листинг 4.4. Метод подсчетом.

```
min:=1;  
for i := 1 to 100 do c[i] := 0;  
for i := min to 6 do  
  c[a[i]] := c[a[i]] + 1;  
new:= min;  
for i := 1 to 100 do  
  for j := 1 to c[i] do  
    begin  
      a[new] := i;  
      new := new + 1; end;
```

Алгоритм быстрой сортировки

Рекурсивный алгоритм сортировки, предназначен для работы с большими массивами. Исходный список разбивается на две части, в качестве разделительного элемента можно использовать:

- середину массива,
- среднее значение,
- случайное число.

Элементы списка сравниваются с разделителем, меньшие помещаются в левую часть списка, большие в правую часть. Данная операция повторяется для каждого подсписка, пока количество элементов не достигнет определенного минимального значения, по умолчанию 1.

Алгоритм блочной сортировки.

В данном алгоритме не используется поэлементное сравнение. Исходный массив разбивается на блоки, элементы помещаются в определенный блок в зависимости от значения. Если в блоке более одного

элемента, то процедура рекурсивно вызывается для данного подписка. Если в каждом блоке находится по одному элементу, то список отсортирован.

Вопросы для самопроверки

1. Условия выхода из сортировки вставкой?
2. Что такое рекурсия?
3. Ограничения на массив при сортировке подсчетом?
4. Преимущества и недостатки различных алгоритмов сортировки?
5. Модифицированный алгоритм прямого перебора?

Задания

1. Дана матрица (10x10). Отсортировать 4-й столбец методом обмена.
2. Дана матрица (10x10). Отсортировать 4-ю строку методом прямого выбора.
3. Дана матрица (10x10). Отсортировать 6-ю строку методом вставки.
4. Дана матрица (10x10). Отсортировать главную диагональ методом обмена по убыванию.
5. Дана матрица (10x10). Отсортировать главную диагональ методом обмена по возрастанию.

Лабораторная работа № 14 РАБОТА С ФАЙЛАМИ

Цель работы: научиться записывать данные в файл и читать их из файла.

Порядок проведения работы

1. Постановка задачи. Создать программу “Конвектор валют” с проверкой внесенных значений и организовать возможность чтения и записи данных в текстовый файл. Для данного приложения использовать объект `MainMenu`, в котором прописать основные команды работы с файлами, выполнения расчетов и вывода справочной информации.
2. Конструирование формы. Для того чтобы создать главное меню формы (программы), нужно выбрать на вкладке `Standard` компонент класса `TmainMenu`. После установки компонента на форму необходимо создать пункты меню. Для этого следует дважды щелкнуть на компоненте левой кнопкой мыши, после чего перейти в окно инспектора объектов и ввести

название пункта меню в строке свойства Caption. Нажать Enter – пункт меню готов и можно переходить к следующему. В результате должна получиться форма вида, изображенного на рис. 2.25

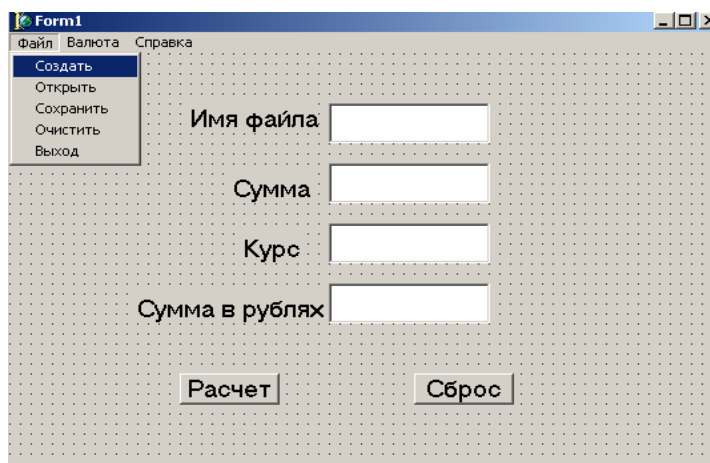


Рис. 2.25. Диалоговое окно программы «Конвектор валют»

3. Написание программы.

Объявление файла

Файл – это именованная структура данных, представляющая собой последовательность элементов данных одного типа, количество элементов последовательности практически не ограничено. Как и любая структура данных (переменная, массив) программы, файл должен быть объявлен в разделе описания переменных. При объявлении файла указывается тип элементов файла.

В общем виде объявление файла выглядит так:

Имя: file of тип элементов;

Примеры:

res: file of char; // файл символов.

t: file of real; // файл вещественных чисел.

n: file of integer; // файл целых чисел.

Файл, компонентами которого являются данные символьного типа, называется символьным, или текстовым. Описание текстового файла в общем виде выглядит так:

Имя: TextFile;

где *Имя* – имя файловой переменной; Textfile – обозначение типа, показывающее, что *имя* – это файловая переменная, представляющая текстовый файл.

Объявление файловой переменной задает только тип компонентов файла. Для того чтобы программа могла выводить данные в файл или считывать

данные из файла, необходимо указать конкретный файл, т. е. связать файловую переменную с конкретным файлом (задать имя файла).

Имя файла задается вызовом процедуры Assignfile, связывающей файловую переменную с конкретным файлом.

Описание процедуры Assignfile выглядит следующим образом:

Assignfile (var f, имя файла:string)

Имя файла задается согласно принятым в Windows правилам. Оно может быть полным, т. е. состоять не только непосредственно из имени файла, но и включать путь к файлу (имя диска, каталогов и подкаталогов).

Примеры:

```
Assignfile (f,'a:\result.txt');
```

```
Assignfile (f,'\students\petrov\result.txt');
```

```
fname:= (f,'korni.txt');
```

```
Assignfile (f,'fname');
```

Вывод в файл

Непосредственно вывод в текстовый файл осуществляется при помощи инструкции write или writeln. В общем виде эти инструкции записываются следующим образом:

```
write(файловая переменная, список вывода);
```

```
writeln(файловая переменная, список вывода);
```

где файловая переменная – переменная идентифицирующая файл, в который выполняется вывод; список вывода – разделенные запятыми имена переменных, значения которых надо вывести в файл. Помимо имен переменных в список вывода можно включать строковые константы.

Например,

```
write(f, 'корни уравнения', x1, x2);
```

Различие между инструкциями write и writeln состоит в том, что инструкция writeln после вывода всех значений, указанных в списке вывода, записывает в файл символ “новая строка”.

Открытие файла для вывода

Перед выводом в файл его необходимо открыть. Если программа, формирующая выходной файл, уже использовалась, то возможно, что файл с результатами работы программы уже есть на диске. Тогда выбираем: либо заменить старые данные, либо новые данные добавить к старым.

Возможны следующие режимы открытия файла для записи в него данных:

перезапись (запись нового файла поверх существующего или создание нового файла);

добавление в существующий файл.

Для того чтобы открыть файл в режиме создания нового файла или замены существующего, необходимо вызвать процедуру Rewrite(f), f – файловая переменная типа Textfile.

Для того чтобы открыть файл в режиме добавления к уже существующим данным, находящимся в этом файле, нужно вызвать процедуру Append(f), f – файловая переменная типа Textfile.

На рис. 2.25 приведено диалоговое окно программы, которая выполняет запись или добавление в текстовый файл.

В листинге 5.1 приведена процедура, которая запускается нажатием командной кнопки **Записать**. Она открывает файл в режиме создания нового замещения существующего файла и записывает текст, находящийся в поле компонента Memo1.

Имя файла нужно ввести во время работы в поле Edit1. Можно задать predetermined имя файла во время разработки формы приложения. Для этого надо присвоить значение, например test.txt, свойству Edit1.text.

Листинг 5.1. Создание нового или замена существующего файла.

```
Begin
fName:= Edit1.text;
Assignfile (f, fName);
rewrite(f);           //открыть для перезаписи;
//запись в файл
for i:=0 to Memo1.lines.Count do // строки нумеруются с нуля.
writeln(f, Memo1.lines[i]);
closefile(f);        // закрыть файл.
end;
```

В листинге 5.2. приведена процедура, которая запускается нажатием командной кнопки **Добавить**. Она открывает файл, имя которого указано в поле Edit1, и добавляет в него содержимое поля Memo1.

Листинг 5.2. Добавление в существующий файл.

```
Begin
fName:= Edit1.text;
Assignfile (f, fName);
Append(f);           //открыть для добавления;
//запись в файл
for I:=0 to Memo1.lines.Count do // строки нумеруются с нуля.
Writeln(f, Memo1.lines[I]);
closefile(f);        // закрыть файл.
End;
```

Ввод из файла

Программа может вводить исходные данные не только с клавиатуры, но и из текстового файла. Для этого нужно объявить файловую переменную типа Textfile, назначить ей при помощи инструкции Assignfile имя файла, из которого будут считываться данные, открыть файл для чтения (ввода) и прочитать (ввести) данные, используя инструкцию read или readln.

Открытие файла

Открытие файла для ввода (чтения) выполняется вызовом процедуры Reset, имеющей один параметр – файловую переменную. Перед вызовом процедуры Reset с помощью функции Assignfile файловая переменная должна быть с конкретным файлом.

Например, следующие инструкции открывают файл для ввода:

```
Assignfile (f,'a:\result.txt');
```

```
Reset (f);
```

Чтение данных из файла

Чтение из файла выполняется при помощи инструкций read и readln, которые в общем виде записываются следующим образом:

```
read(файловая переменная, список переменных);
```

```
readln(файловая переменная, список переменных);
```

где: файловая переменная – переменная типа Textfile; список переменных – разделенные запятыми имена переменных.

Чтение чисел

Например, если текстовый файл a:/data.txt содержит следующие строки:

```
23 15
```

```
45 28
```

```
56 70
```

то в результате выполнения инструкций:

```
Assignfile (f,'a:\data.txt');
```

```
reset (f);
```

```
read(f,a);
```

```
read(f,b,c);
```

```
read(f,d);
```

Значения переменных будут следующими: a=23, b=15, c=45, d=28;

Чтение строк

В программе строковая переменная может быть объявлена с указанием длины или без нее.

Например:

```
stroka: string [15];
```

```
str:string;
```

Если одной инструкцией readln осуществляется ввод нескольких, например двух переменных, то первая переменная будет содержать столько символов, сколько указано в ее объявлении или, если длина не указана, всю

строку файла. Вторая переменная будет содержать оставшиеся символы текущей строки или, если таких символов нет, не будет содержать ни одного символа.

Пусть, например, текстовый файл spisok.txt содержит строки:

Егоров Дима
Захаров Ваня
Иванова Оля

В табл. 2.14 приведено несколько вариантов объявления переменных, инструкции чтения из файла spisok.txt и значения переменных после выполнения инструкций чтения.

Таблица 2.14

Примеры объявления переменных, инструкций чтения из файла

Объявления переменных	Инструкция чтения из файла	Значение переменных после чтения из файла
fam: string[14] name: string[10]	readln(f, fam, name)	fam='Егоров' name='Дима'
fam, name: string;	readln(f, fam, name)	fam='Егоров Дима' name=''
drug: string[80]	readln(f, drug)	drug='Егоров Дима'

Листинг 5.3. Чтение из файла.

```
while not eof (f) do      //пока не конец файла
begin
readln(f,buf);          //прочитать из файла
memo1.lines.add(buf);   //добавить строку в поле memo1
end;
closefile(f);           //закрыть файл
```

Содержание отчета

Программа с учетом основных рекомендаций.

Вопросы для самопроверки

1. Какие функции называются функциями низкого уровня доступа?
2. Отличие функций append и rewrite.
3. Функции работы с текстовыми файлами.
4. Что является аргументом функции Assignfile?

Задания

1. Считать данные из файла. Определить количество положительных элементов массива.
2. Считать данные из файла. Определить сумму четных элементов массива.
3. Считать данные из файла. Определить min-й среди положительных.
4. Записать данные в файл. Определить сумму нечетных элементов массива.
5. Считать данные из файла. Определить количество отрицательных элементов массива.

Лабораторная работа № 15 СОЗДАНИЕ БД

Цель работы: освоить процесс разработки типичной базы данных (используется пример с заданием «Поваренная книга») с использованием визуального интерфейса Microsoft Access.

Порядок выполнения работы

1. Постановка задачи. В приложении Microsoft Access разработать и заполнить базу данных, предназначенную для представления данных, описанных в задании.
2. Разработка структуры базы данных в Microsoft Access.
 - 2.1. Нормализация исходных данных до 3-й нормальной формы.
 - 2.2. Создание нормализованных таблиц в Access.
 - 2.3. Определение связей между таблицами.
 - 2.4. Назначения для связей поддержки целостности в виде каскадного удаления связанных записей.
3. Заполнение таблиц данными для контроля правильности работы базы данных и клиентского приложения.

На рис. 2.26 приведена схема базы данных «Поваренная книга». Детальная структура каждой из таблиц описана ниже.

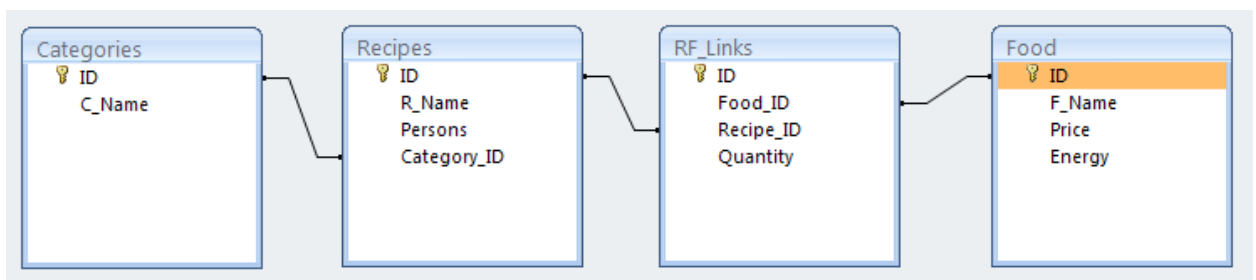


Рис. 2.26 Структура базы данных CookBook (Поваренная книга)

Таблица 1. Структура таблицы Categories (Категории продуктов)

Имя поля	Тип данных	Описание
ID	Счетчик	Идентификатор, первичный ключ
C_Name	Текстовый	Наименование категории

Таблица 2. Структура таблицы Recipes (Рецепты)

Имя поля	Тип данных	Описание
ID	Счетчик	Идентификатор, первичный ключ
R_Name	Текстовый	Наименование блюда
Persons	Числовой	Количество персон
Category_ID	Числовой	Идентификатор категории

Таблица 3. Структура таблицы RF_Links (Связи «рецепты-продукты»)

Имя поля	Тип данных	Описание
ID	Счетчик	Идентификатор, первичный ключ
Food_ID	Числовой	Идентификатор продукта
Recipe_ID	Числовой	Идентификатор рецепта
Quantity	Числовой	Количество продукта в составе рецепта

Таблица 4. Структура таблицы Food (Продукты)

Имя поля	Тип данных	Описание
ID	Счетчик	Идентификатор, первичный ключ
F_Name	Текстовый	Наименование продукта
Price	Денежный	Цена
Energy	Числовой	Калорийность

Содержание отчета

Структурная схема базы данных, полное описание структуры таблиц.

Вопросы для самопроверки

1. В чем необходимость установки связей между таблицами на уровне СУБД?

2. В чем преимущество нормализованной базы данных в случае вашего задания?

3. Обоснуйте ваш выбор типов данных для полей таблиц.

Задание
Поваренная книга

Структура данных:

Таблица 1

Продукты

Наименование	Цена	Калорийность
--------------	------	--------------

Таблица 2

Рецепты

Название блюда	Список продуктов и их количества	Размер готовой порции (число персон)	Категория блюда: первое, второе, напиток и т.д.
----------------	----------------------------------	--------------------------------------	---

Функции:

1. Ведение таблиц.
2. Вывод списка продуктов и сметы для данного блюда на N персон.
3. Поиск цены по наименованию продукта.

Лабораторная работа № 16
ПОСТРОЕНИЕ ПРИЛОЖЕНИЙ ЛОКАЛЬНЫХ БАЗ ДАННЫХ
СРЕДСТВАМИ DELPHI

Цель работы: научиться разрабатывать клиентские приложения в среде Delphi для управления базами данных.

Порядок выполнения работы

1. Постановка задачи. В среде Delphi разработать проект клиентского приложения для управления базой данных «Поваренная книга» (см. задания к лабораторной работе № 15).

2. Создать следующие формы:

- MainForm (модуль MainUnit),
- RecipeForm (модуль RecipeUnit),
- FoodForm (модуль FoodUnit),
- ComponentForm (модуль ComponentUnit),
- ListCalcForm (модуль ListCalcUnit).

3. Расположить и настроить компоненты на формах.

3.1. На форме MainForm:

```
FoodGrid: TDBGrid
    Align = alBottom
    DataSource = FoodDataSource
Splitter1: TSplitter
    Align = alBottom
RecipesGrid: TDBGrid
    Align = alLeft
    DataSource = RecipesDataSource
Splitter2: TSplitter
    Align = alLeft
ComponentsGrid: TDBGrid
    Align = alClient
    DataSource = ComponentsDataSource
CookBookConnection: TADOConnection
    LoginPrompt = False
FoodTable: TADOTable
    Connection = CookBookConnection
    TableName = Food
RFLinksTable: TADOTable
    Connection = CookBookConnection
    TableName = RF_Links
CategoryTable: TADOTable
    Connection = CookBookConnection
    TableName = Categories
RecipesTable: TADOTable
    Connection = CookBookConnection
    TableName = Recipes
ComponentsQuery: TADOQuery
    Connection = CookBookConnection
RecipesQuery: TADOQuery
    Connection = CookBookConnection
    SQL =
        select Recipes.ID, R_Name, Persons, C_Name,
            Category_ID
        from Recipes, Categories
        where Category_ID=Categories.ID
        order by R_Name
OpenDBDialog: TOpenDialog
    Filter = Файлы Access|*.mdb
FoodDataSource: TDataSource
```

DataSet = FoodTable
 RecipesDataSource: TDataSource
 DataSet = RecipesQuery
 ComponentsDataSource: TDataSource
 DataSet = ComponentsQuery
 CookBookMenu: TMainMenu

Создание элементов меню:

Группа	Пункт (Name)	Подпись (Caption)
Файл	OpenDBItem	Открыть БД
	CloseDBItem	Закрыть БД
	ExitItem	Выход
Таблицы	NewRecipeItem	Новый рецепт
	EditRecipeItem	Изменить рецепт
	AddFoodItem	Добавить продукт к рецепту
	RemoveFoodItem	Убрать продукт из рецепта
	EditQuantityItem	Изменить количество продукта
	DeleteRecipeItem	Удалить рецепт
	NewFoodItem	Новый продукт
	EditFoodItem	Изменить продукт
	DeleteFoodItem	Удалить продукт
Сервис	ListCalcItem	Расчет сметы

3.2. На форме RecipeForm:

Label1: TLabel
 Caption = Название:
 Label2: TLabel
 Caption = Персон:
 Label3: TLabel
 Caption = Категория:
 NameEdit: TEdit
 PersonsEdit: TEdit
 CategoryCombo: TComboBox
 OKButton: TButton
 Caption = ОК
 CancelButton: TButton

Caption = Отмена

3.3. На форме FoodForm:

Label1: TLabel

Caption = Название:

Label2: TLabel

Caption = Калорийность:

Label3: TLabel

Caption = Цена

NameEdit: TEdit

EnergyEdit: TEdit

PriceEdit: TEdit

OKButton: TButton

Caption = ОК

CancelButton: TButton

Caption = Отмена

3.4. На форме ComponentForm:

Label1: TLabel

Caption = Количество продукта:

QuantityEdit: TEdit

OKButton: TButton

Caption = ОК

CancelButton: TButton

Caption = Отмена

3.5. На форме ListCalcForm

Label1: TLabel

Caption = Количество персон:

PersonsEdit: TEdit

LCGrid: TStringGrid

ColCount = 2

FixedCols = 0

UpdateButton: TButton

Caption = Обновить

OKButton: TButton

Caption = Закрыть

4. Создать обработчики событий для компонентов формы, определить глобальные переменные, подключить необходимые библиотеки. Тексты модулей для форм приводятся в нижеследующих листингах.

4.1. Листинг модуля MainUnit:

```
unit MainUnit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, ExtCtrls, Grids, DBGrids, Menus, DB, ADODB, FoodUnit,
  RecipeUnit,
  ComponentUnit, ListCalcUnit;

type
  TMainForm = class(TForm)
    FoodGrid: TDBGrid;
    Splitter1: TSplitter;
    RecipesGrid: TDBGrid;
    Splitter2: TSplitter;
    ComponentsGrid: TDBGrid;
    CookBookMenu: TMainMenu;
    CookBookConnection: TADOConnection;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    OpenDBItem: TMenuItem;
    CloseDBItem: TMenuItem;
    N5: TMenuItem;
    ExitItem: TMenuItem;
    NewRecipeItem: TMenuItem;
    EditRecipeItem: TMenuItem;
    NewFoodItem: TMenuItem;
    EditFoodItem: TMenuItem;
    N4: TMenuItem;
    ListCalcItem: TMenuItem;
    OpenDBDialog: TOpenDialog;
    FoodTable: TADOTable;
    RecipesQuery: TADOQuery;
    RecipesDataSource: TDataSource;
    ComponentsDataSource: TDataSource;
    FoodDataSource: TDataSource;
    RFLinksTable: TADOTable;
```

```
ComponentsQuery: TADOQuery;
AddFoodItem: TMenuItem;
RemoveFoodItem: TMenuItem;
EditQuantityItem: TMenuItem;
DeleteFoodItem: TMenuItem;
DeleteRecipeItem: TMenuItem;
CategoryTable: TADOTable;
RecipesTable: TADOTable;
procedure OpenDBItemClick(Sender: TObject);
procedure ComponentsSync(RID: Integer);
procedure RecipesQueryAfterScroll(DataSet: TDataSet);
procedure CloseDBItemClick(Sender: TObject);
procedure NewFoodItemClick(Sender: TObject);
procedure EditFoodItemClick(Sender: TObject);
procedure DeleteFoodItemClick(Sender: TObject);
procedure NewRecipeItemClick(Sender: TObject);
procedure EditRecipeItemClick(Sender: TObject);
procedure DeleteRecipeItemClick(Sender: TObject);
procedure AddFoodItemClick(Sender: TObject);
procedure RemoveFoodItemClick(Sender: TObject);
procedure EditQuantityItemClick(Sender: TObject);
procedure ListCalcItemClick(Sender: TObject);
procedure ExitItemClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  MainForm: TMainForm;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TMainForm.ComponentsSync(RID: Integer);
begin
  with ComponentsQuery do
  begin
    Close;
    SQL.Clear;
```

```
SQL.Add('select RF_Links.ID, F_Name, Quantity');
SQL.Add('from Food, Recipes, RF_Links ');
SQL.Add('where Recipe_ID='+IntToStr(RID));
SQL.Add('and Food_ID=Food.ID');
SQL.Add('and Recipe_ID=Recipes.ID');
SQL.Add('order by Quantity desc');
Open;
end;
end;
```

```
procedure TMainForm.OpenDBItemClick(Sender: TObject);
begin
if (OpenDBDialog.Execute) then
begin
CookBookConnection.ConnectionString:=
'Provider=Microsoft.Jet.OLEDB.4.0;Data Source='+
OpenDBDialog.FileName;
if CookBookConnection.Connected then
begin
ShowMessage('База уже открыта');
Exit;
end;
CookBookConnection.Open;
RecipesQuery.Open;
RFLinksTable.Open;
FoodTable.Open;
CategoryTable.Open;
RecipesTable.Open;
ComponentsSync(RecipesQuery.FieldValues['id']);
end;
end;
```

```
procedure TMainForm.RecipesQueryAfterScroll(DataSet: TDataSet);
begin
if RFLinksTable.Active and ComponentsQuery.Active then
ComponentsSync(RecipesQuery.FieldValues['id']);
end;
```

```
procedure TMainForm.CloseDBItemClick(Sender: TObject);
begin
CookBookConnection.Close;
end;
```

```
procedure TMainForm.NewFoodItemClick(Sender: TObject);
begin
  with FoodForm do
  begin
    Caption:='Новый продукт;
    NameEdit.Clear;
    EnergyEdit.Clear;
    PriceEdit.Clear;
    if ShowModal=mrOK then
      with FoodTable do
      begin
        Append;
        FieldValues['F_Name']:=NameEdit.Text;
        FieldValues['Energy']:=StrToFloat(EnergyEdit.Text);
        FieldValues['Price']:=StrToFloat(PriceEdit.Text);
        Post;
      end;
    end;
  end;
end;
```

```
procedure TMainForm.EditFoodItemClick(Sender: TObject);
begin
  with FoodForm do
  begin
    Caption:='Изменение продукта';
    NameEdit.Text:=FoodTable.FieldValues['F_Name'];
    EnergyEdit.Text:=FoodTable.FieldValues['Energy'];
    PriceEdit.Text:=FoodTable.FieldValues['Price'];
    if ShowModal=mrOK then
      with FoodTable do
      begin
        Edit;
        FieldValues['F_Name']:=NameEdit.Text;
        FieldValues['Energy']:=StrToFloat(EnergyEdit.Text);
        FieldValues['Price']:=StrToFloat(PriceEdit.Text);
        Post;
      end;
    end;
  end;
end;
```

```
procedure TMainForm.DeleteFoodItemClick(Sender: TObject);
begin
```



```
if (MessageBox(Handle, 'Удалить продукт?',  
  'Удаление продукта', MB_YESNO)=IDYES) then  
  FoodTable.Delete;  
end;
```

```
procedure TMainForm.NewRecipeItemClick(Sender: TObject);  
var i: Integer;  
    cids: array of Integer;  
begin  
  with RecipeForm do  
    begin  
      Caption:='Новый рецепт';  
      NameEdit.Clear;  
      PersonsEdit.Clear;  
      CategoryCombo.Clear;  
      CategoryTable.First;  
      SetLength(cids, CategoryTable.RecordCount);  
      for i:=0 to CategoryTable.RecordCount-1 do  
        begin  
          CategoryCombo.Items.Add(CategoryTable.FieldValues['C_Name']);  
          cids[i]:=CategoryTable.FieldValues['ID'];  
          CategoryTable.Next;  
        end;  
      if ShowModal=mrOK then  
        with RecipesTable do  
          begin  
            Append;  
            FieldValues['R_Name']:=NameEdit.Text;  
            FieldValues['Persons']:=StrToFloat(PersonsEdit.Text);  
            FieldValues['Category_ID']:=cids[CategoryCombo.ItemIndex];  
            Post;  
          end;  
        end;  
      RecipesQuery.Requery;  
      Finalize(cids);  
    end;  
end;
```

```
procedure TMainForm.EditRecipeItemClick(Sender: TObject);  
var i: Integer;  
    cids: array of Integer;  
begin  
  with RecipeForm do
```

```

begin
  Caption:='Изменение рецепта';
  NameEdit.Text:=RecipesQuery.FieldValues['R_Name'];
  PersonsEdit.Text:=RecipesQuery.FieldValues['Persons'];
  CategoryCombo.Clear;
  CategoryTable.First;
  SetLength(cids, CategoryTable.RecordCount);
  for i:=0 to CategoryTable.RecordCount-1 do
  begin
    CategoryCombo.Items.Add(CategoryTable.FieldValues['C_Name']);
    cids[i]:=CategoryTable.FieldValues['ID'];
    if (CategoryTable.FieldValues['ID']=
      RecipesQuery.FieldValues['Category_ID']) then
      CategoryCombo.ItemIndex:=i;
    CategoryTable.Next;
  end;
  if ShowModal=mrOK then
    with RecipesTable do
    begin
      Locate('ID', RecipesQuery.FieldValues['ID'], []);
      Edit;
      FieldValues['R_Name']:=NameEdit.Text;
      FieldValues['Persons']:=StrToFloat(PersonsEdit.Text);
      FieldValues['Category_ID']:=cids[CategoryCombo.ItemIndex];
      Post;
    end;
  end;
  RecipesQuery.Requery;
  Finalize(cids);
end;

```

```

procedure TMainForm.DeleteRecipeItemClick(Sender: TObject);
begin
  RecipesTable.Locate('ID', RecipesQuery.FieldValues['ID'], []);
  RecipesTable.Delete;
  RecipesQuery.Requery;
end;

```

```

procedure TMainForm.AddFoodItemClick(Sender: TObject);
begin
  with ComponentForm do
  begin

```

```

Caption:='Добавление продукта';
QuantityEdit.Clear;
if ShowModal=mrOK then
begin
  with RFLinksTable do
  begin
    Append;
    FieldValues['Recipe_ID']:=RecipesQuery.FieldValues['ID'];
    FieldValues['Food_ID']:=FoodTable.FieldValues['ID'];
    FieldValues['Quantity']:=QuantityEdit.Text;
    Post;
  end;
end;
end;
ComponentsQuery.Requery;
end;

procedure TMainForm.RemoveFoodItemClick(Sender: TObject);
begin
  RFLinksTable.Locate('ID', ComponentsQuery.FieldValues['ID'], []);
  RFLinksTable.Delete;
  ComponentsQuery.Requery;
end;

procedure TMainForm.EditQuantityItemClick(Sender: TObject);
begin
  with ComponentForm do
  begin
    Caption:='Изменение количества';
    QuantityEdit.Text:=ComponentsQuery.FieldValues['Quantity'];
    if ShowModal=mrOK then
    begin
      with RFLinksTable do
      begin
        Locate('ID', ComponentsQuery.FieldValues['ID'], []);
        Edit;
        FieldValues['Recipe_ID']:=RecipesQuery.FieldValues['ID'];
        FieldValues['Food_ID']:=FoodTable.FieldValues['ID'];
        FieldValues['Quantity']:=QuantityEdit.Text;
        Post;
      end;
    end;
  end;
end;

```

```

end;
ComponentsQuery.Requery;
end;

procedure TMainForm.ListCalcItemClick(Sender: TObject);
var i: Integer;
begin
  with ComponentsQuery do
  begin
    lcp:=RecipesQuery.FieldValues['Persons'];
    SetLength(lc, RecordCount);
    First;
    ListCalcForm.Caption:='Смета на '+IntToStr(lcp)+' персон';
    ListCalcForm.LCGrid.ColWidths[0]:=200;
    ListCalcForm.LCGrid.ColWidths[1]:=100;
    ListCalcForm.LCGrid.RowCount:=lcp+1;
    ListCalcForm.LCGrid.Cells[0, 0]:='Продукт';
    ListCalcForm.LCGrid.Cells[1, 0]:='Количество';
    for i:=0 to RecordCount-1 do
    begin
      ListCalcForm.LCGrid.Cells[0, i+1]:=FieldValues['F_Name'];
      ListCalcForm.LCGrid.Cells[1, i+1]:=IntToStr(FieldValues['Quantity']);
      lc[i]:=FieldValues['Quantity'];
    end;
  end;
  ListCalcForm.ShowModal;
  Finalize(lc);
end;

procedure TMainForm.ExitItemClick(Sender: TObject);
begin
  Close;
end;

end.

```

4.2. Листинг модуля FoodUnit:

```

unit FoodUnit;

interface

```

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
Dialogs, StdCtrls;

type
TFoodForm = class(TForm)
NameEdit: TEdit;
Label1: TLabel;
PriceEdit: TEdit;
Label2: TLabel;
EnergyEdit: TEdit;
Label3: TLabel;
CancelButton: TButton;
OKButton: TButton;
procedure OKButtonClick(Sender: TObject);
procedure CancelButtonClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
FoodForm: TFoodForm;

implementation

{ \$R *.dfm }

procedure TFoodForm.OKButtonClick(Sender: TObject);
begin
ModalResult:=mrOK;
end;

procedure TFoodForm.CancelButtonClick(Sender: TObject);
begin
ModalResult:=mrCancel;
end;

end.

4.3. Листинг модуля RecipeUnit:

```
unit RecipeUnit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls;

type
  TRecipeForm = class(TForm)
    NameEdit: TEdit;
    Label1: TLabel;
    PersonsEdit: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    CategoryCombo: TComboBox;
    CancelButton: TButton;
    OKButton: TButton;
    procedure CancelButtonClick(Sender: TObject);
    procedure OKButtonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  RecipeForm: TRecipeForm;

implementation

{$R *.dfm}

procedure TRecipeForm.CancelButtonClick(Sender: TObject);
begin
  ModalResult:=mrCancel;
end;
procedure TRecipeForm.OKButtonClick(Sender: TObject);
```

```
begin
  ModalResult:=mrOK;
end;
```

```
end.
```

4.4. Листинг модуля ComponentUnit:

```
unit ComponentUnit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls;

type
  TComponentForm = class(TForm)
    QuantityEdit: TEdit;
    Label1: TLabel;
    CancelButton: TButton;
    OKButton: TButton;
    procedure OKButtonClick(Sender: TObject);
    procedure CancelButtonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  ComponentForm: TComponentForm;

implementation

{$R *.dfm}

procedure TComponentForm.OKButtonClick(Sender: TObject);
begin
  ModalResult:=mrOK;
end;
```

```
procedure TComponentForm.CancelButtonClick(Sender: TObject);
begin
  ModalResult:=mrCancel;
end;

end.
```

1.1. ЛИСТИНГ МОДУЛЯ ListCalcUnit:

```
unit ListCalcUnit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, Grids, Math;

type
  TListCalcForm = class(TForm)
    OKButton: TButton;
    Label1: TLabel;
    PersonsEdit: TEdit;
    UpdateButton: TButton;
    LCGrid: TStringGrid;
    procedure OKButtonClick(Sender: TObject);
    procedure UpdateButtonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  ListCalcForm: TListCalcForm;
  lc: array of Real;
  lcp: Integer;

implementation

{$R *.dfm}
```



```

procedure TListCalcForm.OKButtonClick(Sender: TObject);
begin
  ModalResult:=mrOK;
end;

procedure TListCalcForm.UpdateButtonClick(Sender: TObject);
var c: Real;
    i: Integer;
begin
  c:=StrToInt(PersonsEdit.Text)/lcp;
  LCGrid.RowCount:=Length(lc)+1;
  for i:=0 to Length(lc)-1 do
  begin
    lc[i]:=RoundTo(lc[i]*c, -2);
    LCGrid.Cells[1, i+1]:=FloatToStr(lc[i]);
  end;
  lcp:=StrToInt(PersonsEdit.Text);
end;

end.

```

Содержание отчета

Программа с учетом основных рекомендаций.

Вопросы для самопроверки

1. Опишите схему получения клиентским приложением данных из БД с помощью технологии ADO.
2. Какие способы доступа к полям таблицы посредством компонента TADOTable вы знаете?
3. Что произойдет при удалении записи из таблицы продуктов, если ссылка на этот продукт присутствует в связанной таблице рецептов?
4. Зачем необходимо использовать метод Post при записи в таблицу?
5. В чем состоит необходимость использования метода Requery в тексте листингов?