

Министерство образования и науки Российской Федерации

Университетский колледж
федерального государственного бюджетного образовательного учреждения
высшего образования
«Оренбургский государственный университет»

Е.К. Канивец

ЛЕКЦИИ ПО ИНФОРМАТИКЕ

Учебное пособие

Рекомендовано учёным советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по программам среднего профессионального образования.

Оренбург

2018

УДК 007(075.32)

ББК 32.97Я723

Л19

Рецензент – кандидат технических наук, доцент И.А. Щудро

Канивец, Е.К.

Л19 Лекции по информатике: учебное пособие / сост. Е.К. Канивец; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2018. –180 с.

В учебном пособии рассмотрены такие разделы дисциплины «Информатика», как информационная деятельность человека, информация и информационные процессы, средства информационных и коммуникационных технологий, технологии создания и преобразования информационных объектов и телекоммуникационные технологии.

Учебное пособие предназначено для обучающихся первого курса очного отделения по программам среднего профессионального образования при изучении дисциплины «Информатика» и обеспечивает образовательный процесс в Университетском колледже ОГУ.

УДК 007(075.32)

ББК 32.97Я723

©Канивец Е.К., 2018

©ОГУ, 2018

Содержание

Введение.....	6
1 Лекция 1. Введение. Представление текста, изображения и звука в компьютере	7
1.1 Введение.....	7
1.2 Текстовая информация	7
1.3 Графическая информация.....	8
1.4 Звуковая информация	12
2 Лекция 2. Измерение информации. Алфавитный и содержательный подходы. Представление чисел	14
2.1 Алфавитный подход при измерении информации	14
2.2 Содержательный подход при измерении информации.....	17
2.3 Формула Хартли.....	21
2.4 Представление чисел в компьютере.....	23
3 Лекция 3. Интерфейс MS Word 2007. Работа с файлами в MS Word	29
3.1 Интерфейс Microsoft Word 2007	29
3.2 Работа с файлами в Microsoft Word 2007.....	35
4 Лекция 4. Работа с документом в MS Word. Списки в MS Word	43
4.1 Работа с документом в Microsoft Word 2007.....	43
4.2 Списки в MS Word 2007	50
5 Лекция 5. Создание и изменение таблиц в MS Word	55
5.1 Создание таблиц.....	55
5.2 Работа с таблицами в Microsoft Word 2007	62
6.Лекция 6.Обработка информации и алгоритмы.....	68
6.1 Алгоритмы и величины	68
6.2 Структура алгоритмов	72
7 Лекция 7. Pascal — язык структурного программирования	75
7.1 История Паскаля	75
7.2 Структура процедурных языков программирования высокого уровня	76

7.3 Структура программы на Паскале.....	77
8 Лекция 8. Операции, функции, выражения языка Pascal.....	81
8.1. Операции, функции, выражения.....	81
8.2 Стандартные функции и процедуры	82
8.3 Оператор присваивания, ввод и вывод данных	84
8.4 Линейная программа.....	87
9 Лекция 9. Логические величины, операции, выражения языка Pascal.....	89
9.1 Логические операции.....	89
9.2 Логические функции на области числовых значений.....	91
9.3 Логические выражения языка Паскаль	93
10 Лекция 10. Программирование ветвлений	95
10.1 Полные и неполные ветвления	95
10.2 Пример поэтапной разработки программы решения задачи.....	98
11 Лекция 11. Создание презентаций с использованием программы MS PowerPoint	103
11.1 Знакомство с рабочей областью MS PowerPoint.....	103
11.2 Присвоение имени и сохранение презентации	105
11.3 Добавление, изменение порядка и удаление слайдов	106
11.4 Добавление и форматирование текста.....	109
12 Лекция 12. Содержательные и дизайн-эргономические требования к презентациям	116
12.1 Содержательные требования к презентации.....	116
12.2 Дизайн-эргономические требования к презентации	117
12.3 Цветовой круг	119
13 Лекция 13. Начало работы с MS Access. Создание базы данных	120
13.1 Объекты Microsoft Access.....	120
13.2 Начало работы с Access 2007.....	123
13.3 Пример создания базы данных (структуры таблиц) в Access 2007	125
14 Лекция 14. Проектирование многотабличной базы данных.....	128
14.1 Установка логических связей в БД Access 2007	129

14.2 Заполнение таблиц БД Access 2007.....	131
14.3 Создание запросов и поиск информации в базе данных.....	134
14.4 Создание и использование форм для ввода данных в таблицы базы данных MS Access 2007.....	138
14.5 Отчеты в базе данных Access 2007.....	144
15 Лекция 15. Организация глобальных сетей.....	148
15.1 История развития глобальных сетей.....	148
15.2 Аппаратные средства Интернета.....	151
15.3 Каналы связи.....	152
15.4 Программное обеспечение Интернета.....	153
15.5 Как работает Интернет.....	154
16 Лекция 16. Поиск информации с использованием компьютера. Электронная почта.....	156
16.1 Поиск информации с использованием компьютера.....	156
16.2 Использование подстановочных знаков.....	160
16.3 Организация специального поиска.....	163
16.4 Электронная почта.....	165
17 Лекция 17. Интернет как глобальная информационная система. Инструменты по созданию сайтов.....	166
17.1 Назначение Интернета.....	166
17.2 Коммуникационные службы Интернета.....	167
17.3 Информационные службы Интернета.....	170
17.4 Web-2-сервисы.....	171
17.5 Инструменты по созданию сайтов.....	171
17.6 Основы создания Web-страниц средствами языка разметки гипертекста HTML.....	175
17.7 Создание документа HTML в текстовом редакторе Блокнот.....	178
Список использованных источников.....	180

Введение

Основными составными частями учебного процесса в преподавании дисциплины «Информатика» обучающимся дневной формы обучения являются лекции и лабораторные работы. Очевидно, что обучающиеся, систематически изучающие лекционный материал, способны успешнее освоить предмет. Все лекции необходимо конспектировать. В конспект рекомендуется выписывать определения, формулировки примеры и т.п.

Изучение и изложение информации, полученной в результате изучения дополнительной научной и теоретической литературы, предполагает развитие у обучающихся владения навыками устной речи и способностей к четкому письменному изложению материала.

В результате изучения учебной дисциплины «Информатика» обучающийся должен знать:

- понятие информации, виды и свойства информации;
- способы поиска информации с использованием возможностей прикладных программ и современных поисковых систем;
- технологию сбора, хранения, передачи, обработки и предоставления информации;
- основные источники информации и ресурсы для решения проблем в профессиональном и/или социальном контексте.

- применять различные способы поиска информации
- современные средства и устройства информатизации
- порядок их применения и программное обеспечение в профессиональной деятельности;

уметь:

- применять различные способы поиска информации
- применять средства информатизации и информационных технологий для решения профессиональных задач.

Курс лекций рассчитан на 34 часа.

1 Лекция 1. Введение. Представление текста, изображения и звука в компьютере

1.1 Введение

Информатика - молодая научная дисциплина, изучающая вопросы, связанные с поиском, сбором, хранением, преобразованием и использованием информации в самых различных сферах человеческой деятельности.

Фундаментом для изучения дисциплины «Информатика» на 1 курсе являются знания и умения, которые вы получили, изучая курс информатики в основной школе — базовый курс информатики. Вам уже не требуется объяснять, что такое компьютер, и как он работает; с какой информацией может работать компьютер; что такое программа и программное обеспечение компьютера; что такое информационные технологии. В базовом курсе информатики вы получили представление о том, в каком виде хранится информация в памяти компьютера, что такое алгоритм, информационная модель. Осваивая базовый курс, вы научились обращаться с клавиатурой, мышью, дисками, принтером; работать в среде операционной системы. Вы получили основные навыки работы с текстовыми и графическими редакторами, с базами данных и электронными таблицами. Все эти знания и навыки вам будут необходимы при изучении курса «Информатика» на 1 курсе.

1.2 Текстовая информация

Принципиально важно, что текстовая информация уже дискретна — состоит из отдельных знаков. Поэтому возникает лишь технический вопрос: как разместить её в памяти компьютера?

Модель представления текста в памяти весьма проста. За каждой буквой алфавита, цифрой, знаком препинания и иным общепринятым при записи текста символом закрепляется определенный двоичный код, длина которого фиксирована. В популярных системах кодировки (ASCII, KOI8 и др.) каждый сим-

вол заменяется на 8-разрядное целое положительное двоичное число; оно хранится в одном байте памяти. Это число является порядковым номером символа в кодовой таблице. Согласно главной формуле информатики, определяем, что размер алфавита, который можно закодировать, равен: $2^8 = 256$. Этого количества вполне достаточно для размещения двух алфавитов естественных языков (английского и русского) и всех необходимых дополнительных символов.

Поскольку в мире много языков и много алфавитов, то постепенно совершается переход на международную 16-битовую систему кодировки Unicode. В ней каждый символ занимает 2 байта, что обеспечивает $2^{16} = 65\,536$ кодов для различных символов. При работе с электронной почтой почтовая программа иногда нас спрашивает, не хотим ли мы прибегнуть к кодировке Unicode для пересылаемых сообщений. Таким способом можно избежать проблемы несоответствия кодировок, из-за которой иногда не удается прочитать русский текст.

Не следует представлять себе текст, хранимый в памяти компьютера или на внешнем носителе, лишь как поток байтов, каждый из которых является лишь кодом символа текста. Форматы хранения текстовой информации определяются форматами текстовых файлов, используемых той или иной программой обработки текстов. Файлы, создаваемые с помощью текстовых процессоров (например, Microsoft Word), включают в себя не только коды символов алфавита, но и данные формата: тип и размер шрифта, положение строк, поля, отступы и прочую дополнительную информацию.

1.3 Графическая информация

Принцип дискретности компьютерных данных справедлив и для графики. Здесь можно говорить о дискретном представлении изображения (рисунка, фотографии, видеокадров) и дискретности цвета.

Дискретное представление изображения.

Изображение на экране монитора дискретно. Оно составляется из отдельных точек, которые называются пикселями (picture elements — элементы ри-

сунка). Это связано с техническими особенностями устройства экрана, независимо от его физической реализации, будь то традиционный дисплей на электронно-лучевой трубке, жидкокристаллический или плазменный. Эти «точки» столь близки друг другу, что глаз не различает промежутков между ними, поэтому изображение воспринимается как непрерывное, сплошное. Если выводимое из компьютера изображение формируется на бумаге (принтером или плоттером), то линии на нем также выглядят непрерывными. Однако в основе все равно лежит печать близких друг к другу точек.

В зависимости от того, на какое графическое разрешение экрана настроена операционная система компьютера, на экране могут размещаться изображения, имеющие размер 640x480, 800x600, 1024x768 и более пикселей. Такая прямоугольная матрица пикселей на экране компьютера называется растром. Качество изображения зависит не только от размера раstra, но и от размера экрана монитора, который обычно характеризуется длиной диагонали. Существует параметр разрешения экрана. Этот параметр измеряется в точках на дюйм (по-английски *dotsperinch* — *dpi*). У монитора с диагональю 15 дюймов размер изображения на экране составляет примерно 28x21 см². Зная, что в одном дюйме 25,4 мм, можно рассчитать, что при работе монитора в режиме 800x600 пикселей разрешение экранного изображения равно 72 dpi.

При печати на бумаге разрешение должно быть намного выше. Полиграфическая печать полноцветного изображения требует разрешения 200-300 dpi. Стандартный фотоснимок размером 10x15 см² должен содержать примерно 1000x1500 пикселей.

Дискретное представление цвета.

Восстановим ваши знания о кодировании цвета, полученные из базового курса информатики. Основное правило звучит так: любой цвет точки на экране компьютера получается путем смешивания трех базовых цветов: красного, зеленого, синего. Этот принцип называется цветовой моделью RGB (Red, Green, Blue).

Двоичный код цвета определяет, в каком соотношении находятся интенсивности трех базовых цветов. Если все они смешиваются в одинаковых долях, то в итоге получается белый цвет. Если все три компонента «выключены», то цвет пикселя — черный. Все остальные цвета лежат между белым и черным.

Дискретность цвета состоит в том, что интенсивности базовых цветов могут принимать конечное число дискретных значений.

Пусть, например, размер кода цвета пикселя равен 8 битам — 1 байту. Между базовыми цветами они могут быть распределены так:

К К 3 3 3 С С С
 2 бита под красный цвет, 3 бита — под зеленый и 3 бита — под синий.

Интенсивность красного цвета может принимать $2^2 = 4$ значения, интенсивности зеленого и синего цветов — по $2^3 = 8$ значений. Полное число цветов, которые кодируются 8-разрядными кодами, равно: $4 \times 8 \times 8 = 256 = 2^8$. Снова работает главная формула информатики.

Из описанного правила, в частности, следует:

	красный			зеленый			синий			
0	0	0	0	0	0	0	0	0	0	– код черного цвета
1	1	1	1	1	1	1	1	1	1	– код белого цвета

Обобщение этих частных примеров приводит к следующему правилу. Если размер кода цвета равен b битов, то количество цветов (размер палитры) вычисляется по формуле:

$$K = 2^b, \quad (1)$$

где: b – битовая глубина цвета.

Еще один пример. Битовая глубина цвета равна 24. Размер палитры будет равен: $K = 2^{24} = 16\,777\,216$

В компьютерной графике используются разные цветовые модели для изображения на экране, получаемого путем излучения света, и изображения на бумаге, формируемого с помощью отражения света. Первую модель мы уже рассмотрели — это модель RGB. Вторая модель носит название CMYK.

Цвет, который мы видим на листе бумаги, — это отражение белого (солнечного) света. Нанесенная на бумагу краска поглощает часть палитры, составляющей белый цвет, а другую часть отражает. Таким образом, нужный цвет на бумаге получают путем «вычитания» из белого цвета «ненужных красок». Поэтому в цветной полиграфии действует не правило сложения цветов (как на экране компьютера), а правило вычитания. Мы не будем углубляться в механизм такого способа цветообразования. Расшифруем лишь аббревиатуру СМΥК: Cyan — голубой, Magenta — пурпурный, Yellow — желтый, Black — черный.

Растровая и векторная графика. В растровой графике графическая информация — это совокупность данных о цвете каждого пикселя на экране. Это то, о чем говорилось выше. В векторной графике графическая информация — это данные, математически описывающие графические примитивы, составляющие рисунок: прямые, дуги, прямоугольники, овалы и пр. Положение и форма графических примитивов представляются в системе экранных координат. Растровую графику (редакторы растрового типа) применяют при разработке электронных (мультимедийных) и полиграфических изданий. Растровые иллюстрации редко создают вручную с помощью компьютерных программ. Чаще для этой цели используют сканированные иллюстрации, подготовленные художником на бумаге, или фотографии. Для ввода растровых изображений в компьютер применяются цифровые фото- и видеокамеры. Большинство графических редакторов растрового типа в большей мере ориентированы не на создание изображений, а на их обработку. Достоинство растровой графики — эффективное представление изображений фотографического качества. Основной недостаток растрового способа представления изображения — большой объем занимаемой памяти. Для его сокращения приходится применять различные способы сжатия данных. Другой недостаток растровых изображений связан с искажением изображения при его масштабировании. Поскольку изображение состоит из фиксированного числа точек, то увеличение изображения приводит к тому, что эти точки становятся крупнее. Увеличение размера точек раstra визуально искажает иллюстрацию и делает ее грубой.

Векторные графические редакторы предназначены, в первую очередь, для создания иллюстраций и в меньшей степени для их обработки. Достоинства векторной графики — сравнительно небольшой объем памяти, занимаемой векторными файлами, масштабирование изображения без потери качества. Однако средствами векторной графики проблематично получить высококачественное художественное изображение. Обычно средства векторной графики используют не для создания художественных композиций, а для оформительских, чертежных и проектно-конструкторских работ.

Графическая информация сохраняется в файлах на диске. Существуют разнообразные форматы графических файлов. Они делятся на растровые и векторные. Растровые графические файлы (форматы JPEG, BMP, TIFF и другие) хранят информацию о цвете каждого пикселя изображения на экране. В графических файлах векторного формата (например, WMF, CJM) содержатся описания графических примитивов, составляющих рисунок. Следует понимать, что графические данные, помещаемые в видеопамять и выводимые на экран, имеют растровый формат вне зависимости от того, с помощью каких программных средств (растровых или векторных) они получены.

1.4 Звуковая информация

Принципы дискретизации звука («оцифровки» звука) отражены на рисунке 1 (у — интенсивность (уровень) звукового сигнала, t — время).

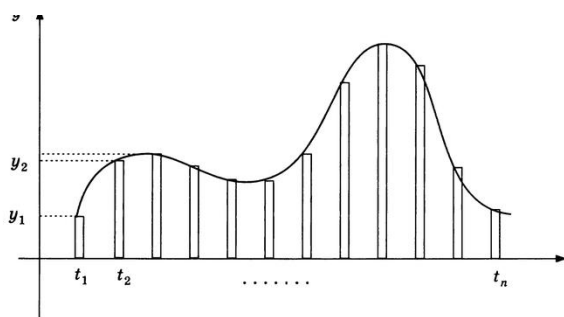


Рисунок 1 – Оцифровка звука

Ввод звука в компьютер производится с помощью звукового устройства (микрофона, радио и др.), выход которого подключается к порту звуковой кар-

ты. Задача звуковой карты — с определенной частотой производить измерения уровня звукового сигнала (преобразованного в электрические колебания) и результаты измерения записывать в память компьютера. Этот процесс называют оцифровкой звука.

Промежуток времени между двумя измерениями называется периодом измерений — t (с). Обратная величина называется частотой дискретизации — $1/t$ (Герц). Чем выше частота измерений, тем выше качество цифрового звука.

Результаты таких измерений представляются целыми положительными числами с конечным количеством разрядов. Вы уже знаете, что в таком случае получается дискретное конечное множество значений в ограниченном диапазоне. Размер этого диапазона зависит от разрядности ячейки — регистра памяти звуковой карты. Здесь работает формула $N=2^i$ (где i – разрядность регистра или разрядностью дискретизации, N – количество уровней измерения звука). Записанные данные сохраняются в файлах специальных звуковых форматов.

Существуют программы обработки звука — редакторы звука, позволяющие создавать различные музыкальные эффекты, очищать звук от шумов, согласовывать с изображениями для создания мультимедиа продуктов и т. д. С помощью специальных устройств, генерирующих звук, звуковые файлы могут преобразовываться в звуковые волны, воспринимаемые слухом человека. При хранении оцифрованного звука приходится решать проблему уменьшения объема звуковых файлов. Для этого кроме кодирования данных без потерь, позволяющего осуществлять стопроцентное восстановление данных из сжатого потока, используется кодирование данных с потерями. Цель такого кодирования — добиться схожести звучания восстановленного сигнала с оригиналом при максимальном сжатии данных. Это достигается путем использования различных алгоритмов, сжимающих оригинальный сигнал путем выкидывания из него слабо слышимых элементов. Методов сжатия, а также программ, реализующих эти методы, существует много.

2 Лекция 2. Измерение информации. Алфавитный и содержательный подходы. Представление чисел

2.1 Алфавитный подход при измерении информации

Вопрос об измерении количества информации является очень важным как для науки, так и для практики. В самом деле, если информация является предметом нашей деятельности, мы ее храним, передаем, принимаем, обрабатываем. Поэтому важно договориться о способе ее измерения, позволяющем, например, ответить на вопросы: достаточно ли места на носителе, чтобы разместить нужную нам информацию, или сколько времени потребуется, чтобы передать ее по имеющемуся каналу связи. Величина, которая нас в этих ситуациях интересует, называется объемом информации. В таком случае говорят об алфавитном, или объемном, подходе к измерению информации.

Алфавитный подход к измерению информации применяется в цифровых (компьютерных) системах хранения и передачи информации. В этих системах используется двоичный способ кодирования информации. При алфавитном подходе для определения количества информации имеет значение лишь размер (объем) хранимого и передаваемого кода. Алфавитный подход еще называют объемным подходом. Если с помощью i -разрядного двоичного кода можно закодировать алфавит, состоящий из N символов (где N — целая степень двойки), то эти величины связаны между собой по формуле:

$$2^i = N, \quad (2)$$

где N — число символов;

i — разрядность двоичного кода.

Если, например, $i = 2$, то можно построить 4 двухразрядные комбинации из нулей и единиц, т. е. закодировать 4 символа. При $i = 3$ существует 8 трехразрядных комбинаций нулей и единиц (кодируется 8 символов):

При $i=2$ (00, 01, 10, 11);

при $i=3$ (000, 001, 010, 011, 100, 101, 110, 111).

Английский алфавит содержит 26 букв. Для записи текста нужны еще как минимум шесть символов: пробел, точка, запятая, вопросительный знак, восклицательный знак, тире. В сумме получается расширенный алфавит мощностью в 32 символа. Поскольку $32 = 2^5$, все символы можно закодировать всевозможными пятиразрядными двоичными кодами от 00000 до 11111. Именно пятиразрядный код использовался в телеграфных аппаратах, появившихся еще в XIX веке. Телеграфный аппарат при вводе переводил английский текст в двоичный код, длина которого в 5 раз больше, чем длина исходного текста.

В двоичном коде каждая двоичная цифра несет одну единицу информации, которая называется 1 бит. Бит является основной единицей измерения информации. Длина двоичного кода, с помощью которого кодируется символ алфавита, называется информационным весом символа. В рассмотренном выше примере информационный вес символа расширенного английского алфавита оказался равным 5 битам. Информационный объем текста складывается из информационных весов всех составляющих текст символов. Например, английский текст из 1000 символов в телеграфном сообщении будет иметь информационный объем 5000 битов.

Алфавит русского языка включает 33 буквы. Если к нему добавить еще пробел и пять знаков препинания, то получится набор из 39 символов. Для двоичного кодирования символов такого алфавита пятиразрядного кода уже недостаточно. Нужен как минимум 6-разрядный код. Поскольку $2^6 = 64$, остается еще резерв для 25 символов ($64 - 39 = 25$). Его можно использовать для кодирования цифр, всевозможных скобок, знаков математических операций и других символов, встречающихся в русском тексте. Следовательно, информационный вес символа в расширенном русском алфавите будет равен 6 битам. А текст из 1000 символов будет иметь объем 6000 битов. Итак, информационный объем I текста выражается формулой:

$$I = K \times i \text{ (битов)}, \quad (3)$$

где i — информационный вес символа алфавита;

K — количество символов в тексте.

Идея измерения количества информации в сообщении через длину двоичного кода этого сообщения принадлежит выдающемуся российскому математику Андрею Николаевичу Колмогорову. Согласно Колмогорову, количество информации, содержащееся в тексте, определяется минимально возможной длиной двоичного кода, необходимого для представления этого текста.

Для определения информационного веса символа полезно знать ряд целых степеней двойки (2, 4, 8, 16, 32, 64, 128, 256 и т.д.).

Поскольку мощность N алфавита может не являться целой степенью двойки, информационный вес символа алфавита мощности N определяется следующим образом. Находится ближайшее к N значение, не меньшее чем N . Соответствующее значение i будет равно информационному весу символа.

Пример. Определим информационный вес символа алфавита, включающего в себя все строчные и прописные русские буквы (66); цифры (10); знаки препинания, скобки, кавычки (10). Всего получается 86 символов.

Поскольку $2^6 < 86 < 2^7$, информационный вес символов данного алфавита равен 7 битам. Это означает, что все 86 символов можно закодировать семиразрядными двоичными кодами.

Для двоичного представления текстов в компьютере чаще всего применяется восьмиразрядный код. С помощью восьмиразрядного кода можно закодировать алфавит из 256 символов, поскольку $256 = 2^8$. В стандартную кодовую таблицу (например, используемую в ОС Windows таблицу ANSI) помещаются все необходимые символы: английские и русские буквы — прописные и строчные, цифры, знаки препинания, знаки арифметических операций, всевозможные скобки и пр.

Более крупной, чем бит, единицей измерения информации является байт: 1 байт = 8 битов.

Информационный объем текста в памяти компьютера измеряется в байтах. Он равен количеству символов в записи текста.

Одна страница текста на листе формата А4 кегля 12 с одинарным интервалом между строками в компьютерном представлении будет иметь объем 4000 байтов, так как на ней помещается примерно 4000 знаков.

Помимо бита и байта, для измерения информации используются и более крупные единицы:

1 Кб (килобайт) = 2^{10} байтов = 1024 байта;

1 Мб (мегабайт) = 2^{10} Кб = 1024 Кб;

1 Гб (гигабайт) = 2^{10} Мб = 1024 Мб;

1 Тб (терабайт) = 2^{10} Гб = 1024 Гб.

Объем той же страницы текста будет равен приблизительно 3,9 Кб. А книга из 500 таких страниц займет в памяти компьютера примерно 1,9 Мб.

– В компьютере любые виды информации: тексты, числа, изображения, звуки — представляются в форме двоичного кода.

– Объем информации любого вида, выраженный в битах, равен длине двоичного кода, в котором эта информация представлена.

2.2 Содержательный подход при измерении информации

Неопределенность знания и количество информации. Содержательный подход к измерению информации отталкивается от определения информации как содержания сообщения, получаемого человеком. Сущность содержательного подхода заключается в следующем: сообщение, информирующее об исходе какого-то события, снимает неопределенность знания человека об этом событии. Чем больше первоначальная неопределенность знания, тем больше информации несет сообщение, снимающее эту неопределенность.

Приведем примеры, иллюстрирующие данное утверждение.

Ситуация 1. В ваш класс назначен новый учитель информатики; на вопрос «Это мужчина или женщина?» вам ответили: «Мужчина».

Ситуация 2. На чемпионате страны по футболу играли команды «Динамо» и «Зенит». Из спортивных новостей по радио вы узнаете, что игра закончилась победой «Зенита».

Ситуация 3. На выборах мэра города было представлено четыре кандидата. После подведения итогов голосования вы узнали, что избран Н. Н. Никитин.

Вопрос: в какой из трех ситуаций полученное сообщение несет больше информации?

– Неопределенность знания — это количество возможных вариантов ответа на интересовавший вас вопрос. Еще можно сказать: возможных исходов события. Здесь событие — выборы мэра; исход — выбор, например, Н. Н. Никитина.

В первой ситуации 2 варианта ответа: мужчина, женщина; во второй ситуации 3 варианта: выиграл «Зенит», ничья, выиграло «Динамо»; в третьей ситуации — 4 варианта: 4 кандидата на пост мэра.

Согласно данному выше определению, наибольшее количество информации несет сообщение в третьей ситуации, поскольку неопределенность знания об исходе события в этом случае была наибольшей.

В 40-х годах XX века проблема измерения информации была решена американским ученым Клодом Шенноном — основателем теории информации. Согласно Шеннону, информация — это снятая неопределенность знания человека об исходе какого-то события.

В теории информации единица измерения информации определяется следующим образом.

– Сообщение, уменьшающее неопределенность знания об исходе некоторого события в два раза, несет 1 бит информации. Согласно этому определению, сообщение в первой из описанных ситуаций несет 1 бит информации, поскольку из двух возможных вариантов ответа был выбран один. Следовательно, количество информации, полученное во второй и в третьей ситуациях, больше, чем один бит. Но как измерить это количество?

Рассмотрим еще один пример. Ученик написал контрольную работу по информатике и спрашивает учителя о полученной оценке. Оценка может оказаться любой: от 2 до 5. На что учитель отвечает: «Угадай оценку за два вопроса, ответом на которые может быть только «да» или «нет»». Подумав, ученик

задал первый вопрос: «Оценка выше тройки?». «Да», — ответил учитель. Второй вопрос: «Это пятерка?». «Нет», — ответил учитель. Ученик понял, что он получил четверку. Первоначально неопределенность знания (количество возможных оценок) была равна четырем. С ответом на каждый вопрос неопределенность знания уменьшалась в 2 раза и, следовательно, согласно данному выше определению, передавался 1 бит информации. Узнав оценку (одну из четырех возможных), ученик получил 2 бита информации.

Рассмотрим еще один частный пример, а затем выведем общее правило. Вы едете на электропоезде, в котором 8 вагонов, а на вокзале вас встречает товарищ. Товарищ позвонил вам по мобильному телефону и спросил, в каком вагоне вы едете. Вы предлагаете угадать номер вагона, задав наименьшее количество вопросов, ответами на которые могут быть только слова «да» или «нет».

Немного подумав, товарищ стал спрашивать:

- Номер вагона больше четырех? Да.
- Номер вагона больше шести? Нет.
- Это шестой вагон? Нет.
- Ты едешь в пятом вагоне!

Каждый ответ уменьшал неопределенность знания в два раза. Всего было задано три вопроса. Значит, в сумме набрано 3 бита информации. То есть сообщение о том, что вы едете в пятом вагоне, несет 3 бита информации. Способ решения проблемы, примененный в примерах с оценками и вагонами, называется методом половинного деления: ответ на каждый вопрос уменьшает неопределенность знания, имеющуюся перед ответом на этот вопрос, наполовину. Каждый такой ответ несет 1 бит информации. Заметим, что решение подобных проблем методом половинного деления наиболее рационально. Таким способом всегда можно угадать, например, любой из восьми вариантов за 3 вопроса. Если бы поиск производился последовательным перебором: «Ты едешь в первом вагоне?» «Нет», «Во втором вагоне?» «Нет» и т. д., то про пятый вагон вы смогли бы узнать после пяти вопросов, а про восьмой — после восьми.

«Главная формула» информатики. Сформулируем одно очень важное условие, относящееся к рассмотренным примерам. Во всех ситуациях предполагается, что все возможные исходы события равновероятны. Равновероятно, что учитель может быть мужчиной или женщиной; равновероятен любой исход футбольного матча, равновероятен выбор одного из четырех кандидатов в мэры города. То же относится и к примерам с оценками и вагонами.

Тогда полученные нами результаты описываются следующими формулировками:

- сообщение об одном из двух равновероятных исходов некоторого события несет 1 бит информации;
- сообщение об одном из четырех равновероятных исходов некоторого события несет 2 бита информации;
- сообщение об одном из восьми равновероятных исходов некоторого события несет 3 бита информации.

Обозначим буквой N количество возможных исходов события, или, как мы это еще называли, — неопределенность знания. Буквой i будем обозначать количество информации в сообщении об одном из N результатов.

В примере с учителем: $N = 2$, $i = 1$ бит;

в примере с оценками: $N = 4$, $i = 2$ бита;

в примере с вагонами: $N = 8$, $i = 3$ бита.

Нетрудно заметить, что связь между этими величинами выражается следующей формулой: $2^i = N$.

Действительно: $2^1 = 2$; $2^2 = 4$; $2^3 = 8$.

С полученной формулой вы уже знакомы из курса информатики для 7 класса и еще не однажды с ней встретитесь. Значение этой формулы столь велико, что мы назвали её главной формулой информатики. Если величина N известна, а i неизвестно, то данная формула становится уравнением для определения i . В математике такое уравнение называется показательным уравнением.

Пример. Вернемся к рассмотренному выше примеру с вагонами. Пусть в поезде не 8, а 16 вагонов. Чтобы ответить на вопрос, какое количество инфор-

мации содержится в сообщении о номере искомого вагона, нужно решить уравнение: $2^i = 16$. Поскольку, $16 = 2^4$, то $i = 4$ бита. Количество информации i , содержащееся в сообщении об одном из N равновероятных исходов некоторого события, определяется из решения показательного уравнения: $2^i = N$.

Пример. В кинозале 16 рядов, в каждом ряду 32 места. Какое количество информации несет сообщение о том, что вам купили билет на 12-й ряд, 10-е место?

Решение задачи: в кинозале всего $16 \cdot 32 = 512$ мест. Сообщение о купленном билете однозначно определяет выбор одного из этих мест. Из уравнения $2^i = 512 = 2^9$ получаем: $i = 9$ битов. Но эту же задачу можно решать иначе. Сообщение о номере ряда несет 4 бита информации, так как $2^4 = 16$. Сообщение о номере места несет 5 битов информации, так как $2^5 = 32$. В целом сообщение про ряд и место несет: $4 + 5 = 9$ битов информации. Данный пример иллюстрирует выполнение закона аддитивности количества информации (правило сложения): количество информации в сообщении одновременно о нескольких результатах независимых друг от друга событий равно сумме количеств информации о каждом событии отдельно. С формулой $2^i = N$ мы уже встречались, обсуждая алфавитный подход к измерению информации. В этом случае N рассматривалось как мощность алфавита, а i — как информационный вес каждого символа алфавита. Если допустить, что все символы алфавита появляются в тексте с одинаковой частотой, т. е. равновероятно, то информационный вес символа i тождественен количеству информации в сообщении о появлении любого символа в тексте. При этом N — неопределенность знания о том, какой именно символ алфавита должен стоять в данной позиции текста. Данный факт демонстрирует связь между алфавитным и содержательным подходами к измерению информации.

2.3 Формула Хартли

Если значение N равно целой степени двойки (4, 8, 16, 32, 64 и т. д.), то показательное уравнение легко решить в уме, поскольку i будет целым числом.

А чему равно количество информации в сообщении о результате матча «Динамо» – «Зенит»? В этой ситуации $N = 3$. Можно догадаться, что решение уравнения $2^i = 3$ будет дробным числом, лежащим между 1 и 2, поскольку $2^1 = 2 < 3$, а $2^2 = 4 > 3$. А как точнее узнать это число? В математике существует функция, с помощью которой решается показательное уравнение. Эта функция называется логарифмом, и решение нашего уравнения записывается следующим образом: $i = \log_2 N$. Читается это так: «Логарифм от N по основанию 2». Смысл очень простой: логарифм по основанию 2 от A — это степень, в которую нужно возвести 2, чтобы получить N . Например, вычисление уже известных вам значений можно представить так: $\log_2 2 = 1$, $\log_2 4 = 2$, $\log_2 8 = 3$.

Значения логарифмов находятся с помощью специальных логарифмических таблиц. Также можно использовать инженерный калькулятор или табличный процессор. Определим количество информации, полученной из сообщения об одном исходе события из трех равновероятных, с помощью электронной таблицы.

В табличном процессоре Microsoft Excel функция логарифма имеет следующий вид: LOG (аргумент; основание). Аргумент — значение N находится в ячейке A2, а основание логарифма равно 2. В результате получаем с точностью до девяти знаков после запятой: $i = \log_2 3 = 1,584962501$ (бита). Формула для измерения количества информации: $i = \log_2 N$ была предложена американским ученым Ральфом Хартли — одним из основоположников теории информации.

Формула Хартли:

$$i = \log_2 N, \quad (4)$$

где i — количество информации, содержащееся в сообщении;

N — число равновероятных исходов события.

Данный пример показал, что количество информации, определяемое с использованием содержательного подхода, может быть дробной величиной, в то время как информационный объем, вычисляемый путем применения алфавитного подхода, может иметь только целочисленное значение.

2.4 Представление чисел в компьютере

Главные правила представления данных в компьютере.

Если бы мы могли заглянуть в содержание компьютерной памяти, то увидели бы там примерно то, что условно изображено на рисунке 2.

1	1	0	0	1	1	1	1	0	0	0	1	0	1	0	0
0	1	0	0	1	1	1	0	1	0	0	0	1	1	1	1
1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1
1	1	0	0	1	0	1	0	0	0	1	1	0	1	1	1
1	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0
1	1	1	0	0	1	0	1	0	1	1	1	0	0	0	1
1	0	1	0	1	0	1	0	1	1	1	1	1	0	0	0
1	1	0	1	0	1	0	1	1	0	0	1	1	1	0	0
0	1	0	1	0	0	1	1	1	1	1	0	0	0	1	1
1	0	0	1	1	0	0	0	1	1	1	0	1	0	1	0
1	0	0	0	1	1	0	1	0	1	0	1	0	1	0	0
1	1	0	0	1	1	1	0	0	1	1	0	1	0	1	1
0	0	1	1	1	0	0	1	1	0	1	0	1	0	1	0
0	1	1	1	1	0	1	0	1	0	1	0	0	1	1	1
0	0	1	1	1	0	1	0	1	0	1	0	1	1	1	0
1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	1

Рисунок 2 – Образ компьютерной памяти

Рисунок 2 отражает известное вам еще из курса информатики основной школы правило представления данных.

Правило 1: данные (и программы) в памяти компьютера хранятся в двоичном виде, т. е. в виде цепочек единиц и нулей. Современный компьютер может хранить и обрабатывать данные, представляющие информацию четырех видов: числовую, текстовую, графическую, звуковую. Двоичный код, изображенный на рис. 2.8, может относиться к любому виду данных.

Правило 2: представление данных в компьютере дискретно.

Правило 3: множество представимых в памяти компьютера величин ограничено и конечно.

Сначала поясним на образном примере, что такое дискретность. Дискретное множество состоит из отделенных друг от друга элементов. Например, песок дискретен, поскольку он состоит из отдельных песчинок. А вода или масло непрерывны (в рамках наших ощущений, поскольку отдельные молекулы мы все равно ощутить не можем). Этот пример нужен нам только для аналогии. Здесь мы не станем углубляться в изучение материального мира, а вернемся к предмету изучения информатики — информации.

Самым традиционным видом данных, с которым работают компьютеры, являются числа. ЭВМ первого поколения умели решать только математические задачи. Люди начали работать с числами еще с первобытных времен.

Первоначально человек оперировал лишь целыми положительными (натуральными) числами: 1, 2, 3, 4, Очевидно, что натуральный ряд — это дискретное множество чисел. В математике ряд натуральных чисел бесконечен и не ограничен. С появлением в математике понятия отрицательного числа (Р. Декарт, XVII век в Европе; в Индии значительно раньше) оказалось, что множество целых чисел не ограничено как «справа», так и «слева». Покажем это на числовой оси. Символы ∞ обозначают бесконечность. Из сказанного следует вывод: множество целых чисел в математике дискретно и не ограничено. Отметим еще один факт: разность соседних чисел натурального ряда (данного и предыдущего) всегда равна единице. Эту величину назовем шагом числовой последовательности. Любое вычислительное устройство (компьютер, калькулятор) может работать только с ограниченным множеством целых чисел. Возьмите в руки калькулятор, на индикаторном табло которого помещается 10 знаков. Самое большое положительное число, которое на него поместится:

	9	9	9	9	9	9	9	9	9
--	---	---	---	---	---	---	---	---	---

Самое большое по абсолютной величине (модулю) отрицательное число:

-	9	9	9	9	9	9	9	9	9
---	---	---	---	---	---	---	---	---	---

Аналогично дело обстоит и в компьютере.

Целые числа в компьютере.

Правило 4: в памяти компьютера числа хранятся в двоичной системе счисления. Если под целое число выделяется ячейка памяти размером в 16 битов, то самое большое целое положительное число будет таким:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

В десятичной системе счисления оно равно: $2^{15} - 1 = 32\,767$.

Здесь первый бит играет роль знака числа. Ноль — признак положительного числа. Самое большое по модулю отрицательное число равно $-32\,768$. Напомним (это было в курсе информатики основной школы), как получить его внутреннее представление:

1) перевести число $32\,768$ в двоичную систему счисления; это легко, поскольку $32\,768 = 2^{15}$:

1000000000000000;

2) инвертировать этот двоичный код, т. е. заменить нули на единицы, а единицы — на нули:

0111111111111111;

3) прибавить единицу к этому двоичному числу (складывать надо по правилам двоичной арифметики), в результате получим:

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Единица в первом бите обозначает знак «минус». Не нужно думать, что полученный код — это «минус ноль». Этот код представляет число $-32\,768$. Таковы правила машинного представления целых чисел. Данное представление называется дополнительным кодом. Если под целое число в памяти компьютера отводится N битов, то диапазон значений целых чисел: $[-2^{N-1}, 2^{N-1} - 1]$, т. е. ограниченность целого числа в компьютере возникает из-за ограничений на размер ячейки памяти. Отсюда же следует и конечность множества целых чисел.

Мы рассмотрели формат представления целых чисел со знаком, т. е. положительных и отрицательных. Бывает, что нужно работать только с положительными целыми числами. В таком случае используется формат представления целых чисел без знака. В этом формате самое маленькое число — ноль (все биты — нули), а самое большое число для 16-разрядной ячейки:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

В десятичной системе это $2^{16} - 1 = 65\,535$, примерно в два раза больше по модулю, чем в представлении со знаком.

Из всего сказанного делаем вывод: целые числа в памяти компьютера — это дискретное, ограниченное и конечное множество. Границы множества целых чисел зависят от размера выделяемой ячейки памяти под целое число, а также от формата: со знаком или без знака. Шаг в компьютерном представлении последовательности целых чисел, как и в математическом, остается равным единице.

Вещественные числа в компьютере. Понятие вещественного (действительного) числа в математику ввел Исаак Ньютон в XVIII веке. В математике множество вещественных чисел непрерывно, бесконечно и не ограничено. Оно включает в себя множество целых чисел и еще бесконечное множество нецелых чисел. Между двумя любыми точками на числовой оси лежит бесконечное множество вещественных чисел, что и означает непрерывность множества. Как мы говорили выше, числа в компьютере (в том числе и вещественные) представлены в двоичной системе счисления. Покажем, что множество вещественных чисел в компьютере дискретно, ограничено и конечно. Нетрудно догадаться, что это, так же как и в случае целых чисел, вытекает из ограничения размера ячейки памяти. Снова для примера возьмем калькулятор с десятиразрядным индикаторным табло. Экспериментально докажем дискретность представления вещественных чисел. Выполним на калькуляторе деление 1 на 3. Из математики вам известно, что $1/3$ — это рациональная дробь, представление которой в виде десятичной дроби содержит бесконечное количество цифр: $0,3333333333\dots$ (3 в периоде). На табло калькулятора вы увидите:

	0.	3	3	3	3	3	3	3	3
--	----	---	---	---	---	---	---	---	---

Первый разряд зарезервирован под знак числа. После запятой сохраняется 8 цифр, а остальные не вмещаются в разрядную сетку (так это обычно называют). Значит, это не точное значение, равное $1/3$, а его «урезанное» значение.

Следующее по величине число, которое помещается в разрядную сетку:

	0.	3	3	3	3	3	3	3	4
--	----	---	---	---	---	---	---	---	---

Оно больше предыдущего на $0,00000001$. Это шаг числовой последовательности. Следовательно, два рассмотренных числа разделены между собой конечным отрезком. Очевидно, что предыдущее число такое:

	0.	3	3	3	3	3	3	3	2
--	----	---	---	---	---	---	---	---	---

Оно тоже отделено от своего «соседа справа» по числовой оси шагом $0,00000001$. Отсюда делаем вывод: множество вещественных чисел, представимых в калькуляторе, дискретно, поскольку числа отделены друг от друга конечными отрезками. А теперь выясним вот что: будет ли шаг в последовательности вещественных чисел на калькуляторе постоянной величиной (как у целых чисел)? Вычислим выражение $100000/3$. Получим:

	3	3	3	3	3.	3	3	3	3
--	---	---	---	---	----	---	---	---	---

Это число в $100\ 000$ раз больше предыдущего и, очевидно, тоже приближенное. Легко понять, что следующее вещественное число, которое можно получить на табло калькулятора, будет больше данного на $0,0001$. Шаг стал гораздо больше. Отсюда приходим к выводу: множество вещественных чисел, представимых в калькуляторе, дискретно с переменной величиной шага между соседними числами. Если отметить на числовой оси точные значения вещественных чисел, которые представимы в калькуляторе, то эти точки будут расположены вдоль оси неравномерно. Ближе к нулю — гуще, дальше от нуля — реже. Все выводы, которые мы делаем на примере калькулятора, полностью переносятся на компьютер с переходом к двоичной системе счисления и с учетом размера ячейки компьютера, отводимой под вещественные числа. Неравномерное расположение вещественных чисел, представимых в компьютере, также имеет место.

Как вы думаете, ограничено ли множество вещественных чисел в памяти компьютера? Если продолжать эксперименты с калькулятором, то ответ на этот вопрос будет таким: да, множество вещественных чисел в калькуляторе ограничено. Причиной тому служит все та же ограниченность разрядной сетки. От-

сюда же следует и конечность множества. Самое большое число у разных калькуляторов может оказаться разным. У самого простого это будет то же число, что мы получали раньше: 999999999. Если прибавить к нему единицу, то калькулятор выдаст сообщение об ошибке. А на другом, более «умном» и дорогом, калькуляторе прибавление единицы приведет к такому результату:

					1	e	+	0	9
--	--	--	--	--	---	---	---	---	---

Данную запись на табло надо понимать так: $1 \cdot 10^9$. Такой формат записи числа называется форматом с плавающей запятой, в отличие от всех предыдущих примеров, где рассматривалось представление чисел в формате с фиксированной запятой. Число, стоящее перед буквой «e», называется мантиссой, а стоящее после — порядком. «Умный калькулятор» перешел к представлению чисел в формате с плавающей запятой после того, как под формат с фиксированной запятой не стало хватать места на табло.

В компьютере то же самое: числа могут представляться как в формате с фиксированной запятой (обычно это целые числа), так и в формате с плавающей запятой.

Но и для формата с плавающей запятой тоже есть максимальное число. В нашем «подопытном» калькуляторе это число:

	9	9	9	9	9	e	+	9	9
--	---	---	---	---	---	---	---	---	---

То есть $99999 \cdot 10^{99}$. Самое большое по модулю отрицательное значение - $99999 \cdot 10^{99}$. Данные числа являются целыми, но именно они ограничивают представление любых чисел (целых и вещественных) в калькуляторе.

В компьютере все организовано аналогично, но предельные значения еще больше. Это зависит от разрядности ячейки памяти, выделяемой под число, и от того, сколько разрядов выделяется под порядок и под мантиссу.

Рассмотрим пример: пусть под все число в компьютере выделяется 8 байтов — 64 бита, из них под порядок — 2 байта, под мантиссу — 6 байтов. Тогда диапазон вещественных чисел, в переводе в десятичную систему счисления, оказывается следующим: $\pm(5 \cdot 10^{-324} - 1,7 \cdot 10^{308})$.

3 Лекция 3. Интерфейс MS Word 2007. Работа с файлами в MS Word

3.1 Интерфейс Microsoft Word 2007

Интерфейс – это инструмент для диалога пользователя с системой. Интерфейс Word 2007, который получил название *Microsoft Office Fluent*. Он упрощает для пользователей работу с приложениями Microsoft Office и дает возможность быстрее получать лучшие результаты. В предыдущих выпусках приложений Microsoft Office пользователям предоставлялась система меню, панелей инструментов, диалоговых окон. Такой интерфейс работал хорошо, пока в приложениях было ограниченное число команд. Теперь, когда программы выполняют намного больше функций, пользователям становится нелегко работать со старой системой: слишком много программных возможностей, и их трудно найти. В Office Fluent дизайн пользовательского интерфейса был переработан, чтобы упростить процедуры поиска и использования всего диапазона возможностей, предоставляемых этими приложениями. Кроме того, предполагалось предотвратить загромождение рабочей области - ради того, чтобы пользователи меньше отвлекались на поиск и могли больше времени и энергии тратить на свою работу.

Главный элемент пользовательского интерфейса Microsoft Word 2007 представляет собой ленту, которая идет вдоль верхней части окна каждого приложения, вместо традиционных меню и панелей инструментов. С помощью ленты можно быстро находить необходимые команды (элементы управления: кнопки, раскрывающиеся списки, счетчики, флажки и т.п.). Команды упорядочены в логические группы, собранные на вкладках. Заменить ленту панелями инструментов или меню предыдущих версий приложения Microsoft Word нельзя. Удалить ленту также нельзя. Однако, чтобы увеличить рабочую область, ленту можно скрыть (свернуть). Для использования ленты в свернутом состоянии щелкните по названию нужной вкладки, а затем выберите параметр или

команду, которую следует использовать. Например, при свернутой вкладке можно выделить текст в документе Word, щелкнуть вкладку *Главная* и в группе *Шрифт* выбрать нужный размер шрифта. После выбора размера шрифта лента снова свернется. Чтобы быстро свернуть ленту, дважды щелкните имя активной вкладки. Для восстановления ленты дважды щелкните вкладку.

Чтобы свернуть или восстановить ленту, можно также нажать комбинацию клавиш *Ctrl + F1*.

По умолчанию в окне отображается семь постоянных вкладок: *Главная*, *Вставка*, *Разметка страницы*, *Ссылки*, *Рассылки*, *Рецензирование*, *Вид*. Для перехода к нужной вкладке достаточно щелкнуть по ее названию (имени).

Каждая вкладка связана с видом выполняемого действия. Например, вкладка *Главная*, которая открывается по умолчанию после запуска, содержит элементы, которые могут понадобиться на начальном этапе работы, когда необходимо набрать, отредактировать и отформатировать текст. Вкладка *Разметка страницы* предназначена для установки параметров страниц документов. Вкладка *Вставка* предназначена для вставки в документы различных объектов. Кроме того, можно отобразить еще одну постоянную вкладку: *Разработчик*. Щелкните правой кнопкой мыши в любом месте ленты. В контекстном меню выберите команду *Настройка ленты*. В категории *Настройка ленты* окна *Параметры Word* установите флажок этой вкладки. На вкладке *Разработчик* собраны средства создания макросов и форм, а также функции для работы с XML.

Файлы и шаблоны, созданные в версиях Word до 2003 года, могли содержать пользовательские панели инструментов. Кроме того, при установке на компьютер некоторых программ (например, FineReader) в Word автоматически добавляются надстройки. В этом случае при открытии таких файлов в Word 2007 появляется еще одна постоянная вкладка: *Надстройки*. Эта вкладка содержит элементы надстроек, а также элементы панелей инструментов, созданных в предыдущих версиях Word. Каждая панель инструментов занимает отдельную строку ленты вкладки. Однако некоторые элементы, имевшиеся на панелях инструментов, в Word 2007 могут отсутствовать.

Стандартный набор вкладок заменяется при переходе из режима создания документа в другой режим, например, *Удаление фона*. Помимо постоянных, имеется целый ряд контекстных вкладок, например, для работы с таблицами, рисунками, диаграммами и т.п., которые появляются автоматически при переходе в соответствующий режим либо при выделении объекта или установке на него курсора. Например, при создании колонтитулов появляется соответствующая вкладка. В некоторых случаях появляется сразу несколько вкладок, например, при работе с таблицами появляются вкладки *Конструктор* и *Макет*. При снятии выделения или перемещения курсора контекстная вкладка автоматически скрывается. Не существует способов принудительного отображения контекстных вкладок.

Элементы управления на лентах и вкладках объединены в группы, связанные с видом выполняемого действия. Например, на вкладке *Главная* имеются группы для работы с буфером обмена, установки параметров шрифта, установки параметров абзацев, работы со стилями и редактирования.

Элементами управления являются: обычные кнопки, раскрывающиеся кнопки, списки, раскрывающиеся списки, счетчики, кнопки с меню, флажки, значки (кнопки) группы.

Кнопки используются для выполнения какого-либо действия. Например, кнопка *Полужирный* группы *Шрифт* вкладки *Главная* устанавливает полужирное начертание шрифта. Чтобы нажать кнопку, надо щелкнуть по ней мышью. В некоторых случаях нажатие кнопки вызывает диалоговое окно. Обычно кнопки работают в режиме переключателей. То есть, для того чтобы отказаться от назначенного кнопке действия надо еще раз нажать на нее. Например, для отказа от полужирного начертания надо нажать на кнопку *Полужирный* группы *Шрифт* вкладки *Главная*.

Раскрывающиеся кнопки имеют стрелку в правой части. При щелчке по стрелке открывается меню или палитра, в которой можно выбрать необходимое действие или параметр. Выбранное действие или параметр запоминаются на кнопке, и для повторного применения не требуется открывать кнопку. Напри-

мер, можно щелкнуть по стрелке кнопки *Подчеркнутый* группы *Шрифт* вкладки *Главная* и выбрать способ подчеркивания. Чтобы еще раз назначить такое же подчеркивание, не нужно щелкать по стрелке, достаточно щелкнуть по самой кнопке.

В списке можно выбрать для применения какой-либо параметр. Для этого надо щелкнуть по нему мышью. Например, в списке *Стили таблиц* вкладки *Конструктор* можно выбрать вариант оформления. Для просмотра списка в пределах ленты можно пользоваться кнопками *Вниз* и *Вверх*. Но обычно список разворачивают, для чего надо щелкнуть по кнопке *Дополнительные параметры*, после чего отображается галерея списка. В нижней части галереи списка могут отображаться команды меню. Если галерея большая, она имеет в правой части полосу прокрутки. При использовании раскрывающегося списка надо щелкнуть по стрелке списка и выбрать требуемый параметр. Например, в раскрывающемся списке *Размер шрифта* группы *Шрифт* вкладки *Главная* можно выбрать размер шрифта. Если раскрывающийся список большой, он имеет в правой части полосу прокрутки. В большинстве случаев можно, не открывая списка, щелкнуть мышью в поле раскрывающегося списка, ввести значение требуемого параметра с клавиатуры и нажать клавишу *Enter*.

Счетчики применяют для установки числовых значений каких-либо параметров. Для изменения значения пользуются кнопками счетчика *Больше (Вверх)* и *Меньше (Вниз)*. Например, в счетчике *Интервал* группы *Абзац* вкладки *Разметка страницы* можно установить значение величины интервала перед абзацем. Также значение в поле счетчика можно ввести с клавиатуры и нажать клавишу *Enter*.

Нажатие некоторых кнопок не приводит к выполнению какого-либо действия, а только отображает меню, в котором следует выбрать нужное действие. Например, щелкнув по кнопке *Регистр* в группе *Шрифт* вкладки *Главная*, в появившемся меню можно выбрать регистр оформляемого текста. В отличие от раскрывающейся кнопки, выбранное действие не запоминается, и для повтор-

ного применения ранее выбранной команды следует снова щелкнуть по кнопке и выбрать ее.

В некоторых случаях выбор команды меню кнопки вызывает диалоговое окно.

Флажки используются для применения какого-либо параметра. Например, флажки группы *Показать* вкладки *Вид* определяют особенности отображения окна документа. Для установки или снятия флажка достаточно щелкнуть по нему мышью.

Значок (кнопка) группы — маленький квадрат в правом нижнем углу группы элементов во вкладке. Щелчок по значку открывает соответствующее этой группе диалоговое окно или область задач для расширения функциональных возможностей. Например, значок группы *Шрифт* вкладки *Главная* открывает диалоговое окно *Шрифт*. А значок группы *Стили* отображает область задач *Стили*. Не каждая группа имеет значок.

Для каждого элемента управления можно отобразить всплывающую подсказку с информацией о назначении этого элемента — достаточно навести на него и на некоторое время зафиксировать указатель мыши. Иногда в подсказке имеется текст. Для получения дополнительных сведений нажмите клавишу *F1*. В этом случае при нажатии клавиши *F1* появляется окно справочной системы со справкой, относящейся непосредственно к выбранному элементу.

Кнопка *Office* расположена в левом верхнем углу экрана. Начиная с версии 2010, команды этой кнопки переместились на вкладку *Файл*. Вкладка *Файл* всегда расположена в ленте первой слева. По своей сути вкладка *Файл* представляет собой меню. Меню содержит команды для работы с файлами (*Сохранить*, *Сохранить как*, *Открыть*, *Заккрыть*, *Последние*, *Создать*), для работы с текущим документом (*Сведения*, *Печать*, *Доступ*), а также для настройки Word (*Справка*, *Параметры*). Команды *Сохранить как* и *Открыть* вызывают соответствующие окна для работы с файловой системой. Команда *Сведения* открывает раздел вкладки для установки защиты документа, проверки совместимости документа с предыдущими версиями Word, работы с версиями документа, а

также просмотра и изменения свойства документа. Команда *Последние* открывает раздел вкладки со списком последних файлов, с которыми работали в Word, в том числе закрытых без сохранения. Команда *Создать* открывает раздел вкладки с шаблонами для создания новых документов. Команда *Печать* открывает раздел вкладки для настройки и организации печати документа, а также предварительного просмотра документа. Команда *Доступ* открывает раздел вкладки для отправки документа по электронной почте, публикации в Интернете или в сети организации и изменения формата файла документа. Команда *Справка* открывает раздел вкладки для просмотра сведений об установленной версии Microsoft Office, проверки наличия обновлений, настройки параметров Word. Команда *Параметры* отображает диалоговое окно *Параметры Word* для настройки параметров Word. Кнопка *Выход* завершает работу с приложением.

Панель быстрого доступа по умолчанию расположена в верхней части окна Word и предназначена для быстрого доступа к наиболее часто используемым функциям. По умолчанию панель содержит всего три кнопки: *Сохранить*, *Отменить*, *Вернуть (Повторить)*. Панель быстрого доступа можно настраивать, добавляя в нее новые элементы или удаляя существующие. Нажмите кнопку *Настройка панели быстрого доступа*. В меню выберите наименование необходимого элемента. Элементы, отмеченные галочкой, уже присутствуют на панели. Для добавления элемента, отсутствующего в списке, выберите команду *Другие команды*. В разделе *Панель быстрого доступа* окна *Параметры Word* в раскрывающемся списке *Выбрать команды из:* выберите вкладку, в которой расположен добавляемый элемент, затем выделите элемент в списке и нажмите кнопку *Добавить*. Для добавления на панель любого элемента из любой вкладки можно также щелкнуть по этому элементу правой кнопкой мыши и в контекстном меню выбрать команду *Добавить на панель быстрого доступа*. Для удаления элемента из панели достаточно щелкнуть по нему правой кнопкой мыши и в контекстном меню выбрать команду *Удалить с панели быстрого доступа*. Для восстановления стандартного состава панели быстрого доступа в

разделе *Панель быстрого доступа* окна *Параметры Word* нажмите кнопку *Сброс* и выберите команду *Сброс только панели быстрого доступа*.

Мини-панели инструментов содержат основные наиболее часто используемые элементы для оформления текста документа, рисунков, диаграмм и других объектов. Мини-панель для оформления текста появляется автоматически при выделении фрагмента документа. Первоначально отображается полупрозрачная мини-панель. Мини-панель станет яркой, как только на нее будет наведен указатель мыши. Чтобы использовать мини-панель, нажмите нужную кнопку или воспользуйтесь раскрывающимися списками. Можно отключить автоматическое отображение мини-панели инструментов для оформления текста. Нажмите на кнопку *Office* (либо перейдите во вкладку *Файл*, если работаете в более поздних версиях). Нажмите кнопку *Параметры*. В разделе *Общие* окна *Параметры Word* снимите флажок *Показывать мини-панель инструментов при выделении*. Мини-панели для оформления рисунков и других объектов отображаются при щелчке по объекту правой кнопкой мыши. Мини-панель инструментов для оформления рисунка, например, содержит счетчики для изменения размера, кнопки для поворота рисунка и обрезки изображения, кнопки для управления расположением рисунка по слоям. Состав элементов мини-панелей инструментов постоянный и неизменный.

3.2 Работа с файлами в Microsoft Word 2007

В Microsoft Office 2007 принят формат файлов, который основан на языке XML. По умолчанию документы, создаваемые в Word 2007, сохраняются с расширением имени файла, которое получается путем добавления суффикса «x» или «m» к привычному расширению *.doc. Суффикс «x» указывает на то, что XML-файл не содержит макросов, а суффикс «m» - что XML-файл макросы содержит. Таким образом, имена обычных файлов Word 2007 имеют расширение *.docx. XML-форматы предоставляют целый ряд преимуществ не только для разработчиков и создаваемых ими продуктов, но также для отдельных пользователей и организаций. Файлы автоматически сжимаются, и в некоторых

случаях их размер может сокращаться на 75 процентов по сравнению с версиями Word до 2003 года. Формат обеспечивает улучшенное восстановление поврежденных файлов. Файлы построены по модульному принципу, при котором разные компоненты данных хранятся отдельно друг от друга. Это позволяет открывать файлы даже при разрушении или повреждении какого-либо компонента (например, диаграммы или таблицы). Обеспечивается более легкое обнаружение документов, содержащих макросы.

С файлами новых форматов невозможно работать в предыдущих версиях Word. Чтобы можно было открывать и изменять файлы Word 2007 в более ранних версиях, следует загрузить необходимые конвертеры файлов. Многочисленные окна для работы с файловой системой в Word 2007 выглядят примерно одинаково. Их внешний вид зависит от операционной системы. Для удобства работы размер окна можно изменить перетаскиванием его границ. При уменьшении размера окна некоторые его элементы могут оказаться скрытыми. В верхней части окна в поле с раскрывающимся списком отображается имя текущей папки и имя (имена) вышерасположенных папок. Для перехода в вышерасположенную папку достаточно щелкнуть по ее имени. Если щелкнуть по стрелке справа от имени папки, будет отображен список всех содержащихся в ней папок. Для перехода в нужную папку достаточно щелкнуть по ее имени в этом списке. Если щелкнуть по двойной стрелке в левой части поля, будет отображен список вышерасположенных папок и специальных папок Windows. Для перехода в нужную папку достаточно щелкнуть по ее имени.

Поле *Поиск* предназначено для поиска файлов в текущей папке и во всех вложенных папках. Кнопки *Назад* и *Далее* позволяют перейти к предыдущей папке или вернуться в только что покинутую. Если щелкнуть по стрелке, расположенной справа от этих кнопок, появится список всех папок, которые были отображены в текущем окне. Щелчок по кнопке *Упорядочить* вызывает меню. Меню содержит команды для работы с файлами (*Вырезать*, *Копировать*, *Удалить* и др.), свойствами файлов, а также команды для настройки внешнего вида окна (*Раскладка*). Команды подчиненного меню *Раскладка* позволяют отобра-

зять или скрыть *Панель навигации* (расположена в левой части окна), *Панель подробностей*, предназначенную для отображения некоторых свойств файла, *Панель просмотра*, предназначенную для предварительного просмотра файла.

Кнопка *Изменить представление* предназначена для выбора режима отображения значков папок и файлов в окне. Как и в *Проводнике Windows* может быть выбран один из семи режимов: *Огромные значки*, *Крупные значки*, *Обычные значки*, *Мелкие значки*, *Список*, *Таблица*, *Плитка*, *Содержимое*.

Для изменения режима отображения значков несколько раз щелкните по кнопке или щелкните по стрелке кнопки и выберите нужный режим. Для работы чаще всего используют режимы *Мелкие значки* и *Список*, при которых в окне отображается максимальное количество значков. В режиме *Таблица* можно просмотреть некоторые свойства файлов. В режиме отображения *Таблица*, значки папок и файлов можно сортировать, используя кнопки *Имя*, *Дата изменения*, *Тип*, *Размер*, *Ключевые слова*. При щелчке правой кнопкой мыши по любой из этих кнопок появляется контекстное меню, в котором можно выбрать и другие поля сортировки, например, *Авторы*, *Заголовок* и др. Помимо сортировки, указанные кнопки можно использовать и как фильтры. Для этого необходимо навести указатель мыши на кнопку, и щелкнуть по появившейся стрелке. Например, можно фильтровать файлы по дате изменения.

Щелчок по кнопке *Новая папка* создает новую папку в текущей папке. После создания папки следует ввести ее имя. Область переходов, которая иногда называется также *Панель навигации*, содержит ссылки на избранные папки пользователя, библиотеки, специальные папки Windows и папки компьютера. Работают с *Областью переходов* так же, как в окне *Проводника Windows*. Для удобства работы можно увеличить размер окна или, даже, развернуть его на весь экран. Можно также изменить положение границы между *Областью переходов* и остальной частью окна, перетащив ее вправо или влево. Поле *Имя файла* предназначено для указания имени файла при открытии или сохранении. Кнопка с указанием типа файла открывает меню, в котором можно выбрать тип файла. Окно сохранения документа, в отличие от других окон, имеет область

свойств документа. В ней можно указать автора документа и ключевые слова. В этой же области имеется кнопка *Скрыть папки*. Щелчок по ней скрывает *Область переходов* и отображает более полный список свойств документа. Для отображения всей области свойств, перетащите вниз нижнюю границу окна. Для заполнения свойств документа щелкните мышью в соответствующем поле и введите информацию с клавиатуры или извлеките из буфера обмена. Для отображения *Область переходов* щелкните по кнопке *Обзор папок*.

Для открытия файла выполните следующие действия. Нажмите на кнопку *Office* (либо перейдите во вкладку *Файл*, если работаете в более поздних версиях) и выберите команду *Открыть*. В окне *Открытие документа* перейдите к папке или библиотеке, содержащей нужный файл, и щелкните два раза по значку открываемого файла или выделите этот файл и нажмите кнопку *Открыть*. При необходимости можно открыть сразу несколько файлов одной папки. Выделите значки требуемых файлов и нажмите кнопку *Открыть*. Для открытия файла можно также щелкнуть по его значку в окне *Проводника Windows*. Для открытия одного из последних файлов, с которыми работали в Word, можно воспользоваться кнопкой *Office* (вкладкой *Файл*). Выберите команду *Последние*, после чего отобразится раздел *Последние*. Щелкните мышью по имени открываемого файла.

Word позволяет открывать файлы разных форматов. По умолчанию в окне *Открытие документа* отображаются только файлы Word. Чтобы открыть файл другого формата, щелкните по кнопке, на которой указан тип открываемых файлов, и выберите необходимый формат или режим *Все файлы*. Таким образом, например, можно открывать файлы формата txt или rtf. При открытии текстовых файлов формата txt может появиться запрос на выбор кодировки для преобразования файла. При открытии в Word 2007 документа, созданного в Word 2003, Word 2002 или Word 2000, автоматически включается режим совместимости, а в строке заголовка окна документа отображается надпись *Режим ограниченной функциональности*. Включение режима совместимости гарантирует, что при работе с документом не будут использоваться новые и рас-

ширенные возможности Word 2007 и пользователи более ранних версий Microsoft Word смогут редактировать любую часть этого документа. Следует отметить, что в режиме совместимости можно открывать, редактировать и сохранять документы Word 97–2003, но использование новых возможностей Word 2007 будет невозможно. Файлы, сохраненные с какого-либо сайта, скопированные с ftp-сервера или полученные по электронной почте, потенциально могут содержать вирусы и другие опасные вложения и быть небезопасны для компьютера. Такие файлы открываются в режиме защищенного просмотра. В данном режиме файл можно просматривать, копировать его содержимое с использованием буфера обмена, но нельзя редактировать, изменять оформление или печатать. Если есть уверенность в безопасности полученного файла и необходимость работы с ним, нажмите кнопку *Разрешить редактирование* в панели сообщений. После этого с документом можно будет выполнять любые действия, как с обычным открытым документом. При последующем обращении к этому файлу он будет открываться в обычном режиме.

Новый документ создается автоматически сразу после запуска Word. Новый пустой документ или документ на основе шаблона можно создать в процессе работы. Перейдите во вкладку *Файл* и выберите команду *Создать*, после чего отобразится раздел *Создать*. Для создания нового пустого дважды щелкните по значку *Новый документ* или выделите этот значок и нажмите кнопку *Создать*. Для создания документа на основе шаблона, установленного на компьютере, щелкните по значку *Образцы шаблонов*. Прокручивая список выберите нужный шаблон и дважды щелкните по его значку или выделите этот значок и нажмите кнопку *Создать*. Для возврата в начало раздела *Создать* нажмите кнопку *Домой*. При наличии подключения к Интернету можно обратиться к шаблонам, расположенным на сайте Microsoft Office. В разделе *Шаблоны Office.com* щелкните по значку нужной группы шаблонов, дождитесь, пока они загрузятся. Прокручивая список, выберите нужный шаблон и дважды щелкните по его значку или выделите этот значок и нажмите кнопку *Загрузить*. Для создания документа на основе одного из ранее использованных шаблонов щелкни-

те по значку *Последние шаблоны*. Выберите нужный шаблон и дважды щелкните по его значку или выделите этот значок и нажмите кнопку *Создать или Загрузить*.

Для сохранения изменений в существующем файле достаточно нажать кнопку *Сохранить* в панели быстрого доступа. Для сохранения нового документа в виде файла или существующего документа в виде нового файла (с другим именем и/или в другой папке) выполните следующие действия. Нажмите на кнопку *Office* (либо перейдите во вкладку *Файл*, если работаете в более поздних версиях) и выберите команду *Сохранить как*. В окне *Сохранение документа* перейдите к нужной папке. В поле *Имя файла* введите (при необходимости) имя файла (расширение имени вводить не следует) и нажмите кнопку *Сохранить*.

По умолчанию все файлы сохраняются в том же формате, в котором были открыты. Новые документы и файлы Word 2007 сохраняются в формате Документ Word, файлы предыдущих версий — в формате Документ Word 97 – 2003, файлы RTF — как Текст в формате RTF и т.д. При сохранении можно изменить формат файла. Для этого в окне *Сохранение документа* щелкните по кнопке с указанием типа файла и в появившемся меню выберите необходимый тип. Выбранный формат может не поддерживать какие-то элементы оформления документа, о чем появится соответствующее предупреждение. При сохранении документа в формате *Обычный текст* необходимо будет выбрать кодировку для преобразования файла.

Для преобразования (сохранения) файла в наиболее популярные форматы можно воспользоваться и другой процедурой. Нажмите на кнопку *Office* (либо перейдите во вкладку *Файл*, если работаете в более поздних версиях) и выберите команду *Доступ*, после чего отобразится раздел *Доступ*. Нажмите кнопку *Изменить тип файла*. В разделе *Изменить тип файла* щелкните по кнопке с названием нужного формата. В окне *Сохранение документа* перейдите в папку, в которую будет сохранен файл, и укажите имя сохраняемого файла. При сохранении файлов Word 97, Word 2000, Word 2002, Word 2003 в формате *Доку-*

мент Word можно включить режим совместимости с предыдущими версиями, т.е. отключить возможности Word 2007. Это необходимо сделать в том случае, если файл в последующем будет сохраняться для пользователей, не имеющих Word 2007. Для отключения новых возможностей в окне *Сохранение документа* установите флажок *Поддерживать совместимость с предыдущими версиями Word*. При этом файл по-прежнему будет сохраняться в формате Word 2007. При сохранении файлов предыдущих версий Word в формате *Документ Word* может появиться предупреждение о возможных изменениях в макете документа. Нажмите кнопку *ОК*. Для удобства дальнейшей работы можно установить флажок *Больше не задавать этот вопрос*.

Если документ был создан в Word 2007, а сохраняется в формате *Документ Word 97-2003*, в нем автоматически будут произведены некоторые изменения. Некоторые данные в стандартных блоках документа, а также элементах автотекста могут быть утеряны. Ссылки и списки литературы будут преобразованы в статический текст и перестанут автоматически обновляться. Внедренные объекты, созданные в приложениях Microsoft Office 2007, невозможно будет редактировать. Формулы будут преобразованы в изображения. Их редактирование будет невозможным, пока документ не будет преобразован в новый формат файла. Рисунки SmartArt будут преобразованы в один объект, который нельзя редактировать. Положение некоторых надписей изменится. Перед сохранением файла в формате *Документ Word 97-2003* можно проверить, нет ли в текущем документе элементов содержимого или оформления, не поддерживаемых в старых версиях Word. Перейдите во вкладку *Файл* и выберите команду *Сведения*, после чего отобразится раздел *Сведения*. Нажмите кнопку *Поиск проблем* и в появившемся меню выберите команду *Проверка совместимости*. В окне результатов проверки совместимости будет отображен список несовместимых элементов. Для продолжения работы нажмите кнопку *Продолжить*. Нажав кнопку *Выберите отображаемые версии*, можно выбрать версии Word, для которых проводится проверка: только Word 97-2003, только Word 2007 или для всех предыдущих версий. Для того, чтобы в последующем такая проверка

осуществлялась автоматически при сохранении файлов в форматах предыдущих версий, установите флажок *Проверка совместимости при сохранении документов*.

Когда файл, созданный в одной из предыдущих версий Word, сохраняется в файл Word 2007, появляется новый файл, а исходный файл остается в той же папке. Вместо этого можно преобразовать файл таким образом, чтобы файл формата Word 2007 заменил старый файл. Нажмите на кнопку *Office* (либо перейдите во вкладку *Файл*, если работаете в более поздних версиях) и выберите команду *Сведения*, после чего отобразится раздел *Сведения*. Нажмите кнопку *Преобразовать*. В окне предупреждения о возможных изменениях в разметке документа нажмите кнопку *ОК*. Для удобства дальнейшей работы можно установить флажок *Больше не задавать вопрос о преобразовании документов*.

Для закрытия файла нажмите кнопку *Закрыть* в строке названия окна Word 2007. Для закрытия единственного открытого файла без завершения работы с Word 2007 нажмите на кнопку *Office* (либо перейдите во вкладку *Файл*, если работаете в более поздних версиях) и выберите команду *Закрыть*. Для одновременного закрытия всех открытых файлов и завершения работы с Word 2007 нажмите на кнопку *Office* (либо перейдите во вкладку *Файл*, если работаете в более поздних версиях) и выберите команду *Выход*. Если файл был изменен, но не сохранен, то при его закрытии появляется соответствующее предупреждение. Нажмите кнопку *Сохранить* для сохранения файла со всеми внесенными на данный момент изменениями. Нажмите кнопку *Не сохранять* для отказа от сохранения изменений, внесенных в файл с момента последнего его сохранения. Нажмите кнопку *Отмена* для продолжения работы с файлом.

Если в процессе работы с файлом от момента последнего сохранения изменений до его закрытия прошло некоторое время (по умолчанию 10 минут), то срабатывает функция автосохранения документов Word. В этом случае при закрытии файла также появляется предупреждение. Однако в этом случае, даже если нажать кнопку *Не сохранять*, в течение нескольких дней сохраняется копия (черновик) файла в последней его редакции. Автосохраненную копию те-

кущего файла можно открыть. Нажмите на кнопку *Office* (либо перейдите во вкладку *Файл*, если работаете в более поздних версиях) и выберите команду *Сведения*, после чего отобразится раздел *Сведения*. Щелкните мышью по соответствующей ссылке. Файл будет открыт. При этом будет отображена панель сообщений. Данную копию можно сохранить как отдельный файл. Можно в панели сообщений нажать кнопку *Восстановить*, и тогда данная копия заменит текущую версию файла.

4 Лекция 4. Работа с документом в MS Word. Списки в MS Word

4.1 Работа с документом в Microsoft Word 2007

Выбрать режим просмотра документа можно при работе в любой вкладке Word 2007. Ярлыки режимов просмотра документа обычно расположены в правой части строки состояния. Если ярлыки не отображаются, щелкните правой кнопкой мыши в любом месте строки состояния и в появившемся контекстном меню выберите команду *Ярлыки режимов просмотра*. Несколько удобнее переключаться между режимами просмотра документа во вкладке *Вид*. Группа *Режимы просмотра документа* содержит кнопки для выбора основных режимов.

Наиболее часто используемый режим просмотра документа – *Разметка страницы*. В этом режиме положение текста, таблиц, рисунков и других элементов отображается так, как они будут размещаться на печатной странице. Режим разметки удобно использовать и при создании документа, и для окончательного оформления документа, в том числе, для изменения колонтитулов и полей, а также работы с колонками текста и графическими объектами. Именно этот режим устанавливается по умолчанию в Word 2007.

Режим *Черновик* удобен для ввода, редактирования и оформления текста. В этом режиме форматирование текста отображается полностью, а разметка страницы – в упрощенном виде, что ускоряет ввод и редактирование текста.

Границы страниц, колонтитулы, сноски, фон, а также графические объекты, для которых не установлено обтекание *В тексте*, в этом режиме не показываются.

Режим *Веб-документ* обеспечивает представление документа в том виде, который он будет иметь при просмотре в Web-обозревателе, а также для отображения электронных документов, предназначенных только для просмотра на экране. В этом режиме отображается фон, текст переносится по границе окна, а рисунки занимают те же позиции, что и в окне Web-обозревателя.

Режим чтения предназначен для чтения документа на экране компьютера. Впрочем, в этом режиме можно также отображать документ в том виде, в каком он будет выведен на печать. Здесь можно и редактировать текст, делать пометки, добавлять примечания, работать в режиме записи исправлений, хотя все это используется редко. Для настройки параметров режима чтения щелкните по кнопке *Параметры просмотра* в верхнем правом углу экрана и выберите соответствующую команду меню. Можно увеличить или уменьшить размер шрифта. Этот размер отображается только в режиме чтения и не влечет изменений фактического размера. Можно отображать одну или две страницы в окне. Для показа документа в режиме предварительного просмотра, т.е. в том виде, в каком он будет выведен на печать, выберите команду *Показать печатную страницу*. Изменить масштаб просмотра в этом случае невозможно. Правда, можно управлять отображением полей страницы. Для этого щелкните по команде *Параметры полей* и выберите соответствующую команду в подчиненном меню. В обычном режиме чтения данная команда недоступна. Следует отметить, что режим отображения печатной страницы запоминается. При последующем переходе к режиму чтения будет автоматически установлен именно такой режим. Для отказа от отображения печатной страницы в режиме чтения щелкните по кнопке *Параметры просмотра* в верхнем правом углу экрана и выберите команду *Показать печатную страницу* еще раз. Для возможности редактирования документа в режиме чтения выберите команду *Разрешить ввод*. Режим редактирования распространяется на все документы, для которых устанавливается режим чтения, а также запоминается и автоматически устанавливается при

повторном переходе к режиму чтения. Некоторые дополнительные параметры режима чтения можно использовать, если щелкнуть по кнопке *Сервис* в верхнем правом углу экрана. Например, можно воспользоваться поиском в документе (команда *Найти*), создать примечание (команда *Создать примечание*), выделить текст цветом (палитра цветом подчиненного меню *Цвет выделения текста*). Для выбора цвета выделения текста и создания примечаний имеются также кнопки рядом с кнопкой *Сервис* в своеобразной панели инструментов. Для выхода из режима чтения нажмите кнопку *Заккрыть* в правом верхнем углу окна или клавишу *Esc*.

Структура — один из самых редко используемых режимов отображения документа, хотя он обладает многими возможностями, которые отсутствуют в других режимах. Здесь можно свернуть документ, оставив только основные заголовки, или развернуть его, отобразив все заголовки и основной текст, можно перемещать и копировать текст, перетаскивая заголовки. Кроме того, в режиме структуры удобно работать с главными документами, включающими несколько частей. Однако, в этом режиме не отображаются границы страниц, колонтитулы, рисунки, фон и многие другие элементы оформления документа. Обратите внимание на то, что в режиме структуры документа появляется новая вкладка *Структура*, упрощающая работу с заголовками, главным и вложенными документами. В других режимах эту вкладку вывести невозможно, но она там и не нужна.

Изменить масштаб отображения документа можно при работе в любой вкладке Word 2007. Щелкните по кнопке со знаком «+» (плюс) для увеличения масштаба или по кнопке со знаком «-» (минус) для уменьшения. Масштаб можно также изменять перетаскиванием ползунка линейки масштаба. Если линейка масштаба не отображается, щелкните правой кнопкой мыши в любом месте строки состояния и в появившемся контекстном меню выберите команду *Масштаб*. Минимальный масштаб отображения 10 %, максимальный — 500 %. Масштаб отображения документа можно изменять также во вкладке *Вид*. В группе *Масштаб* нажмите нужную кнопку: *100%*, *Одна страница*, *Две страни-*

цы или *Ширина страницы*. Нажав кнопку *Масштаб* в диалоговом окне *Масштаб*, можно установить требуемый масштаб отображения документа.

Чтобы поставить курсор в определенное место документа, надо подвести указатель мыши (при наведении на текст он будет иметь вид латинской буквы I) и один раз щелкнуть левой кнопкой мыши. Клавиши \rightarrow и \leftarrow переводят курсор на один символ вправо или влево, а клавиши \downarrow и \uparrow — на одну строку вниз или вверх. Клавиши *Home* и *End* переводят курсор в начало или конец строки, клавиши *PageUp* и *PageDown* — на высоту экрана вверх или вниз. Комбинация клавиш *Ctrl* + \rightarrow переводит курсор на одно слово вправо, а *Ctrl* + \leftarrow на одно слово влево. *Ctrl* + \downarrow переводит курсор к началу следующего абзаца, *Ctrl* + \uparrow — к началу текущего абзаца. Комбинация клавиш *Ctrl* + *PageUp* переводит курсор на следующую страницу, а *Ctrl* + *PageDown* - на предыдущую страницу (нужно отметить, что эти комбинации клавиш клавиатуры не всегда работают именно так). Комбинация клавиш *Ctrl* + *Home* переводит курсор в начало документа к самому первому символу, а *Ctrl* + *End* - к концу документа к самому последнему символу. Некоторые дополнительные возможности имеются при перемещении курсора по тексту, оформленному в несколько колонок. Комбинация левой клавиши клавиатуры *Alt* + \downarrow переводит курсор к началу следующей колонки, а левой клавиши *Alt* + \uparrow — к началу предыдущей колонки. Также некоторые особенности и дополнительные возможности имеются при перемещении курсора в таблицах. Клавиша клавиатуры *Tab* переводит курсор в следующую ячейку таблицы, а комбинация клавиш клавиатуры *Shift* + *Tab* - в предыдущую ячейку. Комбинация клавиш *Alt* + *Home* переводит курсор в первую ячейку строки, а *Alt* + *End* - в последнюю ячейку строки. Комбинация клавиш *Alt* + *PageUp* переводит курсор в верхнюю ячейку столбца, а *Alt* + *PageDown* в нижнюю ячейку столбца. Для перевода курсора к какой-либо определенной странице, разделу, строке и т.д. щелкните по стрелке кнопки *Найти* группы *Редактирование* вкладки *Главная* и выберите команду *Перейти* или нажмите клавишу клавиатуры *F5*. Во вкладке *Перейти* диалогового окна *Найти и заменить* в списке *Объект перехода* выберите объект перехода, введите

его номер и нажмите кнопку *Перейти*. Для перевода курсора к какому-либо объекту в документе (следующая страница, следующий рисунок, следующая таблица и др.) можно использовать также кнопку *Выбор объекта* в нижней части вертикальной полосы прокрутки. Необходимый объект или действие следует выбрать в меню *Выбор объекта*. Для перехода к следующему такому же объекту нажмите кнопку *Далее*. Для возврата к предыдущему — кнопку *Назад*. Вместо кнопки *Назад* можно использовать комбинацию клавиш клавиатуры *Ctrl + PageUp*, а вместо кнопки *Далее* нажать *Ctrl + PageDown*.

Область навигации в Word 2007 позволяет отображать заголовки документа (схему документа), эскизы страниц, а также производить поиск в документе. Для отображения *Области навигации* установите соответствующий флажок в группе *Показать* вкладки *Вид*. Для скрытия *Области навигации* снимите соответствующий флажок в группе *Показать* вкладки *Вид* или нажмите кнопку *Заккрыть* в *Области навигации*. Для просмотра заголовков перейдите в соответствующую вкладку *Области навигации*. Схема документа в области навигации формируется в том случае, если при оформлении заголовков использовались стили типа *Заголовок 1*, *Заголовок 2* и т.д. или абзацам был назначен соответствующий уровень. Для отображения эскизов перейдите в соответствующую вкладку *Области навигации*.

При работе в Word существует возможность отмены действий, выполненных с документом. Для отмены последнего выполненного действия нажмите кнопку *Отменить* в *Панели быстрого доступа*. Можно отменить сразу несколько последних действий. Щелкните по стрелке кнопки *Отменить*. Выберите действие, начиная с которого следует отменить все действия, расположенные выше в списке. Отмененные действия можно вернуть. Для возврата последнего *отмененного действия нажмите кнопку Вернуть* в *Панели быстрого доступа*. Для возврата нескольких действий несколько раз нажмите кнопку *Вернуть*. Если все отмененные действия были возвращены, кнопка *Вернуть* заменяется на кнопку *Повторить*.

Выделять фрагменты текста можно перемещением курсора клавишами клавиатуры при нажатой клавише *Shift*. При этом можно использовать все возможные комбинации клавиш для перевода курсора по документу. Например, если необходимо выделить часть документа от текстового курсора до конца документа, то достаточно нажать комбинацию клавиш клавиатуры *Shift + Ctrl + End*. Для того чтобы снять выделение фрагмента текста, следует один раз щелкнуть левой кнопкой мыши в правом поле документа. Одна строка таблицы выделяется так же, как строка текста, то есть щелчком мыши слева от выделяемой строки. Таким же способом, как строки текста, выделяются и несколько строк таблицы. Для выделения столбца таблицы следует подвести указатель мыши сверху к столбцу, пока он не превратится в маленькую черную вертикальную стрелку, и после этого один раз щелкнуть левой кнопкой мыши. Для выделения нескольких столбцов надо любым способом выделить первый из выделяемых столбцов, а затем, не отпуская левой кнопки мыши, распространить выделение на следующие столбцы. Несколько столбцов можно выделять также с использованием клавиши клавиатуры *Shift*. Любым способом выделите первый столбец, а затем выделите последний столбец при нажатой клавише *Shift*. Выделение одной ячейки при работе с таблицей (установка границ, заливки, разделение ячеек и т. д.) может потребоваться при оформлении текста в ячейке. Для выделения ячейки необходимо подвести указатель мыши слева к границе ячейки и, когда он превратится в маленькую черную стрелку, щелкнуть левой кнопкой мыши. Ячейку, содержащую какой-либо текст, можно выделить следующим способом: три раза щелкнуть левой кнопкой мыши по последнему слову текста ячейки. Для выделения пустой ячейки можно поставить в нее курсор и нажать комбинацию клавиш клавиатуры *Shift + End*. Для выделения нескольких ячеек надо подвести указатель мыши к первой из выделяемых ячеек, а затем при нажатой левой кнопки мыши распространить выделение на соседние ячейки. Эту операцию удобнее выполнять с использованием клавиш клавиатуры. Поставьте курсор в первую из выделяемых ячеек, а затем при нажатой клавише *Shift* распространяйте выделение клавишами \leftarrow , \rightarrow , \uparrow , \downarrow . Выделять всю

таблицу лучше всего по строкам или по столбцам. Можно для выделения использовать также маркеры таблицы. Щелчок левой кнопкой мыши по любому из маркеров выделит всю таблицу.

Для выделения можно использовать мышью, или клавиши клавиатуры, или и то, и другое одновременно (см. таблицу 1).

Таблица 1 – Выделение фрагментов текста

Фрагмент	Действие для выделения
Любой фрагмент текста	Щелкните в начале выделяемого фрагмента в тексте и, удерживая нажатой левую кнопку мыши, проведите по тексту, который нужно выделить
Слово	Дважды щелкните по слову
Строка текста	Подведите указатель к левому краю строки так, чтобы он принял вид стрелки, направленной вправо и вверх, а затем щелкните
Предложение	Удерживая нажатой клавишу <i>Ctrl</i> , щелкните по предложению
Абзац	Трижды щелкните в любом месте абзаца
Несколько абзацев	Подведите указатель к левому краю первого абзаца так, чтобы он принял вид стрелки, направленной вправо и вверх, а затем, удерживая нажатой левую кнопку мыши, проведите указателем вверх или вниз
Большой фрагмент текста	Щелкните в начале выделяемого фрагмента, прокрутите документ до конца фрагмента, а затем, удерживая нажатой клавишу <i>Shift</i> , щелкните в конце фрагмента
Весь документ	Переместите указатель к левому краю текста документа так, чтобы он принял вид стрелки, направленной вправо и вверх, и щелкните трижды
Вертикальный блок текста	Удерживая нажатой клавишу <i>Alt</i> , проведите указателем по тексту

4.2 Списки в MS Word 2007

При оформлении списков следует иметь в виду, что элементом списка может быть только абзац. Список, набранный в строку, автоматически пронумеровать нельзя. Выделите несколько абзацев, оформляемых в виде списка. Щелкните по стрелке кнопки *Нумерация* группы *Абзац* вкладки *Главная* и в галерее списков выберите нужный вариант нумерации. При наведении указателя мыши на выбираемый список срабатывает функция динамического просмотра, и фрагмент документа отображается пронумерованным. Используемые ранее виды списков накапливаются в галерее списков. В верхней части галереи отдельную группу образуют форматы номеров, применявшиеся в текущем сеансе работы в Word 2007. Этот список автоматически очищается при завершении работы в Word. В нижней части галереи отдельную группу образуют форматы номеров, использовавшиеся в текущем документе. Этот список остается доступным при каждом открытии документа.

Можно изменить параметры созданного ранее списка или настроить эти параметры при создании списка. Выделите абзацы, оформленные или оформляемые в виде списка. Щелкните по стрелке кнопки *Нумерация* группы *Абзац* вкладки *Главная* и выберите команду *Определить новый формат номера*. В диалоговом окне *Определение нового формата номера* установите необходимые параметры. В раскрывающемся списке *Нумерация* выберите требуемый вид нумерации. Можно выбрать нумерацию арабскими и римскими цифрами, латинскими и русскими буквами, количественными и порядковыми числительными и т.д. В поле *Формат номера* при необходимости добавьте к нумерации текст, который будет отображаться при каждом номере. Текст можно вводить как перед номером, так и после него. Например, при нумерации списка кафедр можно добавить к номерам слово *Кафедра*.

Обычно нумерация имеет те же параметры шрифта, что и нумеруемый список. Можно изменить параметры шрифта номеров и сопровождающего текста. Нажмите кнопку *Шрифт* и во вкладках *Шрифт* и *Интервал* диалогового

окна *Шрифт* установите требуемые параметры шрифта. В раскрывающемся списке *Выравнивание* выберите вид выравнивания номеров относительно позиции номера. Для имеющегося списка можно изменить начальный номер, с которого начинается список. Можно также оформить текущий список как продолжение предыдущего списка в документе. Выделите абзацы, оформленные в виде списка. Щелкните по стрелке кнопки *Нумерация* группы *Абзац* вкладки *Главная* и выберите команду *Задать начальное значение*. В диалоговом окне *Задание начального значения* установите необходимые параметры. Для выбора начального номера, а также присоединения нумерации текущего списка к предыдущему, можно щелкнуть правой кнопкой мыши по первому абзацу в списке и в контекстном меню выбрать нужную команду.

Каждый вариант оформления нумерованного списка имеет свои параметры отступов оформляемых абзацев, а также устанавливает собственные позиции табуляции, отделяющие нумерацию от текста нумеруемых абзацев. При этом размеры отступов, установленные ранее при оформлении абзацев, могут быть изменены. Отступы в списках можно устанавливать так же, как и в обычных абзацах, но удобнее воспользоваться специальной настройкой. Щелкните правой кнопкой мыши по любому абзацу в списке и в контекстном меню выберите команду *Изменить отступы в списке*. В диалоговом окне *Изменение отступов в списке* установите необходимые параметры. В счетчике *Положение номера* установите расстояние от позиции номера до левого поля страницы. В счетчике *Отступ текста* установите отступ от левого поля страницы всех строк, кроме первой. В раскрывающемся списке *Символ после номера* выберите символ, который будет отделять номер от текста нумеруемого абзаца. Можно выбрать знак табуляции, установить флажок *Добавить позицию табуляции* и в счетчике указать расстояние между номерами и текстом. Можно выбрать пробел между номерами и текстом. Можно выбрать *нет*, чтобы не оставлять места между номерами и текстом. Нумерация удаляется одинаково, независимо от способа установки и вида установленной нумерации. Выделите абзацы, для которых необходимо удалить нумерацию. Нажмите кнопку *Нумерация* группы

Абзац вкладки *Главная* или щелкните по стрелке этой кнопки и выберите режим *нет*. При оформлении списков следует иметь в виду, что элементом списка может быть только абзац. Список, набранный в строку, маркировать нельзя. Выделите несколько абзацев, оформляемых в виде списка. Щелкните по стрелке кнопки *Маркеры* группы *Абзац* вкладки *Главная* и в галерее выберите нужный маркер. При наведении указателя мыши на выбираемый маркер срабатывает функция динамического просмотра, и фрагмент документа отображается маркированным. Используемые ранее виды маркированных списков накапливаются в галерее маркеров. В верхней части галереи отдельную группу образуют маркеры, применявшиеся в текущем сеансе работы в Word 2007. Этот список автоматически очищается при завершении работы в Word. В нижней части галереи отдельную группу образуют маркеры, использовавшиеся в текущем документе. Этот список остается доступным при каждом открытии документа. Можно выбрать маркер для созданного ранее списка или изменить маркер существующего списка. Выделите абзацы, оформленные или оформляемые маркерами. Щелкните по стрелке кнопки *Маркеры* группы *Абзац* вкладки *Главная* и выберите команду *Определить новый маркер*. В диалоговом окне *Определение нового маркера* для выбора нового маркера нажмите кнопку *Символ*. В диалоговом окне *Символ* в раскрывающемся списке *Шрифт* выберите шрифт, символы которого будут использоваться в виде маркера (можно взять любой шрифт, но наиболее интересные символы содержатся в шрифтах *Webdings*, *Wingdings*, *Wingdings2* и *Wingdings3*), и дважды щелкните левой кнопкой мыши по выбранному символу. Можно также выделить символ, а затем нажать кнопку *OK*.

Обычно маркеры имеет те же параметры шрифта, что и список. Можно изменить параметры шрифта маркеров (цвет, начертание, размер и т.д.). Нажмите кнопку *Шрифт* и во вкладках *Шрифт* и *Интервал* диалогового окна *Шрифт* установите требуемые параметры шрифта. При этом сам шрифт изменять нельзя, иначе изменятся и маркеры. В качестве маркера можно использовать рисунки. В диалоговом окне *Определение нового маркера* нажмите кнопку

Рисунок и выберите рисунок в диалоговом окне *Рисованный маркер*. В раскрывающемся списке *Выравнивание* выберите вид выравнивания маркеров относительно позиции маркера. Каждый вариант оформления маркированного списка имеет свои параметры отступов оформляемых абзацев, а также устанавливает собственные позиции табуляции, отделяющие маркеры от текста абзацев. При этом размеры отступов, установленные ранее при оформлении абзацев, могут быть изменены. Отступы в списках можно устанавливать так же, как и в обычных абзацах, но удобнее воспользоваться специальной настройкой. Щелкните правой кнопкой мыши по любому абзацу в списке и в контекстном меню выберите команду *Изменить отступы в списке*. В диалоговом окне *Изменение отступов в списке* установите необходимые параметры. В счетчике *Положение маркера* установите расстояние от позиции маркера до левого поля страницы. В счетчике *Отступ текста* установите отступ от левого поля страницы всех строк, кроме первой. В раскрывающемся списке *Символ после маркера* выберите символ, который будет отделять маркер от текста нумеруемого абзаца.

При оформлении многоуровневых списков следует иметь в виду, что элементом списка может быть только абзац. Выделите несколько абзацев, оформляемых в виде многоуровневого списка. Щелкните по кнопке *Многоуровневый список* группы *Абзац* вкладки *Главная* и в галерее списков выберите нужный вариант нумерации. При наведении указателя мыши выбираемый список отображается более крупно, но функция динамического просмотра в документе не работает. Если абзацы оформляемого текста не различаются отступами слева, то первоначально список не будет многоуровневым. Для понижения уровня отдельных абзацев списка надо увеличить отступ слева. Выделите абзац (абзацы) и нажмите кнопку *Увеличить отступ* группы *Абзац* вкладки *Главная*. Каждое нажатие кнопки понижает выделенные абзацы на один уровень. Для повышения уровня нажмите кнопку *Уменьшить отступ*. Для этой же цели можно использовать аналогичные кнопки мини-панели инструментов. Для понижения уровня элементов в списке можно также использовать клавишу *Tab*, а для повышения комбинацию *Shift + Tab*. Используя ранее виды списков

накапливаются в галерее списков. В верхней части галереи в разделе *Текущий список* отображается текущий или последний применявшийся список. В нижней части галереи отдельную группу образуют списки, использовавшиеся в открытых документах. Этот список автоматически очищается при завершении работы в Word.

Можно изменить параметры ранее созданного списка или настроить эти параметры при создании списка. Выделите абзацы, оформленные или оформляемые в виде списка. Щелкните по кнопке *Многоуровневый список* группы *Абзац* вкладки *Главная* и выберите команду *Определить новый многоуровневый список*. В диалоговом окне *Определение нового многоуровневого списка* установите необходимые параметры. Для отображения всех возможностей окна нажмите кнопку *Больше*. Для скрытия некоторых возможностей нажмите кнопку *Меньше*. Изменение параметров списка производится отдельно для каждого уровня. В списке *Выберите уровень для изменения* выделите уровень изменяемого списка. В раскрывающемся списке *Нумерация* для этого уровня выберите требуемый вид нумерации или маркеров. Можно выбрать нумерацию арабскими и римскими цифрами, латинскими и русскими буквами, количественными и порядковыми числительными и т.д. Если требуется маркер, отсутствующий в списке, то в раскрывающемся списке выберите режим *Новый маркер*. После этого выберите символ в диалоговом окне *Символ*. При необходимости в счетчике *Начать с* укажите начальный номер списка. В поле *Формат номера* при необходимости добавьте к нумерации текст, который будет отображаться при каждом номере. Текст можно вводить как перед номером, так и после него.

Обычно нумерация или маркеры имеют те же параметры шрифта, что и список. Можно изменить параметры шрифта номеров (маркеров) и сопровождающего текста. Нажмите кнопку *Шрифт* и во вкладках *Шрифт* и *Интервал* диалогового окна *Шрифт* установите требуемые параметры. В раскрывающемся списке *Выравнивание* выберите вид выравнивания номеров относительно позиции номера. В счетчике *на* установите расстояние от позиции номера до левого поля страницы. В счетчике *Отступ текста* установите отступ от левого

поля страницы всех строк нумеруемого абзаца, кроме первой. Можно настроить отступы сразу для всех уровней списка. Нажмите кнопку *Для всех уровней*. В окне *Параметры для всех уровней* настройте положение маркера или номера первого уровня, положение текста первого уровня, а затем укажите дополнительный отступ для каждого более низкого уровня. В раскрывающемся списке *Символ после номера* выберите символ, который будет отделять номер от текста нумеруемого абзаца. При необходимости повторите указанные процедуры для каждого уровня нумерации. Для имеющегося списка можно изменить начальный номер, с которого начинается список. Можно также оформить текущий список как продолжение предыдущего списка в документе. Щелкните правой кнопкой мыши по первому абзацу в списке и в контекстном меню выберите команду *Задать начальное значение*. В диалоговом окне *Определение нового формата номера* установите необходимые параметры.

Сортировать можно любые абзацы текста, но чаще всего такая задача встает при работе со списками. Сортируют нумерованные и маркированные списки. Сортировка многоуровневых списков бессмысленна, поскольку при этом разрушается структура списка. Выделите абзацы списка, подлежащего сортировке. Нажмите кнопку *Сортировка* группы *Абзац* вкладки *Главная*. В диалоговом окне *Сортировка текста* в раскрывающемся списке тип выберите тип сортируемых данных (*Текст*, *Число* или *Дата*). Установите переключатель направления сортировки (по возрастанию или по убыванию).

5 Лекция 5. Создание и изменение таблиц в MS Word

5.1 Создание таблиц

Таблицы в документах Word используют, большей частью, для упорядочивания представления данных. В таблицах можно производить вычисления, таблицы можно применять для создания бланков документов. Табличные данные можно использовать для создания диаграмм. Ячейки таблицы могут со-

держат текст, графические объекты, вложенные таблицы. Для вставки таблицы используют вкладку *Вставка*.

Для работы с таблицами в Microsoft Word применяют контекстные вкладки *Конструктор* и *Макет* группы вкладок *Работа с таблицами*. Эти вкладки автоматически отображаются, когда курсор находится в какой-либо ячейке существующей таблицы. Документы могут содержать таблицы, импортированные из Microsoft Excel, и таблицы Microsoft Excel, созданные непосредственно в документе. Таблица всегда вставляется в то место документа, где в данный момент находится курсор. Лучше всего поставить курсор в начало абзаца текста, перед которым должна располагаться создаваемая таблица. Для быстрой вставки простой таблицы во вкладке *Вставка* нажмите кнопку *Таблица* и в появившемся табло при нажатой левой кнопке выделите необходимое число столбцов и строк. При наведении указателя мыши срабатывает функция динамического просмотра, и создаваемая таблица отображается в документе. Таблица занимает всю ширину страницы и имеет столбцы одинаковой ширины.

Можно настроить некоторые параметры создаваемой таблицы. Установите курсор туда, где будет находиться создаваемая таблица. Щелкните кнопку *Таблица* во вкладке *Вставка* и выберите команду *Вставить таблицу*.

В окне *Вставка таблицы* выберите требуемое количество строк и столбцов, а также способ автоподбора. При выборе режима постоянный можно установить ширину столбцов таблицы.

При создании таблицы можно воспользоваться имеющимися заготовками Microsoft Word или собственными, ранее созданными таблицами. Установите курсор туда, где будет находиться создаваемая таблица. Нажмите кнопку *Таблица* во вкладке *Вставка* и выберите команду *Экспресс-таблицы*. Прокрутите список таблиц и выберите нужную. Пользовательские таблицы, занесенные в список экспресс-таблиц, обычно располагаются в разделе *Общие* в конце списка. В результате в документ будет вставлена готовая, содержащая текст и оформление таблица. Оформление встроенных экспресс-таблиц Microsoft Word зависит от выбранной темы документа. Вставленная таблица может иметь заго-

ловок. Ненужный текст можно удалить. Можно также изменить оформление таблицы.

Таблицу из документов (рабочих книг) Microsoft Excel можно копировать в документ Microsoft Word с помощью буфера обмена. Вставляемая таблица может быть связанной или несвязанной. В связанной таблице данные могут обновляться при изменении данных в исходной таблице Microsoft Excel. В несвязанной таблице обновление данных не происходит. Выделите таблицу или фрагмент таблицы на листе документа Microsoft Excel и копируйте в буфер обмена. Установите курсор туда, где будет находиться вставляемая таблица. Для вставки несвязанной таблицы нажмите кнопку *Вставить* во вкладке *Главная*. Для вставки связанной таблицы щелкните по стрелке кнопки *Вставить* во вкладке *Главная* и выберите команду *Специальная вставка*. В окне *Специальная вставка* установите переключатель *связать* и выберите формат вставляемой таблицы.

Для преобразования текста в таблицу необходимо разделить текст с помощью табуляции, точки с запятой или другого знака-разделителя, чтобы указать начало нового столбца. Строка таблицы отмечается знаком абзаца. Выделите фрагмент документа, преобразуемый в таблицу. Щелкните кнопку *Таблица* во вкладке *Вставка* и выберите команду *Преобразовать в таблицу*. В окне *Преобразовать в таблицу* в счетчике *Число столбцов* установите число столбцов создаваемой таблицы (число строк устанавливается автоматически); в разделе *Автоподбор ширины столбцов* выберите способ изменения ширины столбцов таблицы (при выборе параметра *постоянная* можно указать ширину столбцов); выберите знак разделителя.

Для добавления строк в таблицу можно воспользоваться контекстной вкладкой *Макет*. Установите курсор в любую ячейку строки таблицы, выше или ниже которой требуется вставить новую строку. Нажмите кнопку *Вставить сверху* или *Вставить снизу* группы *Строки и столбцы*. При работе в других вкладках для вставки строк можно воспользоваться контекстным меню. Щелкните правой кнопкой мыши по любой ячейке строки таблицы, выше или

ниже которой требуется вставить новую строку, и выберите команду *Вставить*. В подчиненном меню выберите команду *Вставить строки сверху* или *Вставить строки снизу*.

Если в таблицу требуется вставить сразу несколько новых строк, то следует выделить в таблице такое же число строк, а затем воспользоваться любым из описанных выше способов. При работе с клавиатурой для вставки строки можно поставить курсор справа от самой правой ячейки строки и нажать клавишу *Enter*. Новая строка снизу таблицы автоматически появляется, если, находясь в последней ячейке последней строки таблицы, нажать клавишу *Tab*.

Для добавления столбцов в таблицу можно воспользоваться контекстной вкладкой *Макет*. Установите курсор в любую ячейку столбца таблицы, левее или правее которого требуется вставить новый столбец. Нажмите кнопку *Вставить слева* или *Вставить справа* группы *Строки и столбцы*. При работе в других вкладках для вставки столбцов можно воспользоваться контекстным меню.

Если в таблицу требуется вставить сразу несколько новых столбцов, то следует выделить в таблице такое же число столбцов, а затем воспользоваться любым из описанных выше способов. Для удаления строк и столбцов таблицы можно воспользоваться контекстной вкладкой *Макет*. Установите курсор в любую ячейку удаляемой строки или удаляемого столбца таблицы. Нажмите кнопку *Удалить* группы *Строки и столбцы* и в меню выберите соответствующую команду (*Удалить строки* или *Удалить столбцы*).

При работе в других вкладках для удаления строк и столбцов можно воспользоваться клавиатурой или контекстным меню. Выделите удаляемую строку (удаляемый столбец). Нажмите клавишу *BackSpace* или щелкните правой кнопкой мыши по выделенному элементу таблицы и выберите команду *Удалить строки* (*Удалить столбцы*). Если в таблице требуется удалить сразу несколько строк или столбцов, то следует их выделить, а затем воспользоваться любым из описанных выше способов.

Клавиша *Delete* не удаляет строки и столбцы таблицы, а только очищает их содержимое. Однако если выделить одновременно строку текста над табли-

цей и строку (строки) таблицы и нажать *Delete*, то удалится и строка текста, и выделенная часть таблицы. То же самое будет при выделении части таблицы и текста ниже таблицы. Удаление ячеек может потребоваться для исправления структуры таблицы. Установите курсор в ячейку, нажмите кнопку *Удалить* группы *Строки и столбцы*, и выберите команду *Удалить ячейки* или щелкните правой кнопкой мыши по ячейке и выберите команду контекстного меню *Удалить ячейки*. В появившемся окне выберите направление сдвига существующих ячеек таблицы.

Для удаления таблицы можно воспользоваться контекстной вкладкой *Макет*. Установите курсор в любую ячейку таблицы. Нажмите кнопку *Удалить* группы *Строки и столбцы* и в меню выберите команду *Удалить таблицу*. Таблицу также можно удалить, удалив все ее строки или столбцы.

Высоту строки можно быстро изменить перетаскиванием нижней границы строки. Наведите указатель мыши на границу строки так, чтобы он превратился в двунаправленную стрелку. Нажмите на левую кнопку мыши и перетащите границу строки. Если при этом держать нажатой клавишу *Alt*, то на вертикальной линейке будет указана высота каждой строки таблицы. Для установки высоты строк в таблице можно воспользоваться контекстной вкладкой *Макет*. Установите курсор в любую ячейку строки таблицы или выделите несколько смежных строк. В счетчике *Высота строки таблицы* группы *Размер ячейки* установите требуемое значение. Высота строк обычно устанавливается в режиме автоподбора. Например, если установлена высота строки 0,7 см, то при увеличении количества текста или размера шрифта в ячейках строки высота будет автоматически увеличиваться. При уменьшении количества текста высота строки будет автоматически уменьшаться, но не будет меньше установленного значения (0,7 см). Режим автоподбора можно отключить. Установите курсор в любую ячейку строки таблицы или выделите несколько строк. Во вкладке *Макет* в группе *Таблица* нажмите кнопку *Свойства*. Во вкладке *Строка* окна *Свойства таблицы* в поле списка *Режим* выберите параметр *Точно*.

Ширину столбца можно быстро изменить перетаскиванием его границы. Следует иметь в виду, что при этом может изменяться ширина смежных столбцов. Наведите указатель мыши на границу столбца так, чтобы он превратился в двунаправленную стрелку. Нажмите на левую кнопку мыши и перетащите границу столбца. Если при этом держать нажатой клавишу *Alt*, то на горизонтальной линейке будет указана ширина каждого столбца таблицы без учета левого и правого полей ячеек.

Для установки ширины столбцов в таблице можно воспользоваться контекстной вкладкой *Макет*. Установите курсор в любую ячейку столбца таблицы или выделите несколько смежных столбцов. В счетчике *Ширина столбца таблицы* группы *Размер ячейки* установите требуемое значение. Если выделено несколько столбцов, счетчик *Ширина столбца таблицы* может некорректно работать при установленных режимах автоподбора ширины столбцов *Автоподбор по содержимому* или *Автоподбор по ширине окна*.

Можно подобрать ширину столбцов в зависимости от количества текста в ячейках. Установите курсор в любую ячейку таблицы. Во вкладке *Макет* в группе *Размер ячейки* нажмите кнопку *Автоподбор* и выберите требуемый вариант. Автоподбор по содержимому – автоматическое изменение ширины столбцов в таблице по размеру вводимого текста. Автоподбор по ширине окна — автоматическое изменение ширины столбцов таким образом, чтобы таблица занимала всю ширину страницы. При изменении параметров страницы (поля, ориентация страницы, размер бумаги) ширина таблицы изменяется автоматически. Фиксированная ширина столбцов — прекращение режима автоподбора ширины столбцов. При работе в других вкладках для автоподбора ширины столбцов можно воспользоваться контекстным меню. Щелкните правой кнопкой мыши по любой ячейке столбца таблицы и выберите команду *Автоподбор*. В подчиненном меню выберите требуемый вариант.

Можно выровнять высоту нескольких смежных строк или всех строк таблицы. В первом случае выделите выравниваемые строки, во втором — достаточно установить курсор в любую ячейку таблицы. Во вкладке *Макет* в группе

Размер ячейки нажмите кнопку *Выровнять высоту строк*. Для выделенных строк или всех строк таблицы будет установлено среднее значение высоты, но таким образом, чтобы полностью отображалось содержимое самой высокой строки.

Можно выровнять ширину нескольких смежных столбцов или всех столбцов таблицы. В первом случае выделите выравниваемые столбцы, во втором случае достаточно установить курсор в любую ячейку таблицы. Во вкладке *Макет* в группе *Размер ячейки* нажмите кнопку *Выровнять ширину столбцов*. Для выделенных столбцов или всех столбцов таблицы будет установлено среднее значение ширины. Наведите указатель мыши на маркер таблицы, расположенный около ее правого нижнего угла так, чтобы он превратился в двунаправленную стрелку. Нажмите на левую кнопку мыши и перетащите маркер. В процессе перетаскивания указатель мыши примет вид крестика, а граница таблицы будет отображаться пунктиром.

Объединять можно только смежные ячейки, которые суммарно образуют прямоугольную область. Выделите объединяемые ячейки. Во вкладке *Макет* в группе *Объединить* нажмите кнопку *Объединить ячейки*. При работе в других вкладках для объединения ячеек можно щелкнуть правой кнопкой мыши по выделенным ячейкам и в контекстном меню выбрать команду *Объединить ячейки*. При работе во вкладке *Конструктор* для объединения ячеек можно использовать «ластик». Нажмите кнопку *Ластик* в группе *Нарисовать таблицу*. Указатель мыши примет вид ластика. Щелкните или при нажатой левой кнопке мыши проведите ластиком по границе между объединяемыми ячейками.

Ячейку можно разделить на части, как по вертикали (столбцы), так и по горизонтали (строки). Установите курсор в разделяемой ячейке или выделите несколько разделяемых ячеек. Во вкладке *Макет* в группе *Объединить* нажмите кнопку *Разбить ячейки*. В окне *Разбиение ячеек* укажите требуемое число столбцов и строк, на которые разделяется ячейка. Таблицу можно разделить по горизонтали. Установите курсор в любой ячейке строки, с которой будет начи-

наться новая таблица. Во вкладке *Макет* в группе *Объединить* нажмите кнопку *Разбить таблицу* или нажмите комбинацию клавиш *Ctrl + Shift + Enter*.

5.2 Работа с таблицами в Microsoft Word 2007

MSWord предлагает большое число готовых вариантов оформления (стилей оформления) таблиц. Установите курсор в любую ячейку таблицы. Во вкладке *Конструктор* в группе *Стили таблиц* прокрутите список стилей или щелкните по значку *Дополнительные параметры*, чтобы развернуть список стилей и выберите нужный стиль. При наведении указателя мыши на выбираемый стиль срабатывает функция динамического просмотра, и оформление таблицы отображается в документе. Цветовое оформление стилей определяется темой, выбранной при оформлении документа. При последующем изменении темы будет изменено и цветовое оформление таблицы. После выбора стиля в группе *Параметры стилей таблиц* вкладки *Конструктор* можно установить некоторые особенности его применения к конкретной таблице. Например, можно сделать так, чтобы оформление первой строки и/или последней строки отличалось от оформления остальных строк (флажки *Строка заголовка* и *Строка итогов*). То же самое можно сделать с первым и последним столбцами (флажки *Первый столбец* и *Последний столбец*). Для некоторых стилей можно установить чередующееся оформление строк и столбцов (соответствующие флажки).

Для установки границ таблицы можно использовать элементы группы *Рисование границ* и *Стили таблиц* вкладки *Конструктор*. Выделите таблицу или ее фрагмент, для которого будут устанавливаться границы. Щелкните по списку *Стиль пера* и выберите тип линии. Щелкните по списку *Толщина пера* и выберите толщину линии. Толщина линии устанавливается в пунктах (1 пункт примерно равен 0,35 мм). Щелкните по кнопке *Цвет пера* и в палитре выберите цвет линии. В группе *Стили таблиц* щелкните по стрелке кнопки *Границы* и выберите требуемое расположение границ. Выделите таблицу или ее фрагмент, для которого будет удалена граница. В группе *Стили таблиц* щелкните по

стрелке кнопки *Границы* и выберите режим *Нет границы*. После удаления границ на экране отображается голубая пунктирная сетка таблицы. Эта сетка не выводится на печать. Если сетка не отображается, следует во вкладке *Конструктор* в группе *Стили таблиц* щелкнуть по стрелке кнопки *Границы* и выбрать команду *Отобразить сетку* или в во вкладке *Макет* в группе *Таблица* нажать кнопку *Отобразить сетку*.

Чтобы задать цвет заливки таблицы, можно использовать кнопку *Заливка* группы *Стили таблиц* вкладки *Конструктор* или кнопку *Заливка* группы *Абзац* вкладки *Главная*. Выделите таблицу или ее фрагмент, для которого будет устанавливаться заливка. Щелкните по стрелке кнопки *Заливка* и выберите цвет. При наведении указателя мыши на выбираемый цвет срабатывает функция динамического просмотра, и оформление таблицы отображается в документе. Для удаления заливки в следует выбрать режим *Нет цвета*.

Текст в ячейках таблицы оформляется практически так же, как и обычный текст, но есть и некоторые ограничения. Можно изменять параметры шрифта. Нельзя оформить буквицу в ячейке. Можно изменять выравнивание, отступы и интервалы в абзацах. При этом выравнивание и отступы устанавливаются относительно краев ячейки, в которой находится текст. Можно использовать нумерованные и маркированные списки. При этом элементами списка могут быть как абзацы в ячейках, так и сами ячейки таблицы. Можно устанавливать границы и заливку абзацев и текста в ячейках, но обычно так не делают. Вместо этого используют границы и заливку ячеек. Для оформления текста ячеек можно применять стили. Более того, можно использовать стили таблиц, в параметры которых входят и параметры текста в ячейках. Нельзя в ячейке создать многоколоночный текст. Есть и новые возможности, неприменимые для обычного текста: вертикальное выравнивание, поворот текста и автоподбор параметров шрифта.

Текст и иное содержимое ячейки по умолчанию размещается с выравниванием по левой границе в верхней части ячейки. Если высота ячейки позволяет, можно изменить вертикальное выравнивание. Выделите фрагмент таблицы,

для ячеек которого изменяется выравнивание. Если выравнивание изменяется в одной ячейке, ее можно не выделять, достаточно установить в ней курсор. Во вкладке *Макет* в группе *Выравнивание* выберите один из девяти возможных вариантов. При работе в других вкладках для вертикального выравнивания можно воспользоваться контекстным меню. Щелкните правой кнопкой мыши по выделенному фрагменту таблицы и выберите команду *Выравнивание ячеек*. В подчиненном меню (табло) выберите один из девяти возможных вариантов.

Текст в ячейках таблицы можно располагать не только горизонтально, но и под углом ± 90 сверху вниз или снизу вверх. Выделите фрагмент таблицы, в ячейках которого поворачивается текст. Если направление текста изменяется в одной ячейке, ее можно не выделять, достаточно установить в ней курсор. Во вкладке *Макет* в группе *Выравнивание* нажмите кнопку *Направление текста*. Если текст был горизонтальным, то один щелчок по кнопке расположит текст сверху вниз. Следующий щелчок повернет текст снизу вверх. Следующий щелчок снова расположит текст горизонтально. При работе в других вкладках для изменения направления текста можно воспользоваться контекстным меню. Щелкните правой кнопкой мыши по выделенному фрагменту таблицы и выберите команду *Направление текста*. В окне *Направление текста* выберите один из вариантов.

Можно установить автоматический подбор параметров шрифта (интервал и масштаб) таким образом, чтобы содержимое ячейки располагалось в одну строку от края до края ячейки. Выделите ячейки, для которых требуется установить автоподбор параметров шрифта. Во вкладке *Макет* в группе *Таблица* нажмите кнопку *Свойства*. Во вкладке *Ячейка* окна *Свойства таблицы* нажмите кнопку *Параметры*. В окне *Параметры ячейки* установите флажок *Вписать текст*. Если текст не помещался в ячейке в одну строку, он будет «уплотнен» за счет уменьшения масштаба (ширины) символов. Если ширина текста была меньше ширины ячейки, он будет «разрезан» за счет увеличения интервалов между символами. При установке курсора в ячейку с вписанным текстом там

может отображаться подчеркивание бирюзовым цветом. Обычно оно исчезает при выводе курсора из ячейки и в любом случае не выводится на печать.

Поля ячейки определяют расстояние от границ ячейки до текста в ячейке. По умолчанию установлено левое и правое поле 1,9 мм, а верхнее и нижнее — 0 мм. Увеличение полей ячеек облегчает восприятие таблицы. Уменьшение — делает таблицу более компактной. Установите курсор в любой ячейке таблицы. Во вкладке *Макет* в группе *Выравнивание* нажмите кнопку *Поля ячейки*. В окне *Параметры таблицы* в соответствующих счетчиках установите значение полей.

Сортировка — расположение строк в таблице в определенном порядке. Чаще всего необходимо сортировать строки по данным одного или нескольких столбцов. Сортировку можно производить как по возрастанию, так и по убыванию. При сортировке по возрастанию используется следующий порядок. Сначала числа от наименьшего отрицательного до наибольшего положительного числа, затем различные знаки (+, −, ! и др.), затем английские буквы, затем русские буквы. При сортировке по убыванию порядок заменяется на обратный. При сортировке строка заголовков столбцов обычно не сортируется вместе с остальными строками, но сортируемая таблица может и не иметь заголовков столбцов. Последняя строка включается в диапазон сортировки, поэтому нежелательно сортировать таблицы, содержащие итоговые строки. Как правило, при сортировке упорядочиваются целиком строки, но можно сортировать и отдельные ячейки столбца. Сортировку таблицы можно выполнять при работе во вкладке *Главная* или *Макет*. Установите курсор в любую ячейку таблицы. Нажмите кнопку *Сортировка* в группе *Абзац* вкладки *Главная* или в группе *Данные* вкладки *Макет*. В окне *Сортировка* в списке поля *Сначала по* выберите столбец, по данным которого будет выполняться сортировка таблицы. В списке поля *Тип* можно выбрать тип данных, по которым сортируется таблица, но обычно Word это делает автоматически. Установите переключатель направления сортировки (по возрастанию или по убыванию). Если столбец, по которому выполняется сортировка, содержит повторяющиеся записи, можно выполнить

многоуровневую сортировку. Для этого следует в окне *Сортировка* в списке поля *Затем по* выбрать столбец, по данным которого следует сортировать строки таблицы после сортировки по данным первого столбца. При наличии повторяющихся записей в первом и втором столбцах можно произвести еще одну сортировку. Выделите ячейки столбца, подлежащие сортировке. Нажмите кнопку *Сортировка* в группе *Абзац* вкладки *Главная* или в группе *Данные* вкладки *Макет*. В окне *Сортировка* в списке поля *Тип* можно выбрать тип данных, по которым сортируется таблица, но обычно Word это делает автоматически. Установите переключатель направления сортировки (по возрастанию или по убыванию). Нажмите кнопку *Параметры*. В окне *Параметры сортировки* установите флажок *только столбцы*.

В таблицах Word можно выполнять несложные вычисления с использованием формул. Установите курсор в ячейку, в которой требуется получить результат вычисления. Во вкладке *Макет* в группе *Данные* нажмите кнопку *Формула*. Если эта кнопка не отображается, щелкните по стрелке кнопки *Данные* и, после отображения кнопки, нажмите ее. В окне *Формула* в поле *Формула* введите формулу. Для выбора функции можно воспользоваться списком поля *Вставить функцию*. При желании в списке поля *Формат числа* можно выбрать числовой результат вычисления (числовой с разделителем разрядов, денежный, процентный). В некоторых случаях формула в окне *Формула* может быть записана автоматически. Например, если ячейка находится ниже ячеек с числами, будет записана формула = SUM (ABOVE), что означает суммирование всех вышерасположенных ячеек. Если ячейка находится справа от ячеек с числами, будет записана формула = SUM (LEFT), что означает суммирование всех слева расположенных ячеек. Если требуется именно такое действие, можно применить эти формулы. В противном случае поле следует очистить и ввести формулу самостоятельно. Формула вставляется в ячейку таблицы как специальное поле. При изменении значений в вычисляемых ячейках результат не будет автоматически изменяться. Для пересчета следует выделить ячейку и нажать клавишу *F9*.

Вставленная в документ таблица по умолчанию размещается без обтекания текстом с выравниванием по левому полю страницы. Для выравнивания таблицы по краям страницы достаточно ее выделить и воспользоваться кнопками группы *Абзац* вкладки *Главная*. При таком выравнивании таблицы режим обтекания текстом не изменяется. Изменить режим обтекания и положение таблицы можно, перетащив ее мышью. Наведите указатель мыши на маркер перемещения таблицы, расположенный около ее верхнего левого угла так, чтобы около него отобразилась четырехнаправленная стрелка. Нажмите на левую кнопку мыши и перетащите маркер. В процессе перетаскивания граница таблицы будет отображаться пунктиром.

Можно точно настроить положение таблицы на странице документа. Установите курсор в любой ячейке таблицы. Во вкладке *Макет* в группе *Таблица* нажмите кнопку *Свойства*. Во вкладке *Таблица* окна *Свойства таблицы* в разделе *Обтекание* выберите режим *Вокруг* и нажмите кнопку *Размещение*. В окне *Размещение таблицы* в разделе *По горизонтали* в раскрывающемся списке *относительно* можно выбрать, относительно какого элемента страницы будет выравниваться таблица: колонки, поля или страницы, а в списке поля *положение* можно выбрать расположение таблицы (*Слева, Справа, От центра, Внутри, Снаружи*) или ввести с клавиатуры требуемое значение в сантиметрах. Аналогично можно указать размещение таблицы и по вертикали. Для отказа от режима обтекания таблицы текстом следует установить курсор в любой ячейке таблицы, во вкладке *Макет* в группе *Таблица* нажать кнопку *Свойства*, во вкладке *Таблица* окна *Свойства таблицы* в разделе *Обтекание* выбрать режим *Нет*. При создании таблиц, размещаемых на нескольких страницах, необходимо, чтобы названия столбцов таблицы отображались и печатались на каждой странице документа. Установите курсор в первую строку таблицы или, если в качестве заголовков используется несколько первых строк, выделите их. Во вкладке *Макет* в группе *Данные* нажмите кнопку *Повторить строки заголовков*.

Всю таблицу или ее фрагмент можно преобразовать в текст. Установите курсор в любую ячейку таблицы или выделите преобразуемый диапазон. Во

вкладке *Макет* в группе *Данные* нажмите кнопку *Преобразовать в текст*. В окне *Преобразование в текст* укажите символ, который будет разделять фрагменты текста, сформированные из отдельных ячеек таблицы. Это может быть один из типовых символов (знак абзаца, знак табуляции или точка с запятой) или любой символ по вашему усмотрению, например, точка, пробел, запятая и т. п. В этом случае установите переключатель *другой*, поставьте курсор в рядом расположенное поле и с клавиатуры введите символ разделителя. Если таблица содержит вложенные таблицы, их также можно преобразовать в текст (флажок *вложенные таблицы*). Выбранный разделитель разделяет данные ячеек одной строки; независимо от выбранного разделителя разделителем строк таблицы является знак абзаца.

6. Лекция 6. Обработка информации и алгоритмы

6.1 Алгоритмы и величины

Этапы решения задачи на компьютере. Работа по решению любой задачи с использованием компьютера делится на следующие этапы:

- постановка задачи;
- формализация задачи;
- построение алгоритма;
- составление программы на языке программирования;
- отладка и тестирование программы;
- проведение расчетов и анализ полученных результатов.

Часто эту последовательность называют технологической цепочкой решения задачи на компьютере. Непосредственно к программированию в этом списке относятся пункты 3, 4, 5. На этапе постановки задачи должно быть четко определено, что дано и что требуется найти. Здесь очень важно определить полный набор исходных данных, необходимый для решения задачи. Второй этап — формализация задачи. Здесь чаще всего задача переводится на язык математических формул, уравнений, отношений. Если решение задачи требует

математического описания какого-то реального объекта, явления или процесса, то формализация равносильна получению соответствующей математической модели. Третий этап — построение алгоритма. Опытные программисты часто сразу пишут программы на языках программирования, не прибегая к каким-либо специальным способам описания алгоритмов (блок-схемам, псевдокодам). Однако в учебных целях полезно использовать эти средства, а затем переводить полученный алгоритм на язык программирования.

Первые три этапа — это работа без компьютера. Далее следует собственно программирование на определенном языке в определенной системе программирования. Последний (шестой) этап — это уже использование разработанной программы в практических целях. Выполнение учебных заданий на программирование обычно заканчивается пятым этапом, т. е. доказательством правильности составленной программы. Таким образом, программист должен обладать следующими знаниями и навыками:

- уметь строить алгоритмы;
- знать языки программирования;
- уметь работать в соответствующей системе программирования.

Основой программистской грамотности является развитое алгоритмическое мышление.

Одним из фундаментальных понятий в информатике является понятие алгоритма. Сам термин «алгоритм» пришел из математики. Алгоритм — это последовательность команд управления каким-либо исполнителем. В школьном курсе информатики с понятием алгоритма, с методами построения алгоритмов ученики впервые знакомятся на примерах учебных исполнителей: Робота, Черепашки, Чертежника и др. В учебнике для 9 класса описан графический исполнитель — ГРИС. Эти исполнители ничего не вычисляют. Они создают рисунки на экране, перемещаются в лабиринтах, перетаскивают предметы с места на место. Таких исполнителей принято называть исполнителями, работающими в обстановке. В разделе информатики под названием «Программирование» изучаются методы программного управления работой компьютера. Следовательно,

в качестве исполнителя выступает компьютер. Он работает с величинами — различными информационными объектами: числами, символами, кодами и пр. Поэтому алгоритмы, предназначенные для управления компьютером, принято называть алгоритмами работы с величинами.

Совокупность величин, с которыми работает компьютер, принято называть данными. По отношению к программе данные делятся на исходные, результаты (окончательные данные) и промежуточные данные, которые получаются в процессе вычислений.

Например, при решении квадратного уравнения: $ax^2 + bx + c = 0$ исходными данными являются коэффициенты a , b , c ; результатами — корни уравнения x_1 , x_2 ; промежуточными данными — дискриминант уравнения: $D = b^2 - 4ac$.

Для успешного освоения программирования необходимо усвоить следующее правило: всякая величина занимает свое определенное место в памяти компьютера. Иногда говорят — ячейку памяти. Хотя термин «ячейка», с точки зрения архитектуры современных компьютеров, несколько устарел, однако в учебных целях его удобно использовать.

У всякой величины имеются три основных свойства: имя, значение и тип. На уровне команд процессора величина идентифицируется адресом ячейки памяти, в которой она хранится. В алгоритмах и языках программирования величины делятся на константы и переменные. Константа — неизменная величина, и в алгоритме она представляется собственным значением, например: 15, 34.7, 'к', true. Переменные величины могут изменять свои значения в ходе выполнения программы и представляются символическими именами — идентификаторами, например: X, S2, cod15. Любая константа или переменная занимают ячейку памяти, а значение этих величин определяется двоичным кодом в этой ячейке. Теперь о типах величин — типах данных. С понятием типа данных вы уже встречались, изучая в курсе информатики основной школы электронные таблицы и базы данных. Это понятие является фундаментальным для программирования.

В каждом языке программирования существует своя концепция типов данных, своя система типов. Однако в любой язык входит минимально необходимый набор основных типов данных, к которому относятся целый, вещественный, логический и символьный типы. С типом величины связаны три ее свойства: множество допустимых значений, множество допустимых операций, форма внутреннего представления. Типы констант определяются по контексту (т. е. по форме записи в тексте), а типы переменных устанавливаются в описаниях переменных. Есть еще один вариант классификации данных: классификация по структуре. Данные делятся на простые и структурированные.

Для простых величин (их еще называют скалярными) справедливо утверждение: одна величина — одно значение. Для структурированных: одна величина — множество значений. К структурированным величинам относятся массивы, строки, множества и др. Компьютер — исполнитель алгоритмов.

Тип	Значения	Операции	Внутреннее представление
Целый	Целые положительные и отрицательные числа в некотором диапазоне. Примеры: 23, -12, 387	Арифметические операции с целыми числами: +, -, *, целочисленное деление и остаток от деления. Операции отношений (<, >, = и др.)	Формат с фиксированной запятой
Вещественный	Любые (целые и дробные) числа в некотором диапазоне. Примеры: 2.5, -0.01, 45.0, $3.6 \cdot 10^9$	Арифметические операции: +, -, *, /. Операции отношений	Формат с плавающей запятой
Логический	true (истина) false (ложь)	Логические операции: И (and), ИЛИ (or), НЕ (not). Операции отношений	1 бит: 1 — true; 0 — false
Символьный	Любые символы компьютерного алфавита. Примеры: 'a', '5', '+', '\$'	Операции отношений	Коды таблицы символьной кодировки. 1 символ — 1 байт

Рисунок 3 – Свойства основных типов данных

Как известно, всякий алгоритм (программа) составляется для конкретного исполнителя, в рамках его системы команд. О каком же исполнителе идет речь

в теме «Программирование обработки информации»? Ответ очевиден: исполнителем является компьютер. Точнее говоря, исполнителем является комплекс: компьютер + система программирования (СП). Программист составляет программу на том языке, на который ориентирована СП. Независимо от того, на каком языке программирования будет написана программа, алгоритм решения любой задачи на компьютере может быть составлен из команд (присваивания; ввода; вывода; обращения к вспомогательному алгоритму (подпрограмме); цикла; ветвления).

6.2 Структура алгоритмов

Алгоритмические структуры: следование, ветвление, цикл называют базовыми алгоритмическими структурами. С базовыми алгоритмическими структурами вы познакомились, изучая информатику в 9 классе. Там же для описания структур алгоритмов были использованы два способа: блок-схемы и учебный Алгоритмический язык (АЯ).

Следование — это линейная последовательность действий (рисунок 4).

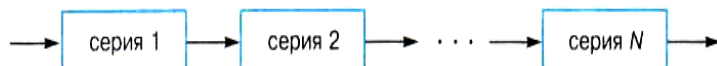
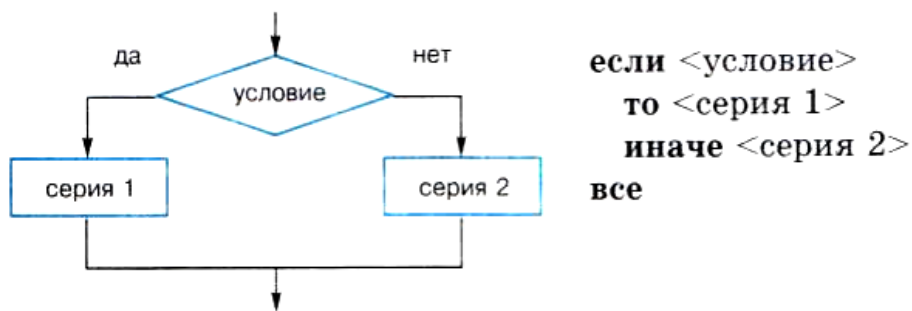


Рисунок 4 – Структура «следование»

В программе на Паскале серия — это либо один отдельный оператор, либо составной оператор: последовательность операторов, заключенная в операторные скобки. Например, в языке Паскаль операторными скобками являются служебные слова `Begin` и `End`.

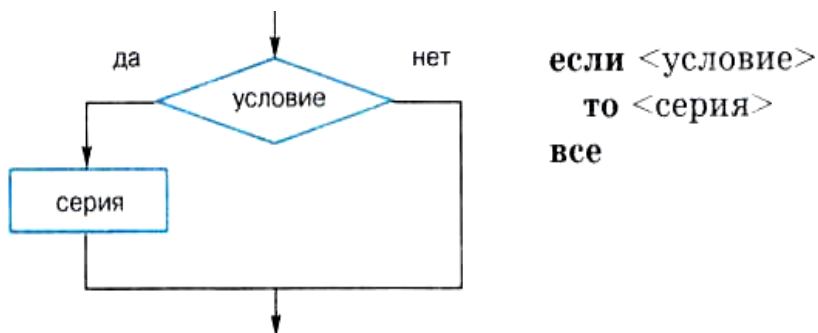
Ветвление — алгоритмическая альтернатива. Управление передается одному из двух блоков в зависимости от истинности или ложности условия. Затем происходит выход на общее продолжение. Вот как изображается ветвление на блок-схеме и АЯ (рисунок 5).



если <условие>
то <серия 1>
иначе <серия 2>
все

Рисунок 5 – Структура «ветвление»

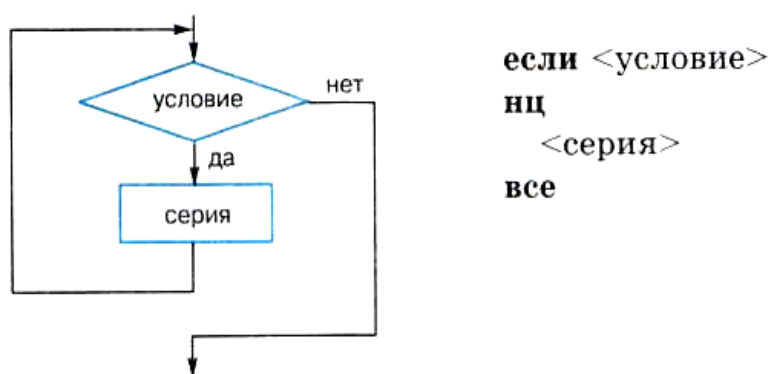
Условие представляет собой утверждение, которое может быть либо истинным, либо ложным. Такое утверждение называется логическим выражением. Неполная форма ветвления имеет место, когда на ветви «нет» пусто (рисунок 6).



если <условие>
то <серия>
все

Рисунок 6 – Неполное ветвление

Цикл — повторение некоторой группы действий по условию. Различают два типа цикла. Первый — цикл с предусловием: «цикл-пока» (рисунок 7).



если <условие>
нц
 <серия>
все

Рисунок 7 – Структура «цикл-пока»

Пока условие истинно, выполняется серия, образующая тело цикла.

Второй тип циклической структуры — цикл с постусловием: «цикл-до» (рисунок 8).

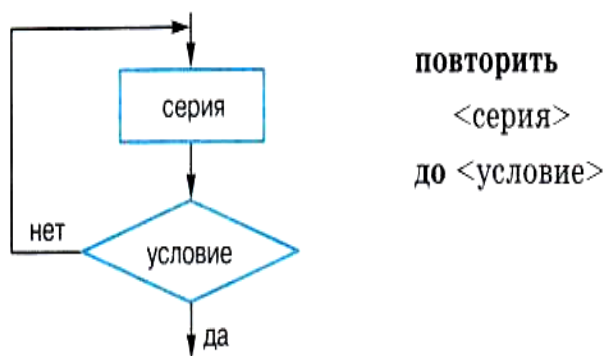


Рисунок 8 – Структура «цикл-до»

Здесь тело цикла предшествует условию цикла. Тело цикла повторяет свое выполнение, если условие ложно. Повторение прекращается, когда условие становится истинным. Теоретически необходимым и достаточным является лишь первый тип цикла — цикл с предусловием. Любой циклический алгоритм можно построить с его помощью. Это более общий вариант цикла, чем «цикл-до». В самом деле, тело цикла-до хотя бы один раз обязательно выполнится, так как проверка условия происходит после завершения его выполнения. А для «цикла-пока» возможен такой вариант, когда тело цикла не выполнится ни разу. Поэтому в любом языке программирования можно было бы ограничиться только «циклом-пока». Однако в ряде случаев применение «цикла-до» оказывается более удобным, и поэтому он используется.

Комбинации базовых структур. Сложный алгоритм состоит из соединенных между собой базовых структур. Соединяться эти структуры могут двумя способами: последовательным и вложенным. Если блок, составляющий тело цикла, сам является циклической структурой, то имеют место вложенные циклы. В свою очередь, внутренний цикл может иметь внутри себя еще один цикл и т. д. В связи с этим вводится представление о глубине вложенности циклов. Точно так же и ветвления могут быть вложенными друг в друга.

Структурный подход требует соблюдения стандарта в изображении блок-схем алгоритмов. Чертить их нужно так, как это делалось во всех приведенных примерах. Каждая базовая структура должна иметь один вход и один выход.

Такие блок-схемы легко читаются. Их структура хорошо воспринимается визуально. Структуре каждого алгоритма можно дать название.

Наглядность структуре описания алгоритма на АЯ придает структуризация внешнего вида текста. Основной используемый для этого прием — сдвиги строк, которые должны подчиняться следующим правилам:

- конструкции одного уровня вложенности записываются на одном вертикальном уровне (начинаются с одной позиции в строке);
- вложенная конструкция записывается смещенной по строке на несколько позиций вправо относительно внешней для нее конструкции.

Такой же способ структуризации используется и в текстах программ (например, на Паскале). Структурное программирование — это не только форма описания алгоритма и программы, но это еще и способ мышления программиста. Размышляя над алгоритмом, нужно стремиться составлять его из стандартных структур. Если использовать строительную аналогию, то структурная методика построения алгоритма подобна сборке здания из стандартных секций, в отличие от складывания по кирпичику.

7 Лекция 7. Pascal — язык структурного программирования

7.1 История Паскаля

Язык программирования Паскаль был создан швейцарским профессором Никлаусом Виртом в 1969 году как язык для обучения студентов структурной методике программирования. Язык получил свое название в честь Блеза Паскаля, изобретателя первого вычислительного механического устройства. Позднее фирма Borland International, Inc (США) разработала систему программирования Турбо Паскаль для персональных компьютеров, которая вышла за рамки учебного применения и стала использоваться для научных и производственных целей. В Турбо Паскаль были внесены некоторые дополнения к базовому стандарту Паскаля, описанному Н. Виртом.

Со временем язык развивался. Начиная с версии 5.5, в Турбо Паскаль вводятся средства поддержки объектно-ориентированного программирования (ООП). В дальнейшем это привело к созданию Object Pascal — языка с возможностями объектно-ориентированного программирования. В начале 1990-х годов объединение элементов ООП в Паскале с визуальной технологией программирования привело к созданию системы программирования Delphi.

7.2 Структура процедурных языков программирования высокого уровня

Во всяком языке программирования определены способы организации данных и способы организаций действий над данными. Кроме того, существует понятие «элементы языка», включающее в себя множество символов (алфавит), служебных слов и других изобразительных средств языка программирования. Несмотря на разнообразие процедурных языков, их изучение происходит приблизительно по одной схеме. Это связано с общностью структуры различных процедурных языков программирования высокого уровня.

Всякий язык программирования образуют три его основные составляющие: алфавит, синтаксис и семантика. Алфавит — это множество символов, допустимых в записи текстов программ. Синтаксис — это правописание языковых конструкций (имен, констант, выражений, операторов и пр.). Семантика — это смысловое содержание языковой конструкции.

Соблюдение правил в языке программирования должно быть более строгим, чем в разговорном языке. Человеческая речь содержит значительное количество избыточной информации. Слушающий или читающий человек может додумать, дополнить, исправить ошибки в воспринимаемом тексте. Компьютер же — автомат, воспринимающий всё буквально. В текстах программ нет избыточности, компьютер сам не исправит даже очевидной (с точки зрения человека) ошибки. Он может лишь указать на место, которое «не понял», и вывести замечание о предполагаемом характере ошибки. Исправить же ошибку должен программист.

7.3 Структура программы на Паскале

По определению стандартного Паскаля, программа состоит из заголовка программы и тела программы (блока), за которым следует точка — признак конца программы. В свою очередь, блок содержит разделы описаний (меток, констант, типов, переменных, подпрограмм) и раздел операторов.

```
Program <имя программы>;  
Label <раздел меток>;  
Const <раздел констант>;  
Type <раздел типов>;  
Var <раздел переменных>;  
Procedure (Function) <раздел подпрограмм>;  
Begin  
    <раздел операторов>  
End.
```

Раздел операторов имеется в любой программе и является основным. Предшествующие разделы носят характер описаний и не все обязательно присутствуют в каждой программе. В Турбо Паскале, в отличие от базового стандарта Паскаля, возможно отсутствие заголовка программы; разделы `Const`, `Type`, `Var`, `Label` могут следовать друг за другом в любом порядке и повторяться в разделе описаний сколько угодно раз. Алфавит языка состоит из множества символов, включающих в себя буквы, цифры и специальные символы.

Латинские буквы: от *A* до *Z* (заглавные) и от *a* до *z* (строчные).

Цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Специальные символы: + - * / = <> [] . , () : ; { } ^ @ \$ #.

Следующие комбинации специальных символов являются единичными символами (их нельзя разделять пробелами):

```
:=   знак присваивания;  
<=   меньше или равно;  
>=   больше или равно;  
(* *) ограничители комментариев (наряду с { });  
<>   не равно;  
(. .) эквивалент [ ].
```

Пробелы — символ пробела (код ASCII 32) и все управляющие символы кода ASCII (от 0 до 31).

Служебные слова. К спецсимволам относятся и служебные слова, смысл которых определен однозначно. Служебные слова не могут быть использованы для других целей. С точки зрения языка, они являются единичными элементами алфавита. Вот некоторые служебные слова: Program, Var, array, If, Do, While.

Идентификаторы. Идентификатором называется символическое имя определенного программного объекта. Такими объектами являются: имена констант, переменных, типов данных, процедур и функций, программ. Идентификатор — это любая последовательность букв и цифр, начинающаяся с буквы. К буквам приравнивается также знак подчеркивания. Длина идентификатора может быть произвольной, но значащими являются только первые 63 символа.

Комментарии. Следующие конструкции представляют собой комментарии и поэтому пропускаются компилятором:

{любой текст, не содержащий символ «фигурная скобка»}
(любой текст, не содержащий символы «звездочка, круглая скобка»*)*
//последующий текст до конца строки.

Буквы русского алфавита употребляются только в комментариях, символьных и текстовых константах.

Концепция типов данных является одной из центральных в любом языке программирования. Как вы знаете, с типом величины связаны три ее свойства: форма внутреннего представления, множество принимаемых значений и множество допустимых операций. Паскаль характеризуется большим разнообразием типов данных, отраженным на рисунке 9.



Рисунок 9 – Система типов данных языка Паскаль

Тип данных называется порядковым, если он состоит из счетного количества значений, которые можно пронумеровать. Каждый тип имеет свой идентификатор. На рисунке 10 представлена информация о простых типах данных, определенных в Турбо Паскале и последующих диалектах языка. Для вещественных типов в скобках указано количество сохраняемых значащих цифр мантиссы в десятичном представлении числа.

Имя типа	Длина в байтах	Диапазон (множество) значений	Десятичных цифр в мантиссе
Целочисленные типы			
Integer	2	-32768..32767	
Byte	1	0..255	
Word	2	0..65535	
Shortint	1	-128..127	
Longint	4	-2147483648..2147483647	
Вещественные типы			
Real	6	$2,9 \cdot 10^{-39} - 1,7 \cdot 10^{38}$	11-12
Single	4	$1,5 \cdot 10^{-45} - 3,4 \cdot 10^{38}$	7-8
Double	8	$5 \cdot 10^{-324} - 1,7 \cdot 10^{308}$	15-16
Extended	10	$3,4 \cdot 10^{-4932} - 1,1 \cdot 10^{4932}$	19-20
Логический тип			
Boolean	1	true, false	
Символьный тип			
Char	1	Все символы 8-разрядной кодировки	

Рисунок 10 – Типы данных

Типы пользователя. Одна из принципиальных возможностей Паскаля состоит в том, что пользователю разрешается определять свои типы данных. Типы пользователя всегда базируются на стандартных типах данных Паскаля. Для описания типов пользователя в Паскале существует раздел типов, начинающийся со служебного слова *Type*. К простым типам пользователя относятся перечислимый и интервальный типы данных.

Перечислимый тип задается непосредственно перечислением (списком) всех значений, которые может принимать переменная данного типа:

Type <имя типа> = (<список значений>)

Определенное таким образом имя типа затем используется для описания переменных. Например:

```
Type Gaz = (C, O, N, F);  
        Metal = (Fe, Co, Na, Cu, Zn);  
Var G1, G2, G3: Gaz;  
        Met1, Met2: Metall;  
        Day: (Sun, Mon, Tue, Wed, Thu, Fri, Sat);
```

Здесь *Gaz* и *Metal* — имена перечислимых типов, которые ставятся в соответствие переменным *G1*, *G2*, *G3* и *Met 1*, *Met2*. Переменной *Day* назначается перечислимый тип, которому не присвоено имени. Значения, входящие в перечислимый тип, являются константами. Действия над ними подчиняются правилам, применимым к константам. Каждое значение в перечислимом типе занимает в памяти 2 байта, поэтому число значений этого типа не должно превышать 65 535. Перечислимый тип — упорядоченное множество. Его элементы пронумерованы, начиная от 0 в порядке следования в описании.

В программе, в которой присутствует данное выше описание переменной *Day*, возможен такой фрагмент:

```
If Day=Sun Then Write('Ура! Сегодня выходной!');
```

Ограниченный тип задается как упорядоченное ограниченное подмножество некоторого порядкового типа: <константа 1>..*константа 2*>

Порядковый номер первой константы не должен превышать номера второй константы в соответствующем базовом типе. При исполнении программы автоматически контролируется принадлежность значений переменной ограниченного типа установленному диапазону. При выходе из диапазона исполнение программы прерывается. Например:

```
Type Numbers = 1..31;  
        Alf = 'A'..'Z';  
Var Data: Numbers;  
        Bukva: Alf;
```

Структурные типы. Особенностью Паскаля является то, что в нем структуры данных рассматриваются как типы — структурные типы данных. Одна величина простого типа представляет собой одно значение: целое число, веще-

ственное число, символ и пр. Одна величина структурного типа представляет собой совокупность множества значений; примеры — числовой массив, символьная строка и пр.

8 Лекция 8. Операции, функции, выражения языка Pascal

8.1. Операции, функции, выражения

К числовым типам данных относятся группы вещественных и целочисленных типов. К ним применимы арифметические операции и операции отношений. Операции над данными бывают унарными (применимые к одному операнду) и бинарными (применимые к двум операндам). Бинарные арифметические операции стандартного Паскаля описаны на рисунке 11.

Обозначение	Операция	Типы операндов	Тип результата	Пример
+	сложение	integer, real	integer, real	$X + Y$
-	вычитание	integer, real	integer, real	$X - Y$
*	умножение	integer, real	integer, real	$X * Y$
/	деление действительных чисел	integer, real	real	X / Y
div	целочисленное деление	integer	integer	$I \text{ div } 2$
mod	остаток целочисленного деления	integer	integer	$I \text{ mod } 6$

Рисунок 11 – Бинарные операции Паскаля

Унарная арифметическая операция в Паскале одна. Это операция изменения знака. Ее формат:

– *<величина>*

8.2 Стандартные функции и процедуры

В Паскале существует большое количество стандартных функций и процедур, к которым программист может обращаться в своих программах. Наиболее часто используются математические функции, например: $\text{sqrt}(x)$ — квадратный корень, $\text{abs}(x)$ — абсолютная величина, $\text{sin}(x)$ и др. Часто используемые стандартные процедуры: $\text{Read}(\dots)$ — процедура ввода, $\text{Write}(\dots)$ — процедура вывода данных.

Стандартные функции и процедуры являются внешними подпрограммами по отношению к вызывающей их программе. Они объединены в модули, которые подключаются к основной программе и становятся доступными для использования. Наиболее часто используемые подпрограммы объединены в модуль под названием SYSTEM. Этот модуль подключается к программе автоматически. Рисунок 12 содержит описания стандартных математических функций Паскаля.

Стандартные математические функции языка Паскаль	
Вызов функции	Функция
$\text{Abs}(x)$	Абсолютное значение (модуль)
$\text{Arctan}(x)$	Арктангенс
$\text{Cos}(x)$	Косинус
$\text{Exp}(x)$	e^x — экспонента
$\text{Frac}(x)$	Дробная часть
$\text{Int}(x)$	Целая часть (возвращает результат вещественного типа)
$\text{Trunc}(x)$	Целая часть (возвращает результат целого типа)
$\text{Ln}(x)$	Натуральный логарифм
$\text{Sin}(x)$	Синус
$\text{Sqr}(x)$	Возведение в квадрат
$\text{Sqrt}(x)$	Корень квадратный

Рисунок 12 – Стандартные математические функции Паскаля

Для подключения других модулей необходимо в начале программы (после заголовка) записать строку:

Uses <имя модуля>

Для управления символьным выводом на экран используется стандартный модуль CRT. К программе он подключается командой:

Uses CRT

В дальнейшем из этого модуля мы будем использовать *процедуру очистки экрана* для символьного вывода, обращение к которой производится оператором *ClrScr*. Арифметическое выражение задает порядок выполнения действий над числовыми величинами. Арифметические выражения содержат числовые константы и переменные, арифметические операции, функции, круглые скобки. Одна константа или одна переменная — простейшая форма арифметического выражения.

Например, рассмотрим математическое выражение:

$$\frac{2a + \sqrt{0,5 \sin(x + y)}}{0,2c - \ln(x - y)}$$

. На Паскале оно выглядит так:

$$(2*A + Sqrt(0.5*sin(X + Y))) / (0.2*C - ln(X - Y))$$

Для того чтобы правильно записывать арифметические выражения, нужно соблюдать следующие правила.

1. Все символы пишутся в строчку на одном уровне. Проставляются все знаки операций (нельзя пропускать знак *).

2. Не допускаются два следующих подряд знака операций.

(Нельзя: $A+-B$; можно: $A+(-B)$.)

3. Операции с более высоким приоритетом выполняются раньше операций с меньшим приоритетом. Порядок убывания приоритетов:

вычисление функции;

унарная операция смены знака (–);

*, /,

div, mod;+, -.

4. Несколько записанных подряд операций одинакового приоритета выполняются последовательно слева направо.

5. Часть выражения, заключенная в скобки, вычисляется в первую очередь. (Например, в выражении $(A+B) * (C-D)$ умножение производится после сложения и вычитания.)

Не следует записывать выражения, не имеющие математического смысла, например: деление на нуль, логарифм отрицательного числа и т. п. В Паскале нет операции или стандартной функции возведения числа в произвольную степень. Для вычисления x^y рекомендуется поступать следующим образом:

а) если y — целое положительное значение, то его степень вычисляется через умножение; например $x^3 \rightarrow x*x*x$; большие степени следует вычислять умножением в цикле;

б) если y — целое отрицательное число, то степень вычисляется так:

$$x^y = (1/x)^{|y|}; \text{ а при } y = 0: x^0 = 1.$$

в) если y — вещественное значение, не равное нулю, то используется следующая математическая формула: $x^y = e^{y \ln(x)}$

На Паскале получим арифметическое выражение: $\text{exp}(Y * \ln(x))$

Очевидно, что в этом случае не допускается нулевое или отрицательное значение x . Для целого y такого ограничения нет. Выражение имеет целочисленный тип, если в результате его вычисления получается величина целочисленного типа. Выражение имеет вещественный тип, если результатом его вычисления является вещественная величина.

8.3 Оператор присваивания, ввод и вывод данных

Присваивание — это действие, в результате которого переменная величина получает определенное значение. В программе на Паскале существуют три способа присваивания значения переменной:

- 1) оператор присваивания;
- 2) оператор ввода;
- 3) передача значения через параметры подпрограммы.

Оператор присваивания имеет следующий формат:

<переменная>:=<выражение>

Сначала вычисляется выражение, затем полученное значение присваивается переменной. Типы переменной и выражения должны совпадать. Из этого правила есть одно исключение: переменной вещественного типа можно присваивать значение целочисленного выражения. В таком случае значение целого числа преобразуется к формату с плавающей запятой и присвоится вещественной переменной.

Под вводом понимается передача данных с внешнего устройства компьютера в оперативную память. При выводе данные передаются из оперативной памяти на внешнее устройство. Операция ввода называется чтением и выполняется с помощью оператора *Read*. Вывод называется записью, и для его выполнения используется оператор *Write*. К внешним устройствам относятся устройства ввода и вывода (клавиатура, монитор, принтер и др.) и устройства внешней памяти (магнитные и оптические диски, флеш-память и др.). Данные на внешних устройствах организованы в файлы.

Для внешних запоминающих устройств (ВЗУ) файл — это поименованная область памяти этого устройства. В файлы на ВЗУ можно записывать данные по команде *Write* и можно читать данные из файлов по команде *Read*. На одном устройстве ВЗУ может храниться множество файлов одновременно. Правила именования файлов на ВЗУ определяются операционной системой. Имена для файлов, создаваемых пользователем, задает сам пользователь.

Устройства ввода с клавиатуры и вывода на экран монитора являются однофайловыми устройствами. Считается, что с клавиатурой связан один системный файл с именем *INPUT*. Поэтому, ввод с клавиатуры равнозначен чтению из файла *INPUT*. С монитором связан системный файл, который называется *OUTPUT*. Вывод на экран — это запись данных в файл *OUTPUT*. Ввод с клавиатуры производится путем обращения к стандартной процедуре *Read* в следующем формате: *Read(<список ввода>)*

Чтение происходит из системного файла *INPUT*, всегда доступного для любой программы. Элементами списка ввода могут быть переменные символьного типа, числовых типов и строковые переменные. Например: *Read (a, b, c, d)*

При выполнении этого оператора происходит прерывание исполнения программы, после чего пользователь должен набрать на клавиатуре значения переменных *a, b, c, d*, отделяя их друг от друга пробелами. При этом вводимые значения высвечиваются на экране. В конце нажимается клавиша *Enter*. Значения следует вводить в строгом соответствии с синтаксисом Паскаля. Если в программе имеется несколько подряд идущих операторов *Read*, то данные для них можно вводить последовательно (на экране отражаются в одной строке) и лишь в конце ввода нужно нажать клавишу *Enter*.

Вывод на экран производится по оператору обращения к стандартной процедуре:

Write (<список вывода>)

Здесь элементами списка вывода могут быть выражения различных типов (в частности, константы и переменные).

Например: *Write ('Сумма A, ' + ', B, '=' , A+B)*

Если, например, $A = 5$ и $B = 7$, то на экране получим:

Сумма A+B=12

При выводе на экран нескольких значений в строку они не отделяются друг от друга пробелами. Программист сам должен позаботиться о таком разделении. В приведенном примере предусмотрен пробел после слова «Сумма».

Второй вариант процедуры вывода на экран:

WriteLn (<список вывода>)

Writeln — «писать строку». Его действие отличается от оператора *Write* тем, что после вывода последнего в списке значения происходит перевод курсора к началу следующей строки. Оператор *WriteLn*, записанный без параметров, вызывает перевод строки.

8.4 Линейная программа

Следование — простейшая алгоритмическая структура. Программа, реализующая следование, называется линейной программой. В линейной программе могут присутствовать только операторы присваивания, ввода, вывода и обращения к процедурам. Заметим, что операторы Read и Write являются обращениями к стандартным процедурам Паскаля. Одним из обязательных условий хорошего стиля программирования является организация диалога между компьютером и пользователем. Такое диалоговое взаимодействие называется интерактивным интерфейсом.

Пример 1. Составим линейную программу, по которой в диалоге будут вводиться два целых числа и вычисляться их произведение.

```
Program Multiply;
Var A, B, AB: integer;
Begin
    Write('A= '); ReadLn(A);
    Write('B= '); ReadLn(B);
    AB:=A*B;
    Write(A, '*', B, '=', AB)
End.
```

Тестирование этой программы отразится на экране следующим образом.

A= 13. B= 28. 13*28=364. Числа 13 и 28 вводятся пользователем с клавиатуры, всё остальное автоматически выводится по программе.

Пример 2. Дано натуральное трехзначное число. Требуется вычислить сумму его цифр. Например, если дано число 325, то в результате должно получиться: $3 + 2 + 5 = 10$.

```
Program SumCifr;
Var X, Sum: integer;
Begin
    Write('Введите трехзначное число: '); ReadLn(X);
    Sum:=0;
    Sum:=Sum + X mod 10;
    X:=X div 10;
    Sum:=Sum + X mod 10;
    X:=X div 10;
    Sum:=Sum + X;
    Write('Сумма цифр = ', Sum)
End.
```

В этой программе использованы две операции целочисленной арифметики: *div* — целочисленное деление и *mod* — остаток от целочисленного деления (рисунок 13). Остаток от деления на 10 (*mod*) выделяет младшую цифру числа, а целочисленное деление на 10 (*div*) отбрасывает младшую цифру.

Чтобы лучше понять работу программы, выполним ее трассировку. В курсе 9 класса вам уже приходилось строить трассировочные таблицы. Для программы *SumCifr* таблица будет выглядеть следующим образом:

№	Команда	X	Sum
1	readln(X)	325	—
2	Sum:=0		0
3	Sum:=Sum + X mod 10		5
4	X:=X div 10	32	
5	Sum:=Sum + X mod 10		7
6	X:=X div 10	3	
7	Sum:=Sum + X		10
8	Write('Сумма цифр = ', Sum)		10

Рисунок 13 –Трассировки программы

Выполнение программы на компьютере приводит к такому же результату. Заметим, что эту задачу можно решить с помощью всего одного оператора присваивания: $Sum := X \bmod 10 + X \div 10 \bmod 10 + X \div 100$

Присваивание, ввод, вывод
Оператор присваивания: <переменная>:=<выражение> Тип переменной и выражения должны совпадать. Исключение: вещественной переменной можно присваивать значение целого выражения
Ввод — передача данных с внешнего устройства в ОЗУ
Ввод с клавиатуры: Read(<список ввода>) или ReadLn(<список ввода>)
Вывод — передача данных из ОЗУ на внешнее устройство
Вывод на экран: Write(<список вывода>) или WriteLn(<список вывода>)
Линейная программа состоит из операторов ввода, вывода, присваивания и обращения к процедурам

Рисунок 14 – Система основных понятий

9 Лекция 9. Логические величины, операции, выражения языка Pascal

9.1 Логические операции

С элементами математической логики вы уже встречались в курсе информатики основной школы, изучая способы записи запросов к базе данных и условной функции *ЕСЛИ* в электронных таблицах, основы алгоритмизации и программирования. Повторим основные понятия логики с целью дальнейшего углубления ваших знаний в использовании ее для программирования.

К числу основных понятий логики относятся: высказывание, логическая величина, логические операции, логические выражения и формулы.

Высказывание (суждение) — это повествовательное предложение, в котором что-либо утверждается или отрицается. По поводу любого высказывания можно сказать, истинно оно или ложно. Например, высказывание «На улице идет дождь» будет истинным или ложным в зависимости от состояния погоды в данный момент. Истинность высказывания «Значение A больше, чем B », записанного в форме неравенства: $A > B$, будет зависеть от значений переменных A и B .

Логические величины — понятия, выражаемые словами: *ИСТИНА*, *ЛОЖЬ* (true, false). Следовательно, истинность высказываний выражается через логические величины.

Логическая константа: *ИСТИНА* или *ЛОЖЬ*.

Логическая переменная: символически обозначенная логическая величина. Следовательно, если известно, что A , B , X , Y и др. — переменные логические величины, то, значит, они могут принимать значения только *ИСТИНА* или *ЛОЖЬ*.

Логическое выражение — простое или сложное высказывание. Сложное высказывание строится из простых с помощью логических операций (связок).

Конъюнкция (логическое умножение). В русском языке она выражается союзом *И*. В математической логике используются знаки $\&$ или \wedge . Конъюнкция

— двухместная операция; записывается в виде: $A \& B$. Значением такого выражения будет *ЛОЖЬ*, если значение хотя бы одного из операндов ложно.

Дизъюнкция (логическое сложение). В русском языке этой связке соответствует союз **ИЛИ**. В математической логике она обозначается знаком \vee . Дизъюнкция — двухместная операция; записывается в виде: $A \vee B$. Значением такого выражения будет *ИСТИНА*, если значение хотя бы одного из операндов истинно.

Отрицание. В русском языке этой связке соответствует частица **НЕ** (в некоторых высказываниях применяется оборот «неверно, что ...»). Отрицание — унарная (одноместная) операция; записывается в виде: $\neg A$ или \bar{A} .

Правила выполнения рассмотренных логических операций отражены в следующей таблице, которая называется таблицей истинности логических операций (здесь **И** означает «истина», **Л** — «ложь»):

A	B	$\neg A$	$A \& B$	$A \vee B$
И	И	Л	И	И
И	Л	Л	Л	И
Л	И	И	Л	И
Л	Л	И	Л	Л

Рисунок 15 – Правила выполнения логических операций

Логическая формула — формула, содержащая лишь логические величины и знаки логических операций. Результатом вычисления логической формулы является *ИСТИНА* или *ЛОЖЬ*. Последовательность выполнения операций в логических формулах определяется старшинством операций. В порядке убывания старшинства логические операции расположены так: отрицание, конъюнкция, дизъюнкция. Кроме того, на порядок выполнения операций влияют скобки, которые можно использовать в логических формулах.

Например: $(A \& B) \vee (\neg A \& B) \vee (\neg A \& \neg B)$.

Пример. Вычислить значение логической формулы: $\neg X \& Y \vee X \& Z$, если логические переменные имеют следующие значения: $X = \text{ЛОЖЬ}$, $Y = \text{ИСТИНА}$.

$HA, Z = \text{ИСТИНА}$. Решение. Отметим цифрами сверху порядок выполнения

операций в формуле: $\overset{1}{\neg}X \ \& \ \overset{2}{Y} \ \vee \ \overset{4}{X} \ \& \ \overset{3}{Z}$.

Используя таблицу истинности, вычислим формулу по шагам:

- 1) ЛОЖЬ = ИСТИНА;
- 2) ИСТИНА & ИСТИНА = ИСТИНА;
- 3) ЛОЖЬ & ИСТИНА = ЛОЖЬ;
- 4) ИСТИНА \vee ЛОЖЬ = ИСТИНА. Ответ: ИСТИНА.

9.2 Логические функции на области числовых значений

Алгебра чисел пересекается с алгеброй логики в тех случаях, когда приходится проверять принадлежность значений алгебраических выражений некоторому множеству. Например, принадлежность значения числовой переменной X множеству положительных чисел выражается через высказывание: « X больше нуля». Символически это записывается так: $X > 0$. В алгебре такое выражение называют неравенством. В логике — отношением. Отношение $X > 0$ может быть истинным или ложным. Если X — положительная величина, то оно истинно, если отрицательная, то ложно. В общем виде отношение имеет следующую структуру: < выражение 1 >< знак отношения >< выражение 2 >

Здесь выражения 1 и 2 — некоторые математические выражения, принимающие числовые значения. В частном случае выражение может представлять собой одну константу или одну переменную величину. Знаки отношений могут быть следующими:

- = — равно;
- \neq — не равно;
- \geq — больше или равно;
- \leq — меньше или равно;
- > — больше;
- < — меньше.

Например:

$$x = 5; \quad a + b \neq x - 1; \quad b^2 - 4ac \geq 0; \quad \sin(x) < x/2.$$

Итак, отношение — это простое высказывание, а значит, логическая величина. Оно может быть как постоянной: $5 > 0$ — всегда ИСТИНА, $3 * 6 : 2$ — всегда ЛОЖЬ; так и переменной: $a < b$, $x + 1 = c - d$. Если в отношение входят переменные числовые величины, то и значение отношения будет логической переменной. Отношение можно рассматривать как логическую функцию от числовых аргументов. Например: $F(x) = (x > 0)$ или $P(x, y) = (x < y)$. Аргументы определены на бесконечном множестве действительных чисел, а значения функции — на множестве, состоящем из двух логических величин: ИСТИНА, ЛОЖЬ. Логические функции от числовых аргументов еще называют термином предикат. В алгоритмах предикаты играют роль условий, по которым строятся ветвления и циклы. Предикаты могут быть как простыми логическими функциями, не содержащими логических операций, так и сложными, содержащими логические операции.

Пример. Записать предикат (логическую функцию) от двух вещественных аргументов X и Y , который будет принимать значение ИСТИНА, если точка на координатной плоскости с координатами X и Y лежит внутри единичной окружности с центром в начале координат (рисунок 16).

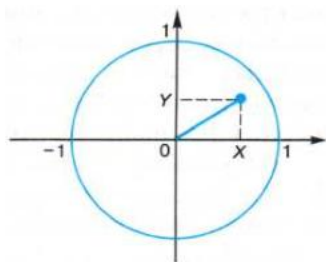


Рисунок 16 – Единичная окружность

Из геометрических соображений понятно, что для всех точек, лежащих внутри единичной окружности, будет истинным значение следующей логической функции: $F(X, Y) = (X^2 + Y^2 < 1)$. Для значений координат точек, лежащих на окружности и вне ее, значение функции F будет ложным.

9.3 Логические выражения языка Паскаль

Мы знаем, что в Паскале имеется логический тип данных. Логические константы: *true* (истина), *false* (ложь). Логические переменные: описываются с типом *Boolean*. Операции отношения: осуществляют сравнение двух операндов и определяют, истинно или ложно соответствующее отношение между ними. Знаки операций отношения: = (равно), <> (не равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно).

Логические операции: *not* — отрицание, *and* — логическое умножение (конъюнкция), *or* — логическое сложение (дизъюнкция), *xor* — исключающее ИЛИ. Логическое выражение может состоять из логических констант и переменных, отношений, логических операций. Логическое выражение принимает значение *true* или *false*.

A	B	not A	A and B	A or B	A xor B
T	T	F	T	T	F
T	F	F	F	T	T
F	T	T	F	T	T
F	F	T	F	F	F

Рисунок 17 – Таблица истинности для операций (Т — *true*; F — *false*)

Например, логическая формула $\neg X \& Y \vee X \& Z$ на Паскале запишется в виде следующего логического выражения:

not X and Y or X and Z, где *X*, *Y*, *Z* — переменные типа *Boolean*.

Логические операции располагаются в следующем порядке по убыванию старшинства (приоритета): 1) **not**, 2) **and**, 3) **or**, **xor**. Операции отношения имеют самый низкий приоритет. Поэтому если операндами логической операции являются отношения, то их следует заключать в круглые скобки. Например, математическому неравенству $1 \leq X \leq 50$ соответствует следующее логическое выражение: **(1 <= X) and (X <= 50)**

Логическая функция *odd* (*x*) принимает значение *true*, если значение целочисленного аргумента *x* является нечетным, иначе — *false*.

Для правильной записи сложного логического выражения (предиката) нужно учитывать относительные приоритеты арифметических, логических операций и операций отношений, поскольку все они могут присутствовать в логическом выражении. По убыванию приоритета операции располагаются в следующем порядке.

1. Арифметические операции: - (минус унарный), *, /, +, -
2. Логические операции: not, and, or, xor
3. Операции отношения: =, <>, >, <, >=, <=

Еще раз обратите внимание, что в логическом выражении, соответствующем предикату из примера 3: $(Y > -X) \text{ and } (Y < 1) \text{ and } (Y > X * X)$, операции отношения заключены в скобки, поскольку они младше логических операций, а выполнение должны раньше.

Логические величины, операции, выражения			
Базовые понятия логики			
Высказывание (суждение)	Логическая величина	Логическое выражение	Основные логические операции
Некоторое утверждение, которое может быть либо истинным, либо ложным	Принимает одно из двух значений: ИСТИНА, ЛОЖЬ	Простое или сложное высказывание, содержащие логические связи (операции)	- конъюнкция (логическое умножение) — И; - дизъюнкция (логическое сложение) — ИЛИ; - инверсия (отрицание) — НЕ
Логические величины в Паскале: true — ИСТИНА, false — ЛОЖЬ			
Логические операции в языке Паскаль			
not — отрицание	and — конъюнкция	or — дизъюнкция	xor — исключающее ИЛИ
Изменяет логическое значение операнда на противоположное	Равно true, если оба операнда true	Равно false, если оба операнда false	Равно true, если операнды имеют разные значения
Старшинство операций в логическом выражении на Паскале			
1. Арифметические операции: 1) - (минус унарный) 2) *, / 3) +, -	2. Логические операции: 1) not 2) and 3) or, xor	3. Операции отношения: =, <>, >, <, >=, <=	

Рисунок 18 – Система основных понятий

10 Лекция 10. Программирование ветвлений

10.1 Полные и неполные ветвления

Алгоритмическая структура ветвления программируется в Паскале с помощью условного оператора *If*. Вспомним его формат.

Полное ветвление:

If < логическое выражение > **Then** < оператор 1 >

Else < оператор 2 >

Неполное ветвление: If < логическое выражение > **Then** < оператор >

То, что в алгоритмах называется условием, в Паскале является логическим выражением, которое вычисляется в первую очередь. Если его значение равно true, то будет выполняться < оператор 1 > (после Then), если — false, то < оператор 2 > (после Else) для полной формы или оператор, сразу следующий после условного, для неполной формы (без Else). На ветвях может быть как простой оператор, так и составной — серия операторов в операторных скобках Begin, End.

Пример 1. По длинам трех сторон треугольника a, b, c требуется вычислить его площадь. Для решения задачи используется формула Герона

$$\sqrt{p(p-a)(p-b)(p-c)},$$

где $p = (a + b + c)/2$ — полупериметр треугольника.

```
Program Geron;
Var A, B, C, P, S : Real;
Begin
  WriteLn('Введите длины сторон треугольника: ');
  Write('a = '); ReadLn(A);
  Write('b = '); ReadLn(B);
  Write('c = '); ReadLn(C);
  If (A>0) and (B>0) and (C>0) and (A+B>C)
    and (B+C>A) and (A+C>B)
  Then Begin
    P := (A+B+C)/2;
    S := Sqrt(P*(P-A)*(P-B)*(P-C));
    WriteLn('Площадь=', S)
  End
  Else WriteLn('Неверные исходные данные')
End.
```

Исходные данные должны удовлетворять основному соотношению для сторон треугольника — длина каждой стороны должна быть меньше суммы длин двух других сторон, и длины сторон не могут быть отрицательными величинами. Имея возможность в одном условном операторе записывать достаточно сложные логические выражения, используя логические операции, мы можем сразу «отфильтровать» все варианты неверных исходных данных.

Пример 2. Требуется перевести пятибалльную оценку в ее наименование: 5 — «отлично», 4 — «хорошо», 3 — «удовлетворительно», 2 — «неудовлетворительно». Блок-схема алгоритма приведена на рисунке 19.

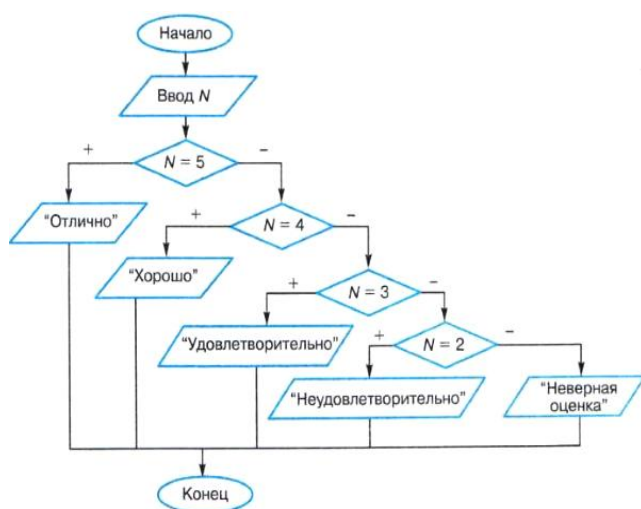


Рисунок 19– Алгоритм перевода числовой оценки в словесную

Этот алгоритм имеет структуру вложенных ветвлений и может быть запрограммирован с использованием условного оператора **If** следующим образом:

```

Program Marks_1;
Var N: Integer;
Begin
  WriteLn('Введите оценку:');
  ReadLn(N);
  If N=5
  Then WriteLn('Отлично')
  Else If N=4
  Then WriteLn('Хорошо')
  Else If N=3
  Then WriteLn('Удовлетворительно')
  Else If N=2
  Then WriteLn('Неудовлетворительно')
  Else WriteLn('Неверная оценка')
End.
  
```

Пример 3. Решение рассмотренной в предыдущем примере задачи можно запрограммировать с помощью одного оператора выбора, имеющегося в языке Паскаль. Вот как будет выглядеть такая программа:

```
Program Marks_2;  
Var N: Integer;  
Begin  
  WriteLn('Введите оценку:');  
  ReadLn(N);  
  Case N of  
    5: WriteLn('Отлично');  
    4: WriteLn('Хорошо');  
    3: WriteLn('Удовлетворительно');  
    2: WriteLn('Неудовлетворительно');  
    Else WriteLn('Неверная оценка')  
  End;
```

Оператор выбора имеет следующий формат:

```
Case < селектор > of  
< список констант 1 > : < оператор 1 > ;  
...  
< список констант N > : < оператор N > ;  
Else <оператор>  
End.
```

Здесь < селектор > — это выражение любого порядкового типа; < константа > — постоянная величина того же типа, что и селектор; < оператор > — любой простой или составной оператор. Выполнение оператора выбора происходит так: вычисляется выражение-селектор; затем в списках констант ищется такое значение, которое совпадает с полученным значением селектора; далее исполняется оператор, помеченный данной константой. Если такой константы не найдено, то происходит переход к выполнению оператора, следующего после слова Else.

Пример 4. В этом примере демонстрируется использование списка констант в операторе выбора. Программа сообщает, сдал студент экзамен или не сдал. Если оценка одна из следующих: 3, 4, 5, то экзамен сдан; если 2, то не сдан.

CaseNof

```
3, 4, 5: WriteLn ('Экзамен сдан');  
2: WriteLn('Экзамен не сдан');  
Else WriteLn('Нет такой оценки')
```

End

Так же как условный оператор, оператор выбора может использоваться в неполной форме, т. е. без ветви *Else*.

Если применить условный оператор, то эта программа запишется так:

```
If (N=3) or (N=4) or (N=5) Then WriteLn ('Экзамен сдан')  
Else If N=2 Then WriteLn ('Экзамен не сдан')  
Else WriteLn ('Нет такой оценки');
```

В условии ветвления использовано сложное логическое выражение, содержащее операции логического сложения *or* (или).

Программирование ветвлений	
Условный оператор	Оператор выбора
<pre>If <логическое выражение> Then <оператор 1> Else <оператор 2></pre> <p>Если <логическое выражение> истинно, то выполняется <оператор 1>, иначе выполняется <оператор 2>.</p> <p><Оператор 1> и <оператор 2> — простые или составные операторы. В неполном ветвлении отсутствует ветвь Else</p>	<pre>Case <селектор> of <список констант 1>: <оператор 1>; ... <список констант N>: <оператор N>; Else <оператор></pre> <p>End</p> <p>Здесь <селектор> — выражение порядкового типа; константы имеют тот же тип. Выполняется только одна из ветвей выбора, которая содержит константу, совпадающую со значением селектора. Ветвь Else может отсутствовать</p>

Рисунок 20 – Система основных понятий

10.2 Пример поэтапной разработки программы решения задачи

Постановка задачи и формализация. Словом «задача» называют проблему, которая требует решения. Решение задачи начинается с ее постановки. На этапе постановки задачи в терминах предметной области (физики, экономики,

биологии и др.) определяются исходные данные и результаты, которые надо получить. Следующий этап — формализация задачи. Чаще всего процесс формализации означает перевод задачи на язык математики: формул, уравнений, неравенств, систем уравнений, систем неравенств и т. п. Решение полученной математической задачи требует знания математики, умения выполнять анализ математической задачи. Такой анализ необходим для того, чтобы построить правильный алгоритм решения, обладающий всеми свойствами алгоритма.

Анализ математической задачи. Пусть в результате формализации некоторой задачи было получено квадратное уравнение: $ax^2 + bx + c = 0$, где коэффициенты a, b, c являются исходными данными. Требуется решить это уравнение, т. е. найти его корни. Проведем анализ этой математической задачи.

Рассмотрим различные варианты значений исходных данных, которые приводят к разным результатам для решающего ее алгоритма. Ограничимся только поиском вещественных корней уравнения. Проанализируем все возможные варианты множества значений коэффициентов a, b, c :

Если $a = 0, b = 0, c = 0$,	то любое x — решение уравнения.
Если $a = 0, b = 0, c \neq 0$,	то уравнение решений не имеет.
Если $a = 0, b \neq 0$,	то это линейное уравнение, которое имеет одно решение: $x = -c/b$.
Если $a \neq 0$ и $d = b^2 - 4ac \geq 0$,	то уравнение имеет два вещественных корня: $x_1 = (-b + \sqrt{d}) / (2a)$, $x_2 = (-b - \sqrt{d}) / (2a)$.
Если $a \neq 0$ и $d < 0$,	то уравнение не имеет вещественных корней.

Рисунок 21 – Анализ вариантов

Построение алгоритма. Построим блок-схему алгоритма решения квадратного уравнения (рисунок 22), учитывающего все ситуации, описанные в анализе задачи. Здесь вместо слов «да» и «нет» использованы знаки «+» и «-». Построенный алгоритм, несомненно, удовлетворяет свойству универсальности по отношению к исходным данным. Запишем этот же алгоритм на учебном Алгоритмическом языке. Обратите внимание на смещения строк в тексте алгоритма — соблюдается принцип структуризации внешнего вида. Повторим

его: запись всякой вложенной структуры должна быть смещена на несколько позиций вправо относительно записи внешней структуры, а конструкции одного уровня вложенности записываются на одном вертикальном уровне.

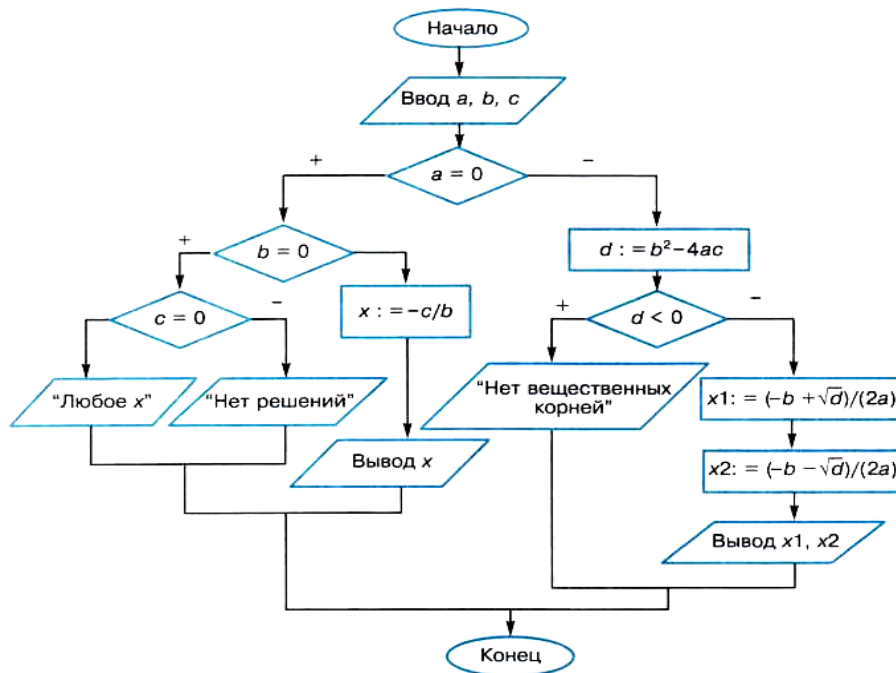


Рисунок 22 – Блок-схема алгоритма решения квадратного уравнения

```

алг корни квадратного уравнения
вещ  $a, b, c, d, x, x1, x2$ 
нач ввод  $a, b, c$ 
  если  $a=0$ 
  то
  если  $b=0$ 
  то
    если  $c=0$ 
    то вывод "Любое  $x$  – решение"
    иначе вывод "Нет решений"
    все
  иначе
     $x := -c/b$ 
    вывод  $x$ 
  все
  иначе
     $d := b^2 - 4ac$ 
    если  $d < 0$ 
    то вывод "Нет вещественных корней"
    иначе
       $x1 := (-b + \sqrt{d}) / (2a); x2 := (-b - \sqrt{d}) / (2a)$ 
      вывод " $x1 =$ ",  $x1$ , " $x2 =$ ",  $x2$ 
    все
  все
кон
  
```

Рисунок 23 – Алгоритм решения квадратного уравнения

Программирование. Алгоритмический язык (АЯ) — это язык описания алгоритмов с русскими служебными словами. После того как алгоритм записан на АЯ, составление программы на Паскале становится несложной задачей. Основное внимание следует уделять строгому соблюдению синтаксических правил языка. Правило смещения строк в тексте программы то же, что было сформулировано для АЯ. Соответствующие друг другу служебные слова *Begin* и *End* должны располагаться друг под другом. Текст программы решения квадратного уравнения:

```
Program Roots;
Var a, b, c, d, x, x1, x2: Real;
Begin
  WriteLn('Введите коэффициенты квадратного уравнения:');
  Write('a='); ReadLn(a);
  Write('b='); ReadLn(b);
  Write('c='); ReadLn(c);
  If a=0
  Then
    If b=0
    Then
      If c=0
      Then WriteLn('Любое x - решение')
      Else WriteLn('Нет решений')
    Else
      Begin
        x:=-c/b;
        WriteLn('x=', x)
      End
    Else
      Begin
        d:=b*b-4*a*c;
        If d<0
        Then WriteLn('Нет вещественных корней')
        Else
          Begin
            x1:=(-b+sqrt(d))/2/a;
            x2:=(-b-sqrt(d))/2/a;
            WriteLn('x1=', x1);
            WriteLn('x2=', x2)
          End
        End
      End
    End
  End.
```

Чем больше текст программы, тем больше вероятность совершения ошибок при ее записи и вводе в компьютер. Ошибки, нарушающие правила грамма-

тики языка, называются синтаксическими ошибками. Поиск и устранение синтаксических ошибок в программе называются отладкой. Отладить программу программисту помогает система программирования на данном языке, которая автоматически обнаруживает ошибки и сообщает о них программисту. Тестирование — это этап, на котором экспериментально доказываются правильность алгоритма, заключенного в программе, и работоспособность программы. Тест — это вариант решения задачи с заданными исходными данными, для которых известен результат. Предварительно должен быть составлен план тестирования. Для ветвящегося алгоритма должны быть протестированы все его ветви. В нашем примере пять ветвей, пять вариантов ответа. Значит, в плане тестирования должно быть не менее пяти вариантов теста. На рисунке 24 представлен план тестирования программы Roots и результаты проведенного тестирования.

№	Исходные значения	Верные результаты	Результаты тестирования
1	$a = 0, b = 0, c = 0$	Любое X — решение	Любое X — решение
2	$a = 0, b = 0, c = 1$	Нет решений	Нет решений
3	$a = 0, b = 2, c = -6$	$x = 3$	$x=3$
4	$a = 2, b = 1, c = -3$	$x_1 = 1, x_2 = -1,5$	$x1=1 \quad x2=-1,5$
5	$a = -1, b = -1, c = -2$	Нет вещественных корней	Нет вещественных корней

Рисунок 24 – План и результаты тестирования

Теперь, анализируя результаты тестирования, делаем вывод: правильность алгоритма и работоспособность программы доказаны. Если какой-то из вариантов теста не дает ожидаемого результата, то в программе есть ошибки. Например, пусть программист ошибочно записал следующие операторы присваивания для вычисления корней:

$$x1 := (-b + \text{sqrt}(d)) / 2 * a; \quad x2 := (-b - \text{sqrt}(d)) / 2 * a;$$

Результаты всех тестов, кроме 4-го, совпали с ожидаемыми, а в 4-м тесте получилось: $x_1 = 4, x_2 = -6$. После этого программист обратит внимание на вы-

ражения для вычисления корней и исправит ошибки: либо заменит знак умножения на знак деления, либо заключит в скобки выражение $2*a$.

Этапы решения задачи	
Постановка задачи	Определение исходных данных и искомых результатов (в терминах предметной области)
Формализация	Переход к задаче обработки некоторой знаковой системы, например к математической задаче
Анализ математической задачи	Определение всех вариантов множеств значений исходных данных. Определение для каждого варианта способа решения и вида выходных данных (результатов)
Построение алгоритма	Определение структуры алгоритма, последовательности команд. Представление на каком-либо языке описания алгоритмов (блок-схема, учебный Алгоритмический язык)
Составление программы	Запись и отладка программы на языке программирования. Строгое соблюдение правил синтаксиса языка
Тестирование	Экспериментальное доказательство правильности алгоритма и работоспособности программы. Тест — вариант решения задачи с заданными исходными данными, для которых известен результат. План тестирования строится так, чтобы наиболее полно проверить работу программы

Рисунок 25 – Этапы решения задачи

11 Лекция 11. Создание презентаций с использованием программы MS PowerPoint

11.1 Знакомство с рабочей областью MS PowerPoint

При запуске программа PowerPoint открывается в режиме, называемом обычным режимом, который позволяет создавать слайды и работать с ними.

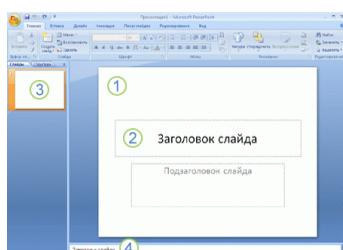



Рисунок 26 – Окно программы MS PowerPoint

В области *Слайд* можно работать непосредственно с отдельными слайдами. Пунктирные линии показывают местозаполнители, в которые можно ввести

текст или вставить изображения, диаграммы и другие объекты. Рамки (заполнители) — это поля с пунктирными границами, являющиеся частью макетов большинства слайдов. Эти поля содержат заголовки и основной текст либо такие объекты, как диаграммы, таблицы и рисунки. Объектом может быть так же формула или данные другого типа. Объекты, созданные в одном приложении (например, электронные таблицы), а затем связанные или внедренные в другом приложении, являются объектами OLE. Вкладка *Слайды* содержит эскизы всех полноразмерных слайдов, отображаемых в области *Слайд*. Эскизы — это миниатюрное представление рисунка. После добавления других слайдов для появления нужного слайда в области *Слайд* можно щелкнуть соответствующий эскиз на вкладке *Слайды*. Можно также перетаскивать эскизы, чтобы изменить порядок слайдов в презентации. Кроме того, вкладка *Слайды* позволяет добавлять и удалять слайды. Область *Заметки* позволяет ввести заметки о текущем слайде.

Область заметок — это область в обычном режиме для ввода заметок к слайду, распечатываемых в режиме страниц заметок или отображаемых на экране при сохранении презентации в формате веб-страницы. Можно раздать заметки аудитории или обращаться к ним во время показа презентации в режиме докладчика. По умолчанию Office PowerPoint 2007 использует для новых презентаций шаблон новой презентации, представленный на предыдущей иллюстрации. Новая презентация — это самый простой и самый общий из шаблонов Office PowerPoint 2007. Его удобно использовать в начале работы с PowerPoint, так как он прост и может быть адаптирован для различных типов презентаций. Для создания новой презентации на основе этого шаблона нажмите кнопку *Microsoft Office*, выберите команду *Создать*, в группе *Шаблоны* выберите *Пустые* и последние и дважды щелкните *Новая презентация* в группе *Пустые и последние*. После открытия шаблона новой презентации отображается только небольшая часть области *Заметки*. Чтобы увеличить видимую часть области *Заметки* и получить больше места для ввода данных, выполните следующие действия:

1. Наведите указатель на верхнюю границу области *Заметки*.
2. Когда указатель превратится в , перетащите границу вверх, чтобы увеличить область заметок докладчика, как показано на следующей иллюстрации.

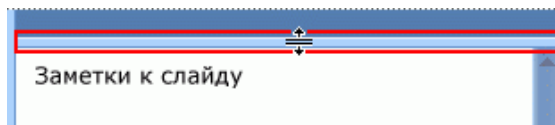






Рисунок 27 – Заметки к слайду


Обратите внимание, что размер слайда в области *Слайд* уменьшается автоматически в соответствии с доступным пространством. В верхней части экрана находятся три кнопки, которые могут оказаться полезны:

Кнопка *Отменить*  отменяет последнее изменение. (Изменение можно также отменить, нажав сочетание клавиш *CTRL+Z*). Кнопки *Вернуть*  и *Повторить* , возвращают или повторяют последнее изменение, в зависимости от выполненного действия. Чтобы увидеть подсказку о повторяемом или возвращаемом действии, поместите указатель на кнопку. Повторить или вернуть действие можно также, нажав сочетание клавиш *CTRL+Y*. Кнопка , открывает область *Справка PowerPoint*. Справку также можно открыть, нажав клавишу *F1*.

11.2 Присвоение имени и сохранение презентации

Как и при работе с любой другой программой, рекомендуется сразу же присвоить имя презентации, а затем во время работы часто выполнять ее сохранение. Нажмите кнопку *Microsoft Office*, укажите команду *Сохранение документа*, а затем выполните одно из следующих действий. Для презентации, которая может быть открыта только в Office PowerPoint 2007, выберите *Презентация PowerPoint*. Для презентации, которая может быть открыта либо в Office PowerPoint 2007, либо в более ранних версиях PowerPoint, выберите значение

Презентация PowerPoint 97-2003. Если выбран этот вариант, новые функции Office PowerPoint 2007 становятся недоступными.

В списке *Папка* диалогового окна *Сохранение документа* выберите папку или другое место, в котором нужно сохранить презентацию. В поле *Имя файла* введите имя презентации или примите имя файла, предлагаемое по умолчанию, и нажмите кнопку *Сохранить*. С этого момента можно нажимать сочетание клавиш *CTRL+S* или щелкать кнопку *Сохранить*  в верхней части экрана, чтобы в любой момент быстро сохранить презентацию.

11.3 Добавление, изменение порядка и удаление слайдов

Слайд, который автоматически появляется в презентации, содержит два местозаполнителя, один из которых отформатирован для заголовка, а второй — для подзаголовка. Порядок прототипов на слайде называется макетом. Макет предлагает о В Office PowerPoint 2007 также предусмотрены другие типы местозаполнителей, например местозаполнители для изображений и рисунков SmartArt. Чтобы одновременно с добавлением слайда в презентацию выбрать макет нового слайда, можно выполнить следующие действия. На вкладке *Слайды* щелкните непосредственно под единственным содержащимся на этой вкладке слайдом. В группе *Слайды* вкладки *Главная* щелкните стрелку рядом с кнопкой *Создать слайд*.

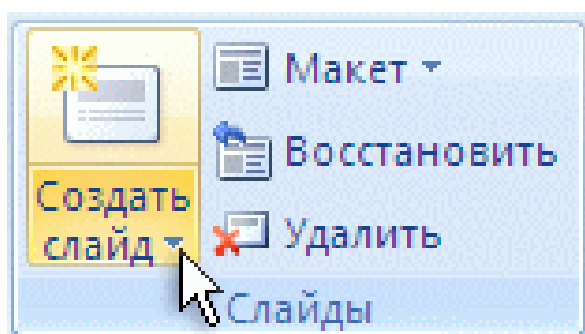


Рисунок 28 – Создание слайда

Появится коллекция, в которой отображаются эскизы различных доступных макетов слайдов.

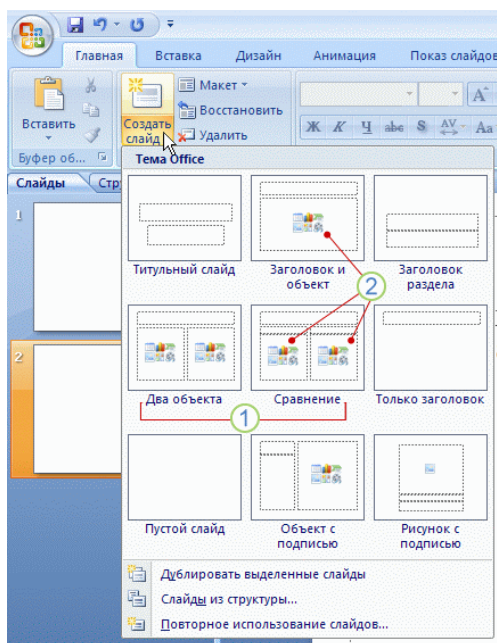


Рисунок 29 – Выбор макета

Имя определяет содержимое, для которого спроектирован каждый из макетов. Местозаполнители с цветными значками могут содержать текст, но в них также можно щелкнуть эти значки, чтобы автоматически вставить объекты, включая рисунки SmartArt и клип. Щелкните нужный макет для нового слайда. Новый слайд появляется и на вкладке *Слайды*, где он выделяется как текущий, и в области *Слайд*. Повторите эту процедуру для каждого добавляемого слайда. Если нужно, чтобы для нового слайда использовался тот же макет, что и для предыдущего слайда, просто нажмите кнопку *Создать слайд*, вместо того чтобы щелкать стрелку рядом с ней. Чтобы подсчитать нужное число слайдов, создайте план презентации, а затем разделите материал на отдельные слайды. Вероятно, понадобятся следующие слайды: основной титульный слайд, вводный слайд, содержащий основные темы или области презентации, один слайд для каждой темы или области, перечисленной на вводном слайде, итоговый слайд, повторяющий список основных тем или областей презентации. Если используется эта базовая структура, то при наличии трех основных представляемых тем или областей, можно планировать, что презентация будет содержать не менее шести слайдов: титульный слайд, вводный слайд, по одному слайду для каждой из трех основных тем или областей и итоговый слайд.

Если в любой из основных тем или областей нужно представить большой объем материала, может понадобиться создать группу слайдов для этого материала, используя ту же базовую структуру. Подумайте, сколько времени каждый из слайдов должен быть виден на экране в процессе показа презентации. Хорошей оценкой может служить от двух до пяти минут на слайд.

Применение к слайду нового макета. Чтобы изменить макет существующего слайда, выполните следующие действия. На вкладке *Слайды* щелкните слайд, к которому нужно применить новый макет. В группе *Слайды* вкладки *Главная* щелкните элемент *Макет*, а затем выберите нужный новый макет. При применении макета, не имеющего достаточного количества прототипов, соответствующих текущему содержанию слайда, автоматически создаются нужные прототипы для размещения этого содержания.

Копирование слайда. Если нужно создать два слайда, аналогичных по содержанию и макету, можно сэкономить усилия, создавая один слайд с форматированием и содержанием, общими для обоих слайдов, а затем создать копию этого слайда и добавить на каждый из этих слайдов окончательные индивидуальные детали. На вкладке *Слайды* щелкните копируемый слайд правой кнопкой мыши, а затем выберите команду *Копировать* в контекстном меню. Находясь на вкладке *Слайды*, щелкните правой кнопкой мыши место, в котором нужно добавить новую копию слайда, и выберите в контекстном меню команду *Вставить*.



Можно также вставить копию слайда из одной презентации в другую презентацию.

Изменение порядка слайдов. На вкладке *Слайды* щелкните слайд, который нужно переместить, а затем перетащите его в новое место. Чтобы выделить несколько слайдов, щелкните слайд, который нужно переместить, а затем нажмите и удерживайте клавишу *CTRL*, одновременно щелкая по очереди остальные слайды, которые нужно переместить.

Удаление слайда. На вкладке *Слайды* щелкните правой кнопкой мыши слайд, который необходимо удалить, а затем выберите в контекстном меню команду *Удалить слайд*.

11.4 Добавление и форматирование текста

Самым общим содержанием слайдов в презентации PowerPoint является текст — в заголовках, названиях и маркированных списках. Чтобы добавить текст на любой слайд, щелкните местозаполнитель, в который нужно добавить текст, а затем введите или вставьте нужный текст.

Форматирование маркированных списков. В одних местозаполнителях текст автоматически форматируется как маркированный список, а в других местозаполнителях — нет. В группе *Абзац* вкладки *Главная* выполните одно из следующих действий. Чтобы переключиться между маркированным и немаркированным текстом, выделите текст и нажмите кнопку *Маркеры* . Чтобы изменить стиль значков маркеров в маркированном списке, щелкните стрелку рядом с кнопкой *Маркеры* , а затем выберите нужный стиль маркера.

Эти изменения можно также выполнить с помощью мини-панели инструментов — удобной миниатюрной полупрозрачной панели инструментов, которая становится доступной после выделения текста. Чтобы четко увидеть мини-панель инструментов, наведите на нее указатель. Чтобы воспользоваться мини-панелью инструментов, щелкните любую из доступных команд. Вывести на экран мини-панель инструментов можно также, щелкнув невыделенный текст правой кнопкой мыши.

Изменение внешнего вида текста. Существует множество способов изменить внешний вид текста на слайде, от основных кнопок вкладки *Главная*, предназначенных для форматирования характеристик шрифта, стиля, размера, цвета и абзаца, до дополнительных параметров, таких как анимация или преобразование в рисунки SmartArt.

Добавление заметок докладчика. Слишком большое количество текста делает слайд запутанным и непонятным для аудитории. Однако, если убрать с экрана часть данных, сделав их невидимыми для аудитории, как можно их отследить? Решением этой проблемы являются заметки докладчика, которые можно ввести в области Заметки для каждого слайда. Заметки докладчика помогают в процессе презентации избавить экран от избыточного содержания, одновременно позволяя отслеживать все данные, нужные во время презентации. Можно также в любой момент легко вырезать излишне подробный текст из области *Слайд*, а затем вставить этот текст прямо в область *Заметки*, чтобы можно было пользоваться им для справки. Заметки докладчика можно напечатать и заглядывать в них во время презентации. Либо, если презентация Office PowerPoint 2007 запускается с одного монитора (например, на трибуне), а аудитория видит ее на другом мониторе, то для вывода заметок во время презентации только на монитор докладчика можно использовать режим докладчика.

Придание презентации нужного внешнего вида. Необходимо продумать общий внешний вид презентации. Какой визуальный тон нужно использовать? Какой вид презентации сделает ее понятной и привлекательной для аудитории? Office PowerPoint 2007 предоставляет множество. Тема представляет собой набор элементов оформления, придающий особый, единообразный внешний вид всем документам Office, используя конкретные сочетания цветов. Office PowerPoint 2007 автоматически применяет к презентациям, созданным с помощью шаблона новой презентации, тему Office, но внешний вид презентации можно легко изменить в любой момент, применив другую тему.

В группе *Темы* вкладки *Оформление слайда* щелкните нужную тему документа. Для предварительного просмотра внешнего вида текущего слайда после применения конкретной темы наведите указатель на эскиз этой темы. Чтобы увидеть эскизы дополнительных тем, щелкните стрелки рядом со строкой эскизов.



Рисунок 30 – Применение темы


Если не указано иное, Office PowerPoint 2007 применяет темы ко всей презентации. Чтобы изменить внешний вид только выбранных слайдов, на вкладке Слайды нажмите и удерживайте клавишу *CTRL*, одновременно щелкая каждый слайд, который нужно изменить. Выбрав все слайды, щелкните правой кнопкой мыши тему, которую нужно применить к этим слайдам, и выберите в контекстном меню команду *Применить к выделенным слайдам*. Если позднее понадобится использовать другую тему, щелкните эту тему, чтобы применить её.

Добавление клипа, рисунков SmartArt и других объектов. Создаваемая презентация должна быть максимально эффективной визуально — и часто серия слайдов, содержащая только маркированные списки, не является самым динамичным вариантом. Недостаток визуального разнообразия может привести к потере внимания аудитории. Кроме того, для многих видов данных абзац или маркированный список не является оптимальным представлением. Office PowerPoint 2007 позволяет добавлять множество видов аудио и видеоданных, включая таблицы, рисунки SmartArt, клип, фигуры, диаграммы, музыку, фильмы, звуки и анимации. Можно добавить гиперссылки, а также привлекающие глаз переходы между слайдами.

Добавление клипа. Щелкните прототип, в который необходимо добавить клип. Если прототип не выделен или если выделен прототип, в который нельзя вставить изображение, клип вставляется в центр слайда. На вкладке *Вставка* в группе *Иллюстрации* нажмите кнопку *Клип*. Откроется область задач *Клип*. В области задач *Клип* найдите и щелкните нужный клип. Теперь клип можно переместить, изменить ее размер, повернуть, добавить к ней текст и выполнить иные изменения. Чтобы найти дополнительный клип на веб-узле Microsoft

Office Online, щелкните ссылку *Клип* на узле Office Online в нижней части области задач *Клип*.

Преобразование текста слайда в рисунок SmartArt. Рисунок SmartArt — это визуальное представление сведений, которое можно полностью настроить. Преобразование текста в рисунок SmartArt — это быстрый способ преобразовать существующие слайды в профессионально оформленные иллюстрации. Например, можно одним щелчком превратить слайд повестки дня в рисунок SmartArt.

Чтобы наглядно выразить свои мысли и идеи, можно воспользоваться встроенными макетами. Чтобы преобразовать существующий текст в рисунок SmartArt, выполните следующие действия. Щелкните местозаполнитель, содержащий текст, который нужно преобразовать. В группе *Абзац* вкладки *Главная* нажмите кнопку *Преобразовать в рисунок SmartArt* . Чтобы увидеть, как будет выглядеть рисунок SmartArt с нужным текстом, наведите в коллекции указатель на эскиз этого рисунка SmartArt. Коллекция содержит макеты рисунков SmartArt, которые лучше всего подходят для маркированных списков. Для просмотра полного набора макетов нажмите кнопку *Дополнительные рисунки SmartArt*. Найдя нужный рисунок SmartArt, щелкните его, чтобы применить к своему тексту. Теперь рисунок SmartArt можно переместить, изменить его размер, повернуть, добавить к нему текст, применить к нему другой экспресс-стиль и выполнить иные изменения. Быстрые стили — это набор параметров форматирования, облегчающий форматирование документов и объектов. Хотя рисунок SmartArt легче всего создать для существующего текста, можно пойти другим путем и сначала вставить нужный рисунок SmartArt, а затем добавить к нему текст. Щелкните местозаполнитель, в который нужно добавить рисунок SmartArt. Если местозаполнитель не выделен или если выделен местозаполнитель, в который нельзя вставить изображение, рисунок SmartArt вставляется в центр слайда. На вкладке *Вставка* в группе *Иллюстрации* нажмите кнопку *SmartArt*. В крайней левой области диалогового окна *Выбор рисунка SmartArt* щелкните нужный тип рисунка SmartArt. В центральной области найдите и

щелкните нужный макет, а затем нажмите кнопку *ОК*. Для предварительного просмотра любого макета щелкните этот макет. Предварительный просмотр появится в крайней правой области.

Добавление смены слайдов. Смены слайдов представляют собой анимационные эффекты, возникающие при переходе от одного слайда к следующему. Office PowerPoint 2007 предоставляет множество типов смены слайдов, включая стандартные эффекты затухания, растворения, обрезания и стирания, а также более необычные переходы, например колеса и шахматные доски. В группе *Переход к этому слайду* вкладки *Анимации* выберите нужный вариант перехода. Для предварительного просмотра внешнего вида текущего слайда с использованием конкретного варианта перехода наведите указатель на эскиз этого перехода. Чтобы просмотреть эскизы других переходов, щелкните стрелки рядом со строкой эскизов.

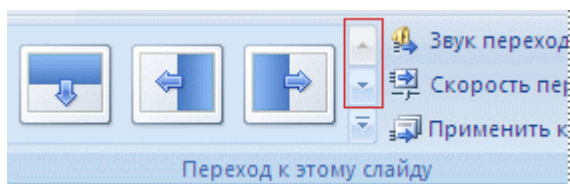


Рисунок 31 – Эскизы переходов

Если позднее понадобится использовать другой вариант перехода слайдов, щелкните этот переход, чтобы применить его. Можно выбрать другие варианты в группе *Переход к этому слайду*, чтобы управлять скоростью перехода, добавить звук и применить этот же вариант перехода ко всем слайдам презентации.

Добавление гиперссылок. Для перехода с одного слайда на другой, к ресурсу в локальной сети или в Интернете либо даже к другому файлу или программе можно воспользоваться гиперссылками. Выделите текст, который нужно щелкнуть для активации гиперссылки. Либо можно выделить объект (например, клип или рисунок SmartArt). В группе *Ссылки* вкладки *Вставка* щелкните элемент *Гиперссылка*. В диалоговом окне *Вставка гиперссылки* нажмите соответствующую кнопку в поле *Мои адреса*, чтобы задать назначение ссылки

(то есть место, на которое указывает ссылка). Например, чтобы перейти на другой слайд презентации, нажмите кнопку *Место в документе*. Найдите и щелкните место назначения, внесите нужные изменения в поля *Отображаемый текст* и *Адрес*, а затем нажмите кнопку *ОК*.

Проверка орфографии в презентации. Несмотря на то, что в программе Office PowerPoint 2007 проверка орфографии включена автоматически, по окончании работы с презентацией рекомендуется еще раз проверить орфографию. Нажмите сочетание клавиш *CTRL+HOME*, чтобы переместиться к началу презентации. В группе *Правописание* вкладки *Рецензирование* выберите *Орфография*. Если Office PowerPoint 2007 обнаруживает орфографические ошибки, появляется диалоговое окно и выделяется первое слово с ошибкой, обнаруженное средством проверки орфографии. Автор определяет, как нужно исправить найденную программой ошибку. После исправления неправильного слова программа находит следующее слово с ошибкой.

Просмотр презентации в виде показа слайдов. Для просмотра презентации на экране компьютера в том виде, в каком она будет представлена аудитории, выполните следующие действия. В группе *Начать показ слайдов* вкладки *Показ слайдов* выполните одно из следующих действий. Для запуска презентации с первого слайда выберите *С начала*. Чтобы начать показ со слайда, в настоящий момент находящегося в области *Слайд*, выберите *С текущего слайда*. Презентация открывается в режиме показа слайдов. Щелкните мышью, чтобы перейти к следующему слайду. Чтобы вернуться в обычный режим, в любой момент можно нажать клавишу *ESC*. Конечно, просмотр возможен в любой момент создания презентации.

Подготовка к проведению презентации. Теперь, когда презентация готова, дальнейшие действия зависят от того, кто будет показывать презентацию, а также от того, где она будет показываться и какое оборудование будет использоваться. Будет ли презентация выполняться на том же компьютере, на котором она была создана, или на другом компьютере, а также будет ли она доставляться на компакт-диске или с местоположения в локальной сети? Будет ли презен-

тация показываться докладчиком, и в этом случае будет ли ее показывать автор или другой докладчик либо она будет демонстрироваться автоматически? В зависимости от ответов на эти вопросы может понадобиться выполнить следующие процессы в указанном порядке, в другом порядке или (в некоторых случаях) выполнение этих процессов не потребуется.

Настройка времени презентации. Важно предусмотреть достаточно времени, чтобы любой докладчик, использующий презентацию, мог наверняка уложиться в график (включая при необходимости время в конце презентации для вопросов аудитории) и правильно распределить время. Рекомендуется настраивать график презентации в помещении, где она будет демонстрироваться, используя оборудование, на котором она будет демонстрироваться, и показывая ее, по крайней мере, одному или двум слушателям. Это позволит докладчику познакомиться с местом для показа презентации и потренироваться перед живой аудиторией — обратная связь с аудиторией может быть полезна, чтобы определить элементы, которые нужно исправить перед реальным показом.

Печать материалов для выдачи и заметок докладчика. Презентацию можно напечатать либо как заметки докладчика, либо материалы для выдачи. Заметки докладчика содержат один слайд в верхней части каждой распечатанной страницы, а также содержание области *Заметки* в нижней части страницы. Они могут использоваться докладчиком во время презентации в качестве сценария или структуры. Они также могут быть розданы аудитории, чтобы каждый слушатель получил все сведения, содержащиеся в презентации. Материалы для выдачи содержат один, два, три, четыре, шесть или девять слайдов на распечатанной странице и предназначены для случаев, когда нежелательно показывать аудитории содержание области *Заметки*. Материалы для выдачи с тремя слайдами на страницу содержат расчерченную в линейку область, где слушатели могут записывать заметки.

Функция упаковки для записи на компакт-диск или для веб-распространения. При использовании функции упаковки для записи на компакт-диск для копирования законченной презентации PowerPoint на компакт-

диск, в местоположение в локальной сети или на жесткий диск компьютера будут скопированы также программа Microsoft Office PowerPoint Viewer 2007 и все файлы, связанные с презентацией (например, фильмы и звуки). Тем самым будут включены все элементы презентации, и ее смогут просмотреть люди, на компьютерах которых не установлена программа Office PowerPoint 2007.

12 Лекция 12. Содержательные и дизайн-эргономические требования к презентациям

12.1 Содержательные требования к презентации

Презентация должна содержать следующие слайды:

- титульную страницу (номер и название темы лекции, Ф.И.О. лектора);
- план лекции (организовать на основе гиперссылок);
- содержательные слайды;
- заключение;
- список рекомендуемых источников и/или аннотированных интернет-ресурсов.

Компактность. Содержание каждого слайда должно быть представлено в виде тезисов, кратко освещающих основные положения и ключевые моменты лекции и не дублирующих устную речь лектора.

Чёткая структуризация представленного материала (следует разбивать большой текст на абзацы, использовать короткие словосочетания и предложения, минимизировать количество предлогов, наречий, исключить вводные слова). Желательно использовать нумерацию слайдов. Если в качестве рекомендуемых источников указываются ссылки на интернет-ресурсы, то целесообразно приводить аннотации, содержащие краткую характеристику ресурса. Весь материал должен быть тщательно проверен на отсутствие орфографических, грамматических и стилистических ошибок. В соответствии с правилами пунктуации точки в конце названия темы и в заголовках недопустимы.

12.2 Дизайн-эргономические требования к презентации

Требования размещения материала. Единый стиль размещения материала (весь дизайн презентации следует выстраивать по общей схеме зонирования: заголовков и основной текст). Расположение заголовков и основного текста на каждом слайде должно быть одинаковым. Желательно, чтобы заголовки присутствовали на каждом слайде. Оптимальность и удобство распределения информации на экране (материал для усвоения зрителями должен занимать не более 2/3 пространства всего слайда). Информация (текст, таблицы, схемы, диаграммы, рисунки и т.п.) не должна скапливаться на одной стороне экрана, логические группы информации должны продуманно размещаться на слайде. Пояснения к иллюстрациям (рисункам, схемам, диаграммам и т.п.) должны располагаться под ними и как можно ближе к ним. Пояснения к таблицам должны располагаться над ними. Формулы желательно размещать в центре экрана (для наглядности их можно выделять заливкой). Основной текст не должен выходить за рамки слайда и перекрывать заголовков. Текст в диаграммах, рисунках, схемах, таблицах не должен выходить за пределы их области. Информация, представленная в виде таблицы, должна быть наглядной и читабельной. Текст не должен выходить за границы таблицы. Следует использовать в ячейках таблицах орфографически правильные переносы. Таблицы желательно создавать средствами самого программного продукта (например, Microsoft PowerPoint). Большие таблицы по возможности следует разбивать на несколько более мелких.

Требования цветового оформления. Необходимо соблюдать единый стиль (шаблон) оформления всех слайдов в пределах всей презентации. Стиль и дизайн презентации должен определяться предметной направленностью её содержания. Необходимо учитывать постоянство используемых цветов (при дизайне презентации следует выбрать 2-3 цвета, которые можно разнообразить их же оттенками). Использовать общую цветовую гамму (все заголовки одного цвета, весь основной текст одного цвета, все списки одного цвета).

Необходимо соблюдать единство и гармоничность стилистического оформления для всех элементов в пределах всей презентации (текста, ссылок, таблиц, схем, диаграмм). Необходимо учитывать сочетание цветового оформления объектов (таблиц, схем, диаграмм и т.п.) с общим цветовым оформлением презентации. Необходимо соблюдать контрастность фона и текста по отношению друг к другу (при подборе цветовой гаммы учитывается цветопередача при проецировании на экран мультимедийной установки). Цвет активных и неактивных гиперссылок должен сочетаться со стилевым решением всего курса лекций

Требования шрифтового оформления. Единый размер шрифта для всех заголовков на каждом слайде, при этом необходимо учитывать читабельность шрифта (например, для Arial не менее 24 пт). Единый размер шрифта для основной информации на каждом слайде, в том числе и для таблиц, при этом необходимо учитывать читабельность шрифта (например, для Arial не менее 18 пт). Нельзя смешивать различные типы шрифтов в одной презентации. Шрифты без засечек (Arial, Verdana, Tahoma) более читабельны в электронных текстах. Читабельность шрифта касается не только основного текста, но и таблиц, схем, рисунков и т.п. Для основного текста желательно использовать только обычное начертание шрифта (не использовать полужирное начертание и курсив для всего текста). При оформлении текста и списков абзацные отступы на каждом слайде должны быть одинаковыми. Основной текст и списки желательно выравнивать по левому краю. При оформлении основного текста следует использовать только строчные буквы (исключить написание прописными буквами всего текста).

Требования по выделению материала. Для выделения ключевой информации необходимо использовать полужирное начертание шрифта и/или другой цвет. С целью выделения ключевой информации не допускается подчеркивание текста, так как оно ассоциируется с гиперссылками. Выделить заголовки можно полужирным начертанием, цветом или большим размером шрифта. Термины, определения, формулы, рисунки, выводы можно помещать в рамочку или вы-

делять другим цветом. Для структурирования и наглядности текста желательно использовать нумерованные и маркированные списки, но не следует забывать про единообразие при их оформлении. Гиперссылки должны быть четко обозначены подчеркиванием и/или цветом (желательно, чтобы гиперссылки содержали подсказки, поясняющие, на какую веб-страницу, документ или внешний объект можно осуществить переход).

Требования по использованию интерактивных и мультимедийных объектов. Удобство навигации по презентации (кнопки перехода на план и между слайдами, если они присутствуют, должны располагаться в одном и том же месте). При использовании звукового сопровождения (голоса диктора, музыкальных вставок и т.п.) необходимо учитывать его уместность и качество. Звуковое сопровождение следует включать в трудных для понимания местах материала презентации. При использовании анимационных объектов (роликов, анимированных изображений) необходимо учитывать их уместность и качество. Уместность использования анимационных эффектов текста, графических изображений и при смене слайдов (эффекты должны использоваться только в тех случаях, когда необходимо сконцентрировать внимание обучающихся на каких-либо формулах, определениях, схемах или при порционной подаче материала лекции). Недопустимо использование избыточных эффектов анимации, приводящих к быстрому утомлению органа зрения (например, «шашки», «ромб», «вращение», «пишущая машинка»).

Программно-технологические требования. Качество графических изображений (фото, рисунков, схем, диаграмм, иллюстраций и т.п.) должно быть высоким, недопустимы нечеткие материалы. Необходимо учитывать обоснованность использования ресурсов персонального компьютера (объем внедренных объектов не должен быть ресурсоемким).

12.3 Цветовой круг

Гармоничные комбинации цвета называют цветовыми схемами. В оформлении используются оттенки одного цвета (составляющие одного и того же сег-

мента). Такая композиция способствует поддержанию атмосферы спокойствия и отдыха, производит впечатление аккуратности и изящества. Гармоничные цвета расположены на цветовом круге на одинаковом расстоянии друг от друга. В этом случае выбирают цвета сегментов, расположенных в круге по соседству друг с другом. Обычно один цвет используется как доминирующий, в то время как другие цвета являются вспомогательными. Желательно не пользоваться цветами, расположенными слишком близко друг к другу на цветовом круге.

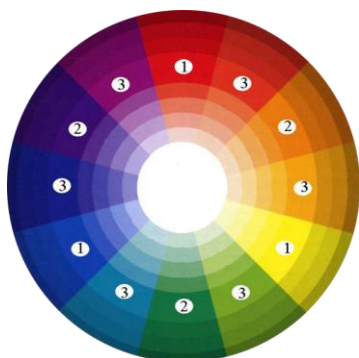


Рисунок 32– Цветовой круг

1 – основные (первичные) цвета; 2 – вторичные; 3 – третичные цвета

Нюансовыми цветами называются цвета близкие по оттенку, но различающиеся по насыщенности и яркости. Опираясь на цветовой круг, можно сформулировать три принципа подбора гармоничных сочетаний цветов, т.е. три типа цветовых комбинаций: тональная (нюансная) цветовая комбинация, гармоничная цветовая комбинация, контрастная цветовая комбинация

13 Лекция 13. Начало работы с MS Access. Создание базы данных

13.1 Объекты Microsoft Access

Microsoft Access является реляционной базой данных. Microsoft Access называет объектами все, что может иметь имя (в смысле Access). В базе данных Access основными объектами являются таблицы, запросы, формы, отчеты, макросы и модули. В других СУБД, как правило, термин база данных обычно относится только к файлам, в которых хранятся данные. В Microsoft Access база

данных включает в себя все объекты, связанные с хранимыми данными, в том числе и те, которые определяются для автоматизации работы с ними. Ниже приведен список основных объектов базы данных Access.

1) Таблица. Объект, который определяется и используется для хранения данных. Каждая таблица включает информацию об объекте определенного типа, например о клиентах. Таблица содержит поля (столбцы), в которых хранятся различного рода данные, например фамилия или адрес клиента, и записи (которые называются также строками). В записи собрана вся информация о некотором объекте (человеке, образце продукции и т.п.). Для каждой таблицы можно определить первичный ключ (одно или несколько полей, содержащих уникальные для каждой записи значения) и один или несколько индексов, помогающих ускорить доступ к данным.

2) Запрос. Объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц. Для создания запроса можно использовать запрос по образцу или инструкции SQL (структурированный язык запросов). Можно создать запросы на выборку, обновление, удаление или добавление данных. С помощью запросов можно также создавать новые таблицы, используя данные из одной или нескольких существующих таблиц.

3) Форма. Объект, предназначенный в основном для ввода данных, отображения их на экране или управления работой приложения. Формы используются для того, чтобы реализовать требования пользователя к представлению данных из запросов или таблиц. Формы можно также распечатать. С помощью формы можно в ответ на некоторое событие, например изменение значения определенных данных, запустить макрос.

4) Отчет. Объект, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения.

5) Макрос. Объект, представляющий собой структурированное описание одного или нескольких действий, которые должен выполнить Access в ответ на определенное событие. Например, можно определить макрос, который в ответ

на выбор некоторого элемента в основной форме открывает другую форму. С помощью другого макроса можно осуществлять проверку значения некоторого поля при изменении его содержимого. В макрос можно включить дополнительные условия для выполнения или невыполнения тех или иных указанных в нем действий. Из одного макроса можно также запустить другой макрос или процедуру VBA.

6) Модуль. Объект, содержащий программы, написанные на языке Visual Basic для приложений. Модули могут быть независимыми объектами, содержащими функции, вызываемые из любого места приложения, но они могут быть и непосредственно «привязаны» к отдельным формам или отчетам для реакции на происходящие в них изменения.

7) Страницы доступа. Страницы - служат для обеспечения доступа к данным, содержащимся в базе, удалённой от потребителя (например, через Интернет).

Прикладная программа Microsoft Access 2007 предназначена для создания базы данных. В Access 2007 используются логически связанные таблицы. СУБД Microsoft Access 2007 обеспечивает удобное и надежное управления данных, которые хранятся в таблицах. Суть создания базы данных в Microsoft Access 2007 состоит в том, что сначала надо создать структуру базы данных (создать структуру таблиц и установить между ними связи), а затем необходимо заполнить таблицы новой базы данных. Кроме того, желательно создать формы для ввода данных в таблицы, запросы для поиска информации в БД и отчеты для вывода из БД необходимой информации в удобном виде на экран, в печать или файл. БД можно создавать как вручную (новую пустую базу данных), так и на основе шаблонов. С их помощью можно быстро создать базу данных близкую к той, что требуется, а затем ее можно изменить и наполнить своими данными. При запуске прикладной программы Access 2007 на экране будет отображаться новая начальная страница «Приступая к работе с Microsoft Access 2007», представленная на рисунке 33.

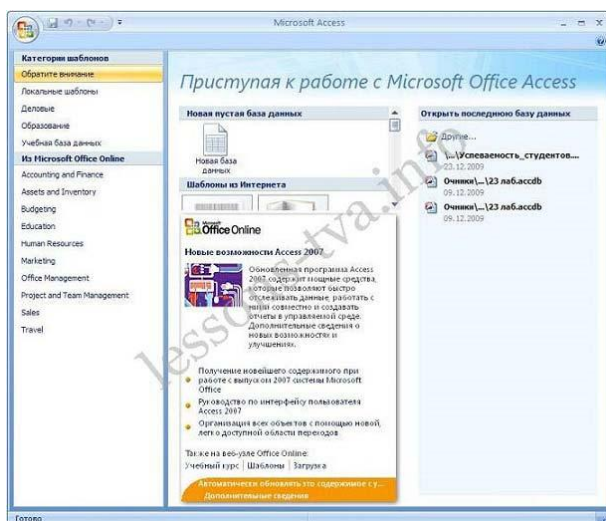


Рисунок 33 – Новая начальная страница

При создании базы данных на основе шаблонов можно выбрать требуемый template на странице «Приступая к работе с Microsoft Access 2007» из «Категории шаблонов» или шаблоны «Из Microsoft Office Online». При выборе, например шаблона *Факультет* в категории *Локальные шаблоны*, справа в окне приложения появится описание БД. Ниже, в текстовом поле указано имя файла *Факультет.accdb* и пиктограмма папки, с помощью которой можно сохранить файл в требуемую директорию (по умолчанию указана папка *Мои документы*).

13.2 Начало работы с Access 2007

Для создания новой пустой базы данных необходимо щелкнуть на пиктограмме «Пустая база данных» в разделе «Новая пустая база данных», справа в окне приложения появится текстовое поле с именем файла *База данных1.accdb* и пиктограмма папки для сохранения файла в требуемый каталог. По умолчанию указано имя файла, имя каталога и диска, в котором будет сохранен файл. Имя файла, директории и диска для хранения файла можно изменить. Необходимо отметить, что Access 2007 для новых баз данных по умолчанию использует формат файла Access 2007 (.accdb), но файл новой базы данных можно сохранить и в формате Access 2002-2003 (.mdb) или формате Access 2000 (.mdb). Скриншот окна сохранения файла БД представлено на рисунке 34.

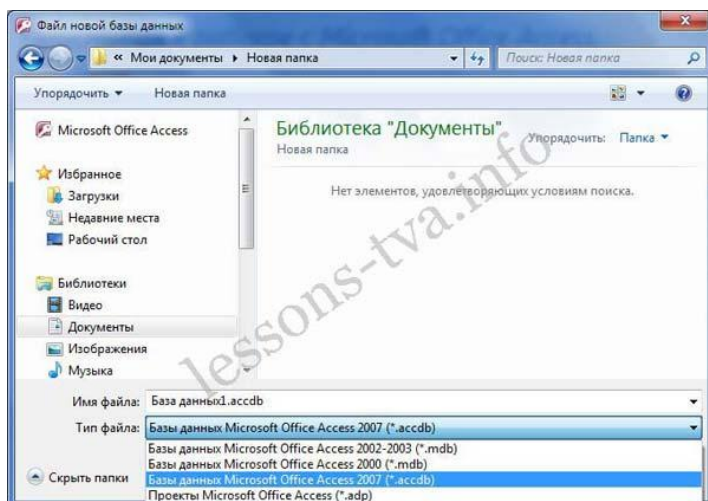


Рисунок 34 – Окно сохранения файла БД Access

После выбора директории, диска, имени файла и формата БД щелкнуть на кнопке *Создать*. Файл базы данных с выбранным именем будет сохранен в указанную папку, и откроется окно Microsoft Access 2007. В этом окне приложения отображается окно БД с назначенным именем на вкладке *Режим таблицы* и новая пустая таблица с именем Таблица 1 в режиме таблица, представленные на рисунке 35. Далее необходимо перейти в режим *Конструктор* и создать структуру первой таблицы базы данных.



Рисунок 35 – Окно таблицы в режиме таблица БД Access

Уже сохраненную базу данных в формате файла Access 2007 можно также сохранить в других форматах, щелкнув на кнопке *Office* в окне БД и выбрав команду *Сохранить как*. В появившемся окне диалога щелкнуть на требуемый формат. Рассмотрим окно приложения Microsoft Access 2007. В верхней части окна расположены: кнопка *Office*, панель быстрого доступа с пиктограммами (сохранить, отменить), Строка заголовка и кнопки изменения размеров окна. Ниже расположена *Лента*, которая состоит из вкладок (*Главная*, *Создание*,

Внешние данные, Работа с базами данных и другие вкладки, которые появляются в зависимости от режима работы). В приложении Access 2007 применяются контекстные инструменты, которые появляются при работе с определенным объектом. Так, например, при работе с таблицей появляются контекстные инструменты для объекта Таблица под названием «Работа с таблицами», которые имеют две вкладки: *Режим таблицы* и *Конструктор*. Лента заменяет используемые в Access 2003 меню и панели инструментов, кроме меню *Office* и панели быстрого доступа. На *Ленте* отображаются определенные наборы команд в зависимости от того, какая вкладка выбрана. Группы команд на выбранной вкладке сгруппированы по выполняемым действиям. Так на вкладке *Режим таблиц* отображаются группы команд *Представление, Поля и столбцы, Форматирование и тип данных, Связи*. Имена перечисленных групп команд отображаются внизу *Ленты*. Под лентой расположены слева *Область переходов*, а справа *Окно редактирования*, в котором отображается редактируемый объект. В области переходов находятся все объекты Access (таблицы, формы, запросы, отчеты и т.д.). В списке *Все объекты Access* можно выбрать требуемый объект. При двойном щелчке на имени объекта в области переходов этот объект будет отображаться в окне редактирования. Внизу окна Access расположена *Строка состояния* и *Кнопки режимов просмотра*.

13.3 Пример создания базы данных (структуры таблиц) в Access 2007

Рассмотрим этапы создания БД «Деканат» с помощью СУБД Access 2007. Сначала составляем модель «сущность – связь» для базы данных «Деканат». После создания модели запускаем приложение Access 2007. Открывается окно приложение Access 2007 на странице *Приступая к работе с Microsoft Access 2007*. В разделе *Новая пустая база данных* щелкаем на пиктограмме *Новая база данных*. В правой части окна появится информация об имени файла и указана директория для его хранения. По умолчанию имя файла *База данных1.accdb*. Изменить имя файла и путь к директории для хранения файла БД можно в окне «Файл новой базы данных» щелкнув на пиктограмме «Поиск

расположения для размещения базы данных». Установив имя файла *Деканат_2018.accdb* и требуемое имя директории в окне «Файл новой базы данных», надо щелкнуть на кнопке *OK*, окно закроется. Далее необходимо щелкнуть на кнопке *Создать*, чтобы создать пустую базу данных. При создании новой пустой базы данных окно приложения Access 2007 открывается на контекстной вкладке «Режим таблицы». В окне отображается новая пустая таблица с именем Таблица 1 в режиме таблица, представленная на рисунке 36.

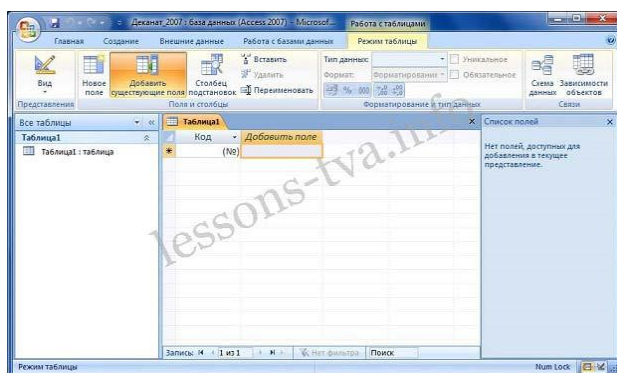


Рисунок 36 – Режим Таблица

Далее необходимо перейти в режим *Конструктор* и создать структуру первой таблицы базы данных. Для этого необходимо щелкнуть на пиктограмме *Вид* и выбрать режим *Конструктор*.

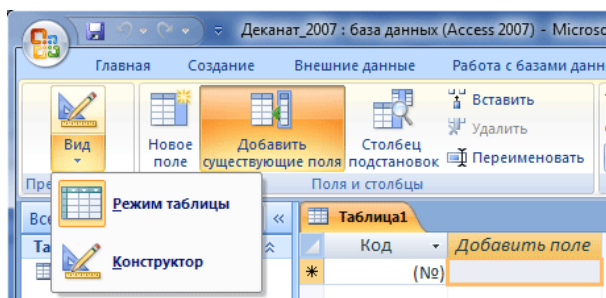


Рисунок 37– Режим Конструктор

Откроется окно *Сохранение*, в котором надо указать название группы и нажать кнопку *OK*.

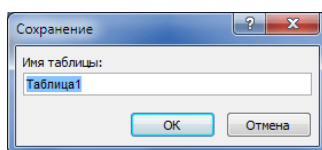


Рисунок 38 – Сохранение таблицы

Откроется таблица *Группы студентов* в режиме *Конструктор*.

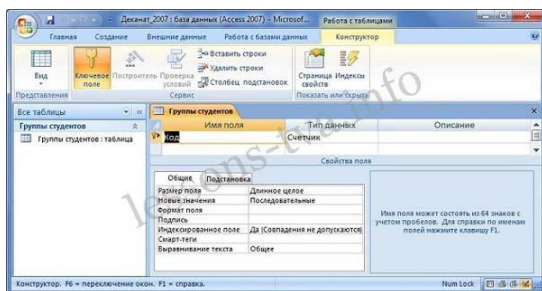


Рисунок 39– Таблица в режиме Конструктор

Необходимо создать структуру таблицы Группы студентов. В первую строку колонки «Имя поля» ввести код группы студентов (КодГруппы) и нажать клавишу *Enter*. Курсор переместится в колонку Тип данных. Access по умолчанию назначает тип данных — *Счетчик*. Нажать клавишу *Enter*, при этом курсор переместится в колонку *Описание*, при необходимости ввести описание данных. Первой строке таблицы (поле КодГруппы) Access по умолчанию назначает поле первичного ключа. Для первичного ключа в свойствах поля устанавливается значение *Индексированного поля: Да (Совпадения не допускаются)*. Далее заполняем вторую строку (второе поле таблицы), *Имя поля Название, Тип данных текстовый. Третья строка: Имя поля Курс, Тип данных числовой и четвертая строка Имя поля Семестр, Тип данных числовой*. При этом для имени поля «Название» в разделе *свойства поля* необходимо установить размер поля 6.

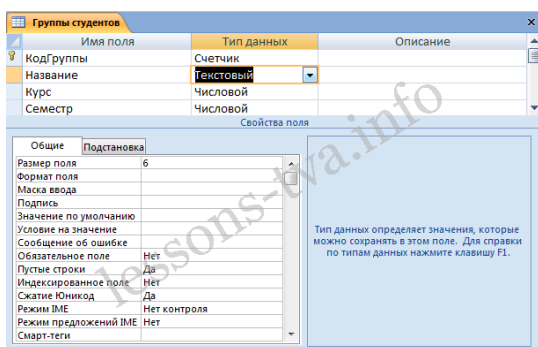
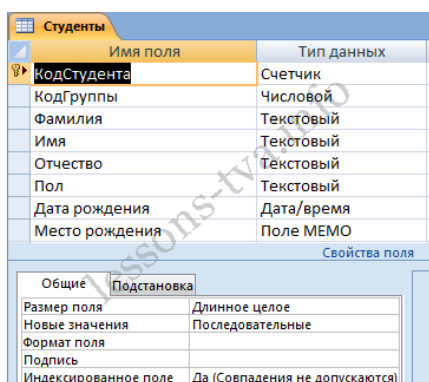


Рисунок 40 – Структура таблицы Группы студентов

Затем создать структуры остальных трех таблиц в соответствии с характеристиками таблиц-объектов *Студенты, Дисциплины, Успеваемость*. Обязательно соблюсти указанную последовательность создания структуры таблиц. В структуре таблицы «Студенты» для поля КодГруппы (вторичный ключ) уста-


новите значение *Индексированного поля*: *Да (Совпадения допускаются)* и тип данных – мастер подстановок. В структуре таблицы «Успеваемость» для поля КодСтуденты (вторичный ключ) и поля КодДисциплины (вторичный ключ) установите значение *Индексированного поля*: *Да (Совпадения допускаются)* и тип данных – мастер подстановок. Структуры остальных таблиц: *Студенты*, *Дисциплины*, *Успеваемость* следует создать в соответствии с рисунками.



Имя поля	Тип данных
КодСтудента	Счетчик
КодГруппы	Числовой
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Пол	Текстовый
Дата рождения	Дата/время
Место рождения	Поле МЕМО

Свойства поля	
Общие	Подстановка
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допускаются)

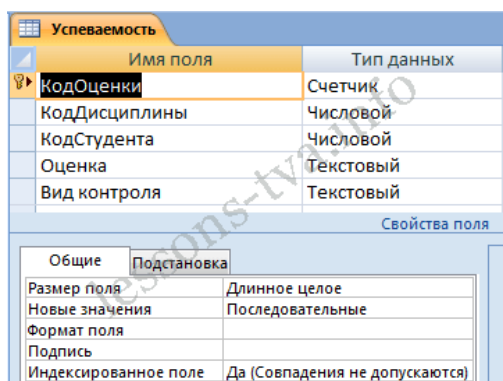
Рисунок 41 – Структура таблицы Студенты



Имя поля	Тип данных
КодДисциплины	Счетчик
Название	Текстовый
Кол часов	Числовой

Свойства поля	
Общие	Подстановка
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допускаются)

Рисунок 42 – Структура таблицы Дисциплины



Имя поля	Тип данных
КодОценки	Счетчик
КодДисциплины	Числовой
КодСтудента	Числовой
Оценка	Текстовый
Вид контроля	Текстовый

Свойства поля	
Общие	Подстановка
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допускаются)

Рисунок 43 – Структура таблицы Успеваемость

После этого необходимо установить логические связи между всеми таблицами.

данных

14.1 Установка логических связей в БД Access 2007

Логические связи устанавливаются между одноименными полями таблиц базы данных Access 2007. Связь данных в одной таблице с данными в других таблицах осуществляется через уникальные идентификаторы (ключи) или ключевые поля. В нашем случае мы должны установить логические связи между таблицами: Группы студентов, Студенты, Дисциплины и Успеваемость. Для установления связей используем ключевые поля: КодГруппы, КодСтудентов и КодДисциплины. Например, между первичным ключом (КодГруппы) tables Группы студентов и вторичным ключом (КодГруппы) таблицы Студенты устанавливаем связь один-ко-многим. Прежде чем приступить к созданию логических связей надо в Окне редактирования закрыть все tables и перейти на вкладку *Работа с базами данных*. Затем щелкнуть на пиктограмме *Схема данных*, в окне редактирования появится активное диалоговое окно «Добавление таблицы» на фоне неактивного окна *Схема данных* (рисунок 44).

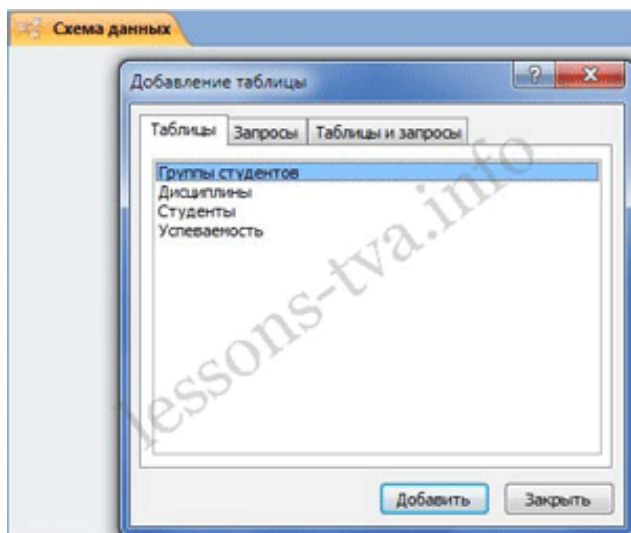


Рисунок 44 – Схема данных

В окне *Добавление таблиц* необходимо выделить имена таблиц и нажать кнопку *Добавить*, при этом в окне «Схема данных» появятся все таблицы (рисунок 45). После этого необходимо закрыть окно диалога.

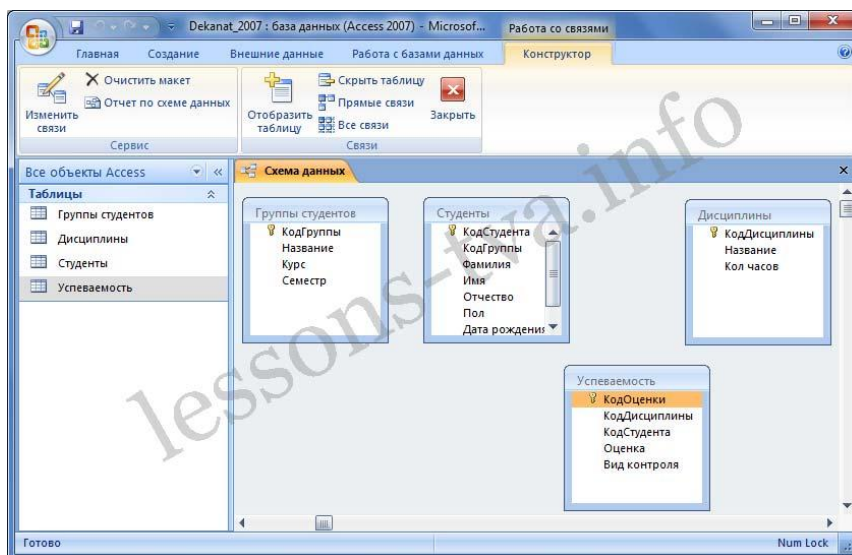


Рисунок 45 – Таблицы

Далее необходимо установить связи между таблицами в окне *Схема данных*. Для этого в окне *Схема данных* необходимо отбуксировать (переместить) поле *КодГруппы* из таблицы *Группы студентов* на соответствующее поле таблицы *Студенты*, в результате этой операции появится окно «Изменение связей» (рисунок 46).

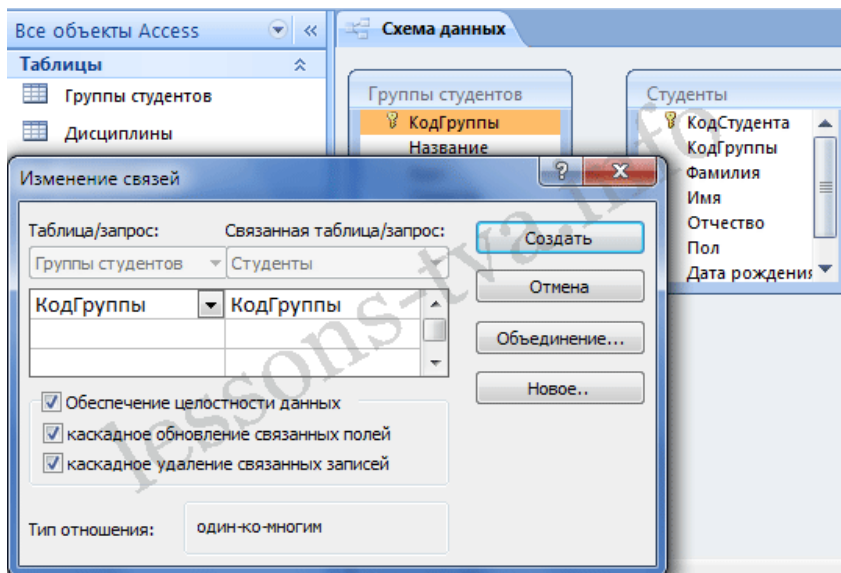


Рисунок 46 – Изменение связей

В появившемся окне диалога «Изменение связей» необходимо установить флажки: «Обеспечить целостность данных», «каскадное обновление связанных полей» и «каскадное удаление связанных записей», убедиться в том, что установлен тип отношений один-ко-многим и нажать кнопку *Создать*.

В окне *Схема данных* появится связь один-ко-многим между таблицами Группы студентов и Студенты. Аналогичным образом надо связать поля КодСтудента в таблицах Студенты и Успеваемость, а затем поля КодДисциплины в таблицах Успеваемость и Дисциплины. В итоге получим схему данных, представленную на рисунке 47.

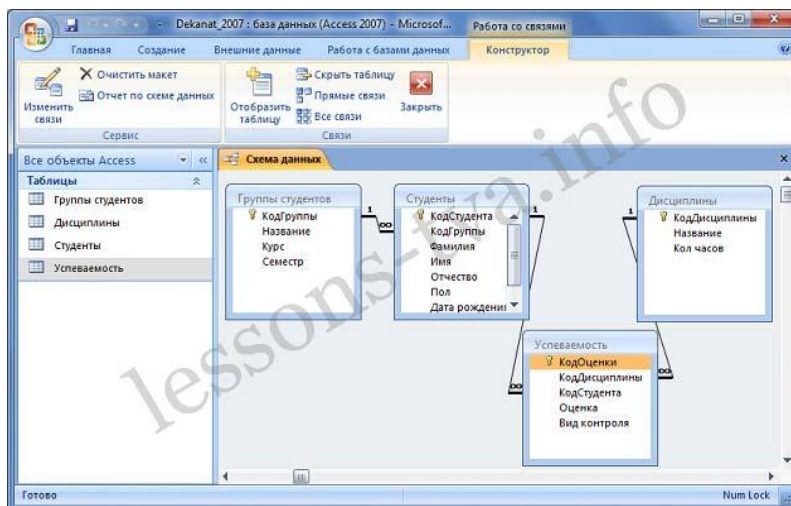


Рисунок 47 – Схема данных

После установки связей между таблицами, окно *Схема данных* необходимо закрыть. Далее необходимо осуществить заполнение всех таблиц. Заполнение целесообразно начинать с табл. Группы студентов, так как поле КодГруппы таблицы *Студенты* используется в качестве столбца подстановки для заполнения соответствующего поля таблицы *Студенты*.

Затем установить связи между таблицами *Студенты* и *Успеваемость*, *Дисциплины* и *Успеваемость*, так как поля КодСтуденты и КодДисциплины таблицы *Успеваемость* используется в качестве столбца подстановки для заполнения соответствующих полей таблицы *Успеваемость*.

14.2 Заполнение таблиц БД Access 2007

Таблицы БД Access 2007 необходимо заполнять данными в определенной последовательности. Сначала надо заполнять главные таблицы, а затем подчиненные. Заполнение таблиц для БД *Деканат* целесообразно выполнять в такой последовательности: *Группы студентов*, *Студенты*, *Дисциплины* и *Успевае-*

мость. В этом случае для заполнения полей вторичных ключей (КодГруппы, КодСтудентов, КодДисциплины) можно использовать раскрывающийся список данных. Раскрывающийся список данных для вторичных ключей появится только в том случае, если при создании структуры подчиненных таблиц в режиме конструктор для полей вторичных ключей выбран тип данных *Мастер подстановок*. Рассмотрим заполнение таблицы *Группы студентов*. В окне базы данных Деканат в области объектов выбираем таблицу *Группы студентов*, и выполняем на ней двойной щелчок. В окне редактирования появится структура таблицы *Группы студентов* в режиме таблицы. Новая таблица состоит из одной пустой строки. Заполнение производится по записям (по строкам). Поле счетчика (поле КодСтудента) заполняется автоматически. Переход к следующему полю осуществляется нажатием клавиши *Tab*. После ввода первой записи курсор смещается на следующую запись. После заполнения таблица *Группы студентов* имеет следующий вид (рисунок 48).

КодГруппы	Название	Курс	Семестр
1	ЭК-12а	1	1
2	ЭК-12б	1	1
3	ЭК-22	1	1
4	ЭК-32а	1	1
5	ЭК-32б	1	1
6	ЭК-52а	1	1
7	ЭК-52б	1	1
8	ЭК-82а	1	1
9	ЭК-82б	1	1
*	(No)		

Рисунок 48 – Таблица после заполнения

Аналогичным образом заполняется таблица *Студенты* (рисунок 49). Необходимо отметить, что в таблице *Студенты* поле КодГруппы заполняется из раскрывающегося списка.

КодСтудент	КодГруппы	Фамилия	Имя	Отчество	Пол	Дата рожде	Место рожд
1	ЭК-12а	Бабиченко	Елена	Ивановна	ж	12.06.1994 г.	Харьков
2	ЭК-12а	Краснова	Ирина	Васильевна	ж	07.11.1995 г.	Харьков
3	ЭК-12б	Ильин	Илья	Ильич	м	23.08.1993 с.	Красное Во
4	ЭК-12б	Иванов	Иван	Иванович	м	30.09.1994 г.	Киев
5	ЭК-22	Петрова	Зоя	Викторовна	ж	17.03.1995 г.	Днепропетр
6	ЭК-22	Воронына	Ольга	Ивановна	ж	21.05.1995 г.	Люботин Ха
7	ЭК-82б	Сидоров	Сидор	Сидорович	м	01.03.1994 г.	Полтава
8	ЭК-82а	Витязь	Петр	Петрович	м	13.02.1994 г.	Изюм Харьк
11	ЭК-82а	Ковтун	Сергей	Анатольевич	м	04.04.1994 г.	Луганск
12	ЭК-32а	Степаненко	Степан	Степанович	м	29.11.1994	Донецк
*	(No)						

Рисунок 49 – Таблица «Студенты»

Если раскрывающийся список не появляется, то необходимо закрыть таблицу *Студенты* и открыть окно *Схема данных*. В этом окне для таблицы *Студенты* разорвать связи, т.е. удалить отношения. Для этого щелкнуть правой кнопкой мыши на одну из связей и из контекстного меню выбрать команду *Удалить*. Затем удалить другую связь. Закрыть окно *Схема данных*. Открыть таблицу *Студенты* в режиме *Конструктор*. Из списка выбираем тип данных для поля КодГруппы *Мастер подстановок*. В процессе назначения типа данных *Мастер подстановок* для поля КодГруппы таблицы *Студенты* выбрать следующие данные: на первом шаге *Группы студентов*, а затем *Название* и далее щелкаем *Готово*. Закрыть таблицу *Студенты*. После этого восстановить связи (отношения) в окне *Схема данных*. Продолжаем заполнение таблицы *Студенты* с использованием раскрывающегося списка для поля КодГруппы.

Далее заполняется таблица *Дисциплины* (рисунок 50).

Дисциплины				
	КодДисцип. ▾	Название ▾	Кол час ▾	Добавить поле
+	1	Информатика	216	
+	2	Математика	108	
+	3	Иностран язык	108	
+	4	Политология	108	
+	5	Физкультура	108	
+	6	Микроэкономика	216	
+	7	Менеджмент	108	
+	8	Культурология	108	
*	(№)			

Рисунок 50 – Таблица «Дисциплины»

Затем заполняется таблица *Успеваемость* (рисунок 51).

Успеваемость						
	КодОценки ▾	КодДисцип. ▾	КодСтудент ▾	Оценка ▾	Вид контро. ▾	
		1	Информатика	Бабиченко	5/A	экзамен
		2	Информатика	Воронина	4/B	экзамен
		3	Математика	Бабиченко	5/B	экзамен
		4	Микроэконом	Петрова	3/D	экзамен
		5	Политология	Ильин	4/C	экзамен
		6	Физкультура	Краснова	5/A	зачет
		7	Менеджмент	Бабиченко	5/A	экзамен
		8	Математика	Иванов	4/C	экзамен
		9	Микроэконом	Бабиченко	5/B	экзамен
		10	Микроэконом	Иванов	5/A	экзамен
*		(№)				

Рисунок 51 – Таблица «Успеваемость»

В таблице *Успеваемость* поля КодДисциплины и КодСтудента заполняются из раскрывающихся списков. Необходимо отметить, что в базах данных Access 2007 применяются различные методы перемещения по таблице. Переходить от записи к записи можно с помощью: клавиш управления курсором; кнопок из области *Запись*, расположенных внизу таблицы в режиме таблицы; команды *Перейти* в группе *Найти*, расположенной на ленте. Для перемещения от поля к полю (слева направо) применяются клавиши *Tab* и *Enter*, а в обратном направлении *Shift+Tab*. Поиск данных в таблице большого объема, можно осуществлять командой поиск расположенный внизу таблицы в режиме таблицы или командой *Найти* в группе *Найти*, расположенной на ленте. Для замены данных в полях необходимо использовать команду *Заменить* в группе *Найти*, расположенной на ленте. После создания структуры таблиц, их заполнения и установления отношений (логических связей) между таблицами можно приступить к построению запросов.

14.3 Создание запросов и поиск информации в базе данных

В СУБД Access 2007 можно создавать запросы (queries) для отображения требуемых полей из записей одной или нескольких таблиц. В СУБД Access 2007 применяются различные типы запросов: на выборку, на обновление, на добавление, на удаление, перекрестный запрос (query), выполнение вычислений, создание таблиц. Наиболее распространенным является запрос на выборку. Применяются два типа запросов: запрос по образцу (QBE) и запрос на основе структурированного языка запросов (SQL). Запросы на выборку используются для отбора требуемой пользователю информации, содержащейся в нескольких таблицах. Они создаются только для связанных таблиц. Запросы могут основываться как на нескольких таблицах, так и существующих запросах. СУБД Access 2007 включает такие средства создания запросов, как *Мастер* и *Конструктор*. Кроме того, в СУБД Access 2007 существует множество средств для поиска и отображения информации, которая хранится в базе данных. Данные в таблицах можно отсортировать на основе любого поля или комбинации

полей. Для извлечения из базы данных необходимых записей можно отфильтровать таблицу, применив средства фильтрации. На скриншоте (рисунок 52) средства сортировки и фильтрации выделены скругленным прямоугольником красного цвета.

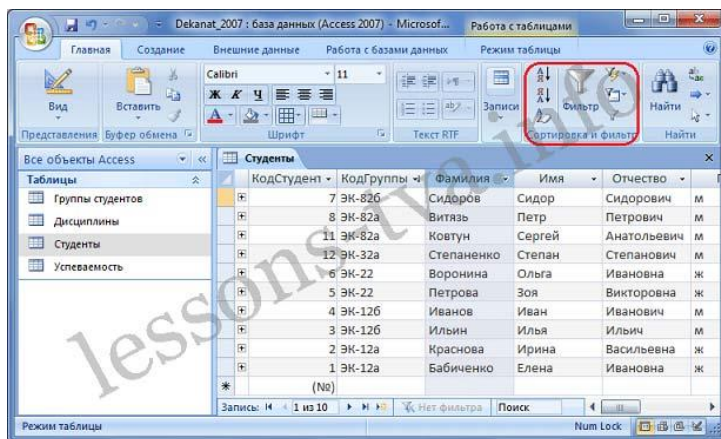


Рисунок 52 – Средства сортировки и фильтрации

Рассмотрим пример создания запроса на выборку с помощью *Конструктора*. Для создания нового пустого запроса в режиме конструктора щелкнуть на пиктограмме *Конструктор запросов* (рисунок 53).

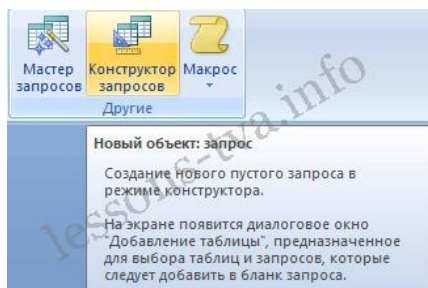


Рисунок 53 – Конструктор запросов

Откроется активное окно диалога *Добавление таблицы* (рисунок 54) на фоне неактивного окна «Запрос1». В этом окне можно выбрать таблицы и запросы для создания новых запросов.

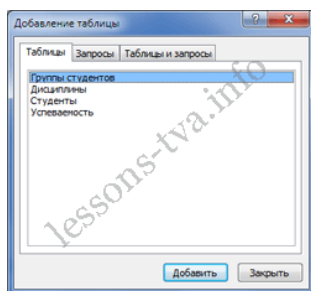


Рисунок 54 – Добавление таблицы

В окне *Добавление таблицы* следует выбрать несколько таблиц из представленного списка таблиц, на основе которых будет проводиться выбор данных, и щелкнуть на кнопке *Добавить*. После этого закрыть окно *Добавление таблицы*, а окно «Запрос1» станет активным (рисунок 55).

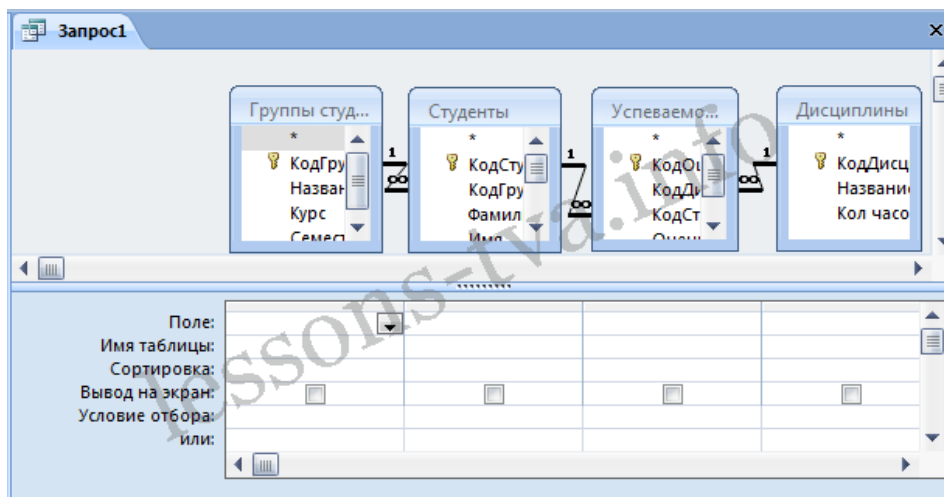


Рисунок 55 – Запрос 1

Окно *Конструктора* состоит из двух частей – верхней и нижней. В верхней части окна размещается схема данных запроса, которая содержит список связанных таблиц. В нижней части окна находится *Бланк построения запроса* QBE, в котором каждая строка выполняет определенную функцию. Переместим имена полей с таблиц-источников в *Бланк*. Из таблицы *Группы студентов* переместим поле *Название* в первое поле *Бланка*, из таблицы *Студенты* переместим поле *Фамилии* во второе поле, а из таблицы *Успеваемость* переместим поле *Оценка* в третье поле и из таблицы *Дисциплины* переместим поле *Название* в четвертое поле *Бланка запросов*. При необходимости можно задать принцип сортировки (по возрастанию или по убыванию) результатов запроса. В строке «Вывод на экран» автоматически устанавливается флажок просмотра информации. Условия ограниченного поиска или критерий поиска информации вводятся в строке «Условия» отбора и строке «Или». Например, введем критерий поиска - «5/A» в строке «Условия» для поля Оценка. В этом случае в результате выполнения запроса на экране будут отображаться все фамилии студентов, которые получили оценку 5/A (рисунок 56).

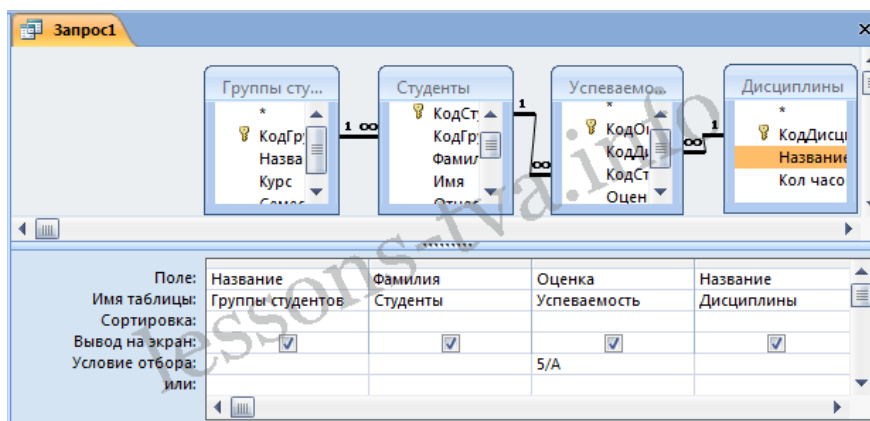


Рисунок 56 – Запрос с критерием поиска

Далее закрыть окно запроса Запрос1, появится окно диалога *Сохранить*, ответить «Да» и ввести имя запроса, например «Успеваемость студентов». Для запуска запроса дважды щелкнем на запрос «Успеваемость студентов», откроется таблица с результатами выполненного запроса (рисунок 57).

Группы студ.	Фамилия	Оценка	Дисциплины.На
ЭК-12а	Бабиченко	5/А	Информатика
ЭК-12а	Бабиченко	5/А	Менеджмент
ЭК-12а	Краснова	5/А	Физкультура
ЭК-12б	Иванов	5/А	Микроэкономика
*			

Рисунок 57 – Результаты выполненного запроса

Далее создаем параметрический запрос или запрос с параметрами. Создаем этот запрос также как и предыдущий, в режиме конструктора, но только в строке *Условия отбора* для поля *Фамилия* введем условие отбора в виде приглашения в квадратных скобках, например, *[Введите фамилию]*. В этом случае в результате выполнения запроса на экране будет отображаться фамилия студента и все дисциплины, по которым он получил оценку. Закрыть окно запроса на выборку. На вопрос о сохранении изменения ответить - Да и ввести имя запроса, например «Параметрический запрос». Запустим Параметрический запрос, дважды щелкнув на нем. В открывшемся на экране окне диалога «Введите значение параметра» надо ввести фамилию студента, информацию об успеваемости которого необходимо получить (рисунок 58).

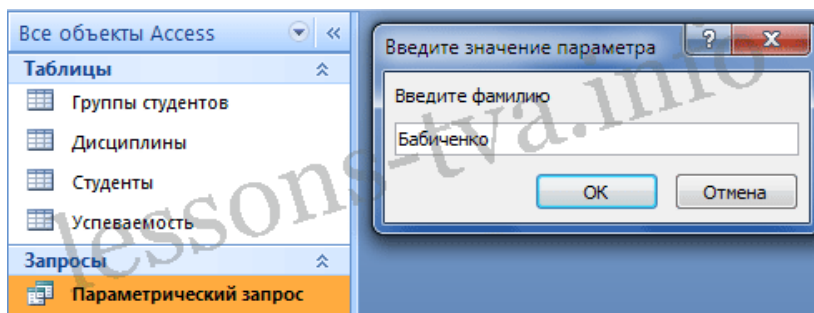


Рисунок 58 – Запрос на выборку

Затем щелкнуть на кнопке ОК, откроется таблица с результатами выполненного запроса (рисунок 59).

Группы студ.	Фамилия	Оценка	Дисциплины.На
ЭК-12а	Бабиченко	5/А	Информатика
ЭК-12а	Бабиченко	5/А	Менеджмент
ЭК-12а	Бабиченко	5/В	Математика
ЭК-12а	Бабиченко	5/В	Микроэкономика
*			

Рисунок 59 – Результат запроса на выборку

В некоторых случаях для создания запросов можно использовать *Мастер запросов*. После создания запросов на выборку информации из БД Access 2007 можно приступить к формированию форм.

14.4 Создание и использование форм для ввода данных в таблицы базы данных MS Access 2007

В Access 2007 можно вводить данные непосредственно в таблицу в режиме таблица. Но обычно для ввода данных в БД Access 2007 используют формы (forms). Формы ускоряют работу с базой данных. Форма в БД — это структурированное интерактивное окно с элементами управления, в котором отображаются поля одной или нескольких таблиц или запросов. Форму можно использовать для ввода, изменения или отображения данных из таблицы или запроса. В Microsoft Office Access 2007 предусмотрены новые средства, помогающие быстро создавать формы, а также новые типы форм и функциональные возможности. Формы в БД Access можно создавать с помощью различных средств: инст-

румента *Формы*; инструмента *Разделенная форма*; инструмента *Несколько элементов*; инструмента *Пустая форма*; *Мастера форм*; *Конструктора форм*. Все средства создания форм помещены в группу *Формы* на вкладке *Создание* (рисунок 60).

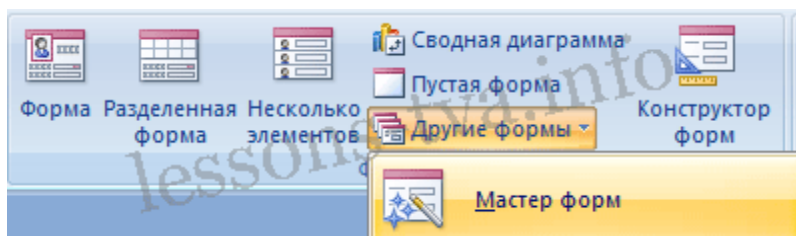


Рисунок 60 – Мастер форм

Формы, которые используют таблицы целесообразно выполнять с помощью *Мастера* или указанных инструментов, а дорабатывать их, т.е. вносить необходимые изменения, можно в режиме макета или конструктора. В Microsoft Access 2007 предусмотрено два режима внесения изменений и дополнений в формы: режим макета и режим конструктора. Переход между режимами (открыть, режим макета, конструктор) можно выполнить, щелкнув правой кнопкой мыши форму в области переходов, а затем выбрать нужный режим в контекстном меню. Режим макета — это более наглядный режим редактирования (изменения) форм, чем режим конструктора. В режиме макета изменения выполняются фактически в реальной форме, поэтому в этом режиме целесообразно выполнять более простые изменения, связанные с ее внешним видом. В тех случаях, когда в режиме макета невозможно выполнить изменения в форме, целесообразно применять режим конструктора. Режим конструктора предоставляет пользователю более широкие возможности для редактирования (изменения) форм, в этом режиме можно добавлять поля, настраиваемые элементы и составлять программы.

Инструмент *Форма*. Для быстрого создания формы, т.е. создания одним щелчком мыши можно воспользоваться инструментом *Форма*. В этом случае надо выделить таблицу в области объектов. Затем перейти на вкладку *Создание* и щелкнуть на пиктограмме *Форма*. На экране будет отображена форма (рисунок 61).

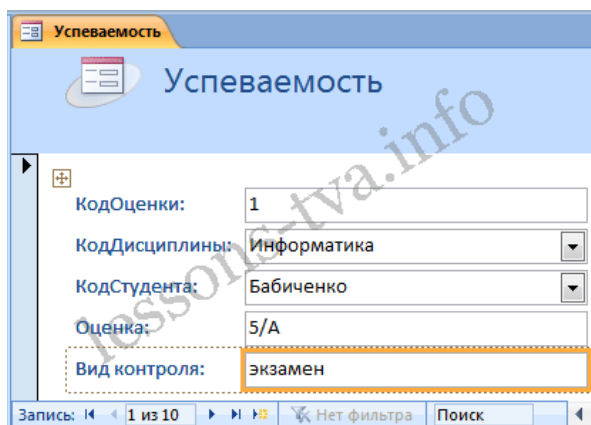


Рисунок 61 – Форма «Успеваемость»

Если Access обнаруживает одну таблицу, связанную отношением «одно-многим» с таблицей или запросом, который использовался для создания формы, Access добавляет таблицу данных в форму, основанную на связанной таблице или запросе. Если таблица данных в форме не нужна, ее можно удалить.

Средство «Разделенная форма». Разделенная форма — новая возможность в Microsoft Access 2007, которая позволяет одновременно отображать данные в режиме формы и в режиме таблицы. В области объектов (переходов) выделить таблицу, например, *Успеваемость*. Далее щелкнуть на пиктограмме *Разделенная форма* на вкладке *Создать*. На экране будет отображена форма (рисунок 62).

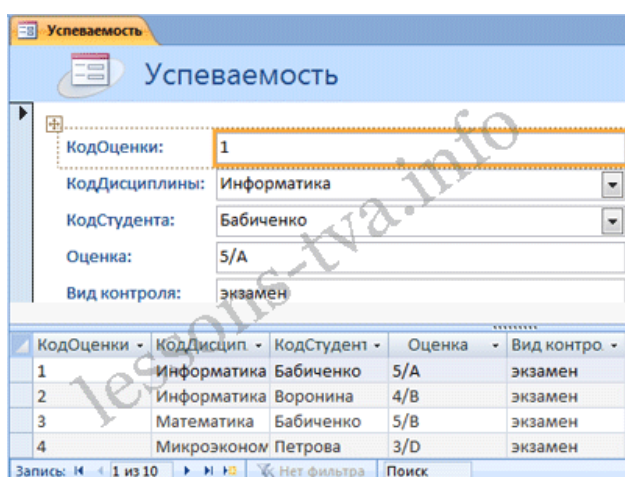


Рисунок 62 – Разделённая форма

Инструмент «Несколько элементов». Форму, в которой отображается не одна, а одновременно несколько записей, можно создать инструментом «Не-

сколько элементов». Чтобы создать данную форму выделим в области объектов (переходов) следует открыть одну из таблиц (например, *Успеваемость*). Затем перейти на вкладку *Создание* и щелкнуть на пиктограмме *Несколько элементов*. На экране будет отображена форма (рисунок 63) в режиме макета.

КодОценки	КодДисциплины	КодСтудента	Оценка	Вид контроля
1	Информатика	Бабиченко	5/A	экзамен
2	Информатика	Воронина	4/B	экзамен
3	Математика	Бабиченко	5/B	экзамен
4	Микроэкономика	Петрова	3/D	экзамен
5	Политология	Ильин	4/C	экзамен
6	Физкультура	Краснова	5/A	зачет
7	Менеджмент	Бабиченко	5/A	экзамен
8	Математика	Иванов	4/C	экзамен
9	Микроэкономика	Бабиченко	5/B	экзамен

Рисунок 63 – Форма в режиме макета

Форма похожа на таблицу, в ней одновременно отображаются несколько записей. Но эта форма предоставляет возможности для настройки, так как она отображается в режиме макета. В режиме макета можно легко осуществлять доработку формы (например, добавлять элементы управления и т.д.).

Средство «Пустая форма». Этот инструмент можно использовать в том случае, если необходимо быстро создать форму с несколькими полями. Форма открывается в режиме «Работа с макетами форм» и при этом отображается область *Список полей* (рисунок 64).

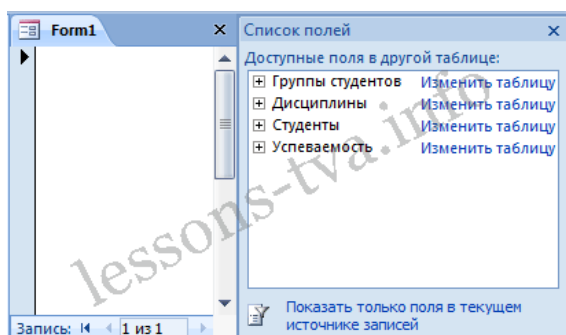


Рисунок 64 – Список полей

Мастер форм. Создание форм при помощи мастера форм осуществляется быстро, и это средство позволяет включить в форму поля из нескольких связанных таблиц или запросов. На вкладке *Создание* в группе *Формы* надо нажать кнопку *Другие формы*, а затем выбрать команду *Мастер форм*. Откроется окно

диалога *Создание форм*, в котором необходимо отвечать на вопросы каждого текущего экрана *Мастера* и щелкать на кнопке *Далее*. В первом окне необходимо выбрать поля из источника данных (таблиц или запросов). Для этого надо открыть список *Таблицы и запросы*, щелкнув на кнопку, справа. Например, выберем из списка таблицу *Студенты*.

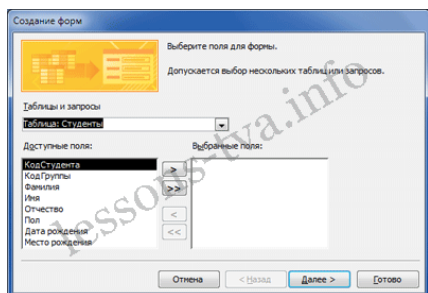


Рисунок 65 – Выбор поля

Затем все «Доступные поля» переведем в «Выбранные поля», выделив их и щелкнув на кнопку >>. Необходимо отметить, что, если форма создается на основе нескольких таблиц, необходимо повторить действия для каждой таблицы-источника. Затем необходимо щелкнуть на кнопке *Далее*. В следующем окне надо выбрать внешний вид, например в один столбец и щелкнуть *Далее*. В следующем окне выберем требуемый стиль — официальный. После выбора стиля, требуется перейти в последнее окно, щелкнув на кнопке *Далее*. В последнем окне *Мастера* требуется ввести имя (например, *Студенты_мастер_форм*) и указать дальнейшие действия: открыть форму для просмотра и ввода данных; изменить макет формы. После ввода имени формы (например, *Студенты*), выбора режима: «Открыть форму для просмотра и ввода данных» и щелчка на кнопке *Готово*, получим следующую форму для ввода и просмотра записей в таблицу *Студенты*.

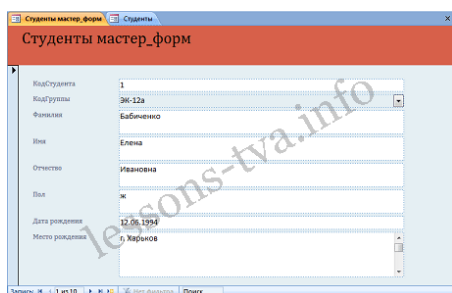


Рисунок 66 – Форма для ввода и просмотра записей в таблицу *Студенты*

Конструктор форм. Для создания новой пустой формы Студенты необходимо выполнить следующее. В окне приложения Access 2007 выбрать вкладку *Создание*. Выполнить щелчок на пиктограмме «Конструктор форм». В окне редактирования появится окно *Форма1* с пустой областью данных. Для отображения списка полей требуемой таблицы выполнить щелчок на пиктограмме «Добавить существующие поля», появится список таблиц. Щелкнув на знак «+» таблицы (например, *Студенты*), откроется список необходимых полей (рисунок 67).

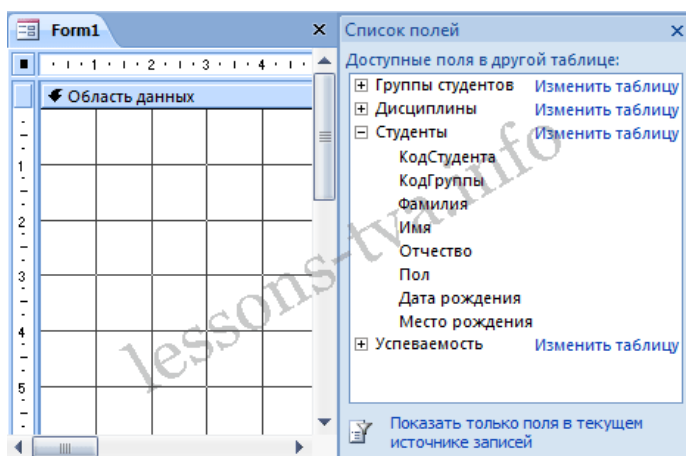


Рисунок 67 – Список полей

Поля из списка переместить на форму. Добавление полей осуществляется при нажатой левой кнопки мыши. Поместить поля на форму (рисунок 68).

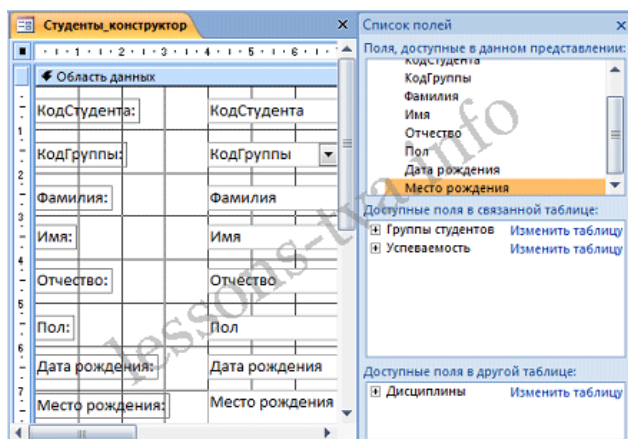


Рисунок 68 – Добавление полей на форму

Перемещение полей и их имен по форме производится следующим образом. Выделить поле с именем щелчком мыши. Вокруг него появятся маркеры перемещения и изменения размеров. Перемещать поле можно вместе с привязанным к нему именем или отдельно от него. Для перемещения поместить ука-

затель мыши на квадратик, находящийся в левом верхнем углу элемента. Указатель мыши в виде четырех направленной стрелки позволяет перемещать объект. Нажать кнопку мыши и, удерживая ее, буксировать поле или его имя в нужное место в форме. Затем отпустить кнопку мыши. Для изменения надписи, связанной с полем необходимо выполнить на ней двойной щелчок мышью и выполнить необходимые изменения. Затем закрыть окно. Для изменения размеров поместить курсор на размерные маркеры, при этом курсор примет вид двунаправленной стрелки. Нажать кнопку мыши, буксировать в нужном направлении, затем отпустить кнопку мыши. Для удаления поля выделить его, нажать клавишу *Delete* или другим способом.

Сохранить форму. Просмотреть форму *Студенты_конструктор*, выполнив на ней двойной щелчок в области переходов.

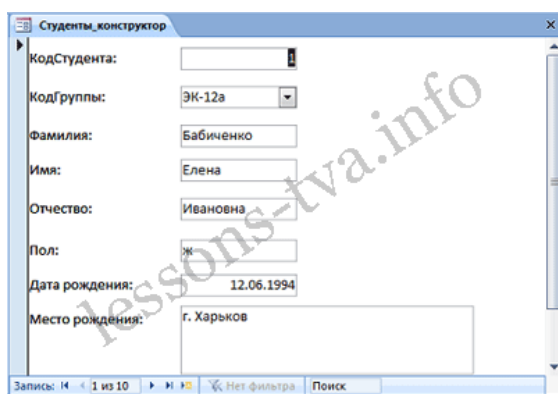


Рисунок 69 – Просмотр формы

Если вид формы не удовлетворяет, ее можно открыть в режиме Конструктор и внести необходимые изменения, затем сохранить.

14.5 Отчеты в базе данных Access 2007

Отчет (report) — это объект базы данных, который используется для вывода на экран, в печать или файл структурированной информации. Reports позволяют извлечь из таблиц или запросов базы данных необходимую информацию и представить ее в виде удобном для восприятия. Отчёт содержит заголовок, область данных, верхний и нижний колонтитулы, примечание и разбит на страницы.

В Microsoft Access 2007 для создания отчетов можно использовать различные средства: *Мастер отчетов*; *Конструктор отчетов*; *Инструмент Report*; *Пустой report*.

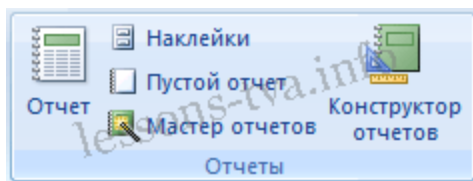


Рисунок 70 – Группа «Отчёты»

Отчеты целесообразно выполнять с помощью *Мастера* или других указанных инструментов, а дорабатывать их, т.е. вносить необходимые изменения можно в режиме макета или конструктора. В Microsoft Access 2007 предусмотрено два режима внесения изменений и дополнений в reports: режим макета и режим конструктора. Режим макета — это более наглядный режим редактирования и форматирования (изменения) отчетов, чем режим конструктора. В тех случаях, когда в режиме макета невозможно выполнить изменения в отчете, то целесообразно применять режим конструктора.

Мастер отчетов. Для создания отчета при помощи *Мастера отчетов* необходимо выполнить следующие действия. В окне базы данных Access щелкнуть на вкладке *Создание* и затем щелкнуть на кнопке *Мастер отчетов* в группе *Отчеты*. Появится диалоговое окно *Создание отчетов*.

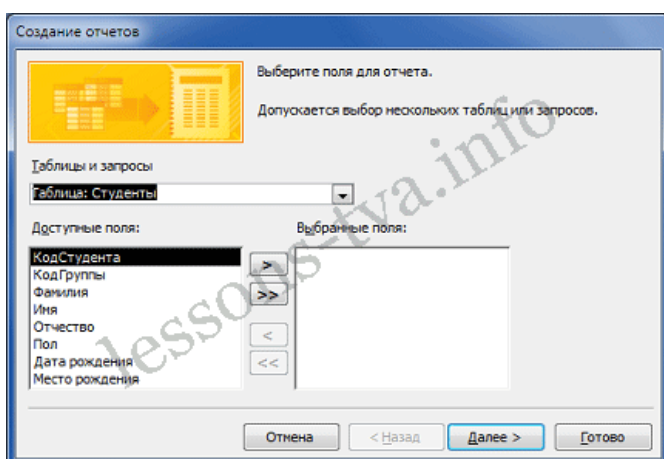


Рисунок 71 – Окно «Создание отчётов»

В поле *Таблицы и отчеты* щелкнуть на стрелке и выбрать в качестве источника данных таблицу *Студенты*. Щелкнуть на кнопке *ОК*. Все «Доступные

поля» переведем в «Выбранные поля», выделив их и щелкнув на кнопку >>. На следующем шаге (Добавить уровни группировки) щелкнуть *Далее*. На шаге «Выберите порядок сортировки записей». В раскрывающемся списке выбрать «Фамилия» для сортировки по возрастанию. На шаге «Выберите вид макета для отчета». Выбрать: макет – блок, ориентация – книжная. Щелкнуть на кнопке *Далее*. На шаге «Выберите требуемый стиль». Выбрать – *Изящная*. Следующий шаг «Задайте имя отчета». Ввести имя – *Студенты_мастер_отчетов*. Дальнейшие действия: просмотреть отчёт; изменить макет отчета. Выбрать *Просмотреть*, щелкнуть на кнопке *Готово*. Отчёт открывается в режиме *Предварительного просмотра*, который позволяет увидеть, как будет выглядеть отчёт в распечатанном виде.

КодГр	КодСт	Фамилия	Имя	Отчество	Пол	Дата рожд.	Место рождения
1	2	Краснова	Ирина	Васильевна	ж	07.11.1995	г. Харьков
1	Бабиченко	Елена	Ивановна	ж	12.05.1994	г. Харьков	
2	4	Иванов	Иван	Иванович	м	30.09.1994	г. Киев
3	3	Ильин	Илья	Ильич	м	23.08.1993	с. Красное Волчанского р-на Харьковской обл.
3	6	Воронова	Ольга	Ивановна	ж	21.05.1995	г. Лобовин Харьковской обл.
5	Петрова	Зоя	Викторовна	ж	17.03.1995	г. Днепропетровск	
4	12	Степаненко	Степан	Степанович	м	29.11.1994	Донецк
8	8	Вилъяв	Петр	Петрович	м	13.02.1994	г. Изюм Харьковской области
11	Ковтун	Сергей	Анагольевич	м	04.04.1994	г. Луганск	
9	7	Сидоров	Сидор	Сидорович	м	01.03.1994	г. Полтава

Рисунок 72 – Режим предварительного просмотра

Перейти в режим *Конструктора* и выполнить редактирование и форматирование отчета. Для перехода из режима предварительного просмотра в режим конструктора необходимо в области переходов щелкнуть правой кнопкой мыши на имени отчета и в контекстном меню выбрать режим конструктора. На экране появится герогт в режиме *Конструктора* (рисунок 73).

КодГр	КодСт	Фамилия	Имя	Отчество	Пол	Дата рожд.	Место рождения
КодГр	КодСт	Фамилия	Имя	Отчество	Пол	Дата рожд.	Место рождения

Рисунок 73 – Режим конструктора

Редактирование: удалите поля КодСтудента в верхнем колонтитуле и области данных; удалите поля КодГруппы в верхнем колонтитуле и области данных; на место поля КодГруппы переместить поле «Название» из списка «Группы студентов»; переместите влево все поля в верхнем колонтитуле и области данных; измените надпись в заголовке страницы, введите название своего учебного заведения, нажмите *Enter*; переместите *Надпись*. В Нижнем колонтитуле выделить поле =Now() и перетащить его в Заголовок под название Студенты. Дата будет отображаться под заголовком.

Форматирование: выделите заголовок Студенты; измените гарнитуру, начертание и цвет шрифта, а также цвет заливки фона. Report в режиме конструктора примет вид, представленный на рисунке 74.

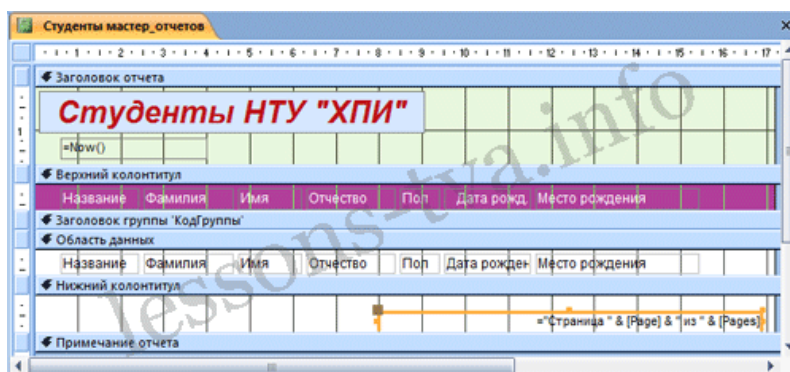


Рисунок 74 – Результат

Перейдите в режим предварительного просмотра. Для перехода в режим предварительного просмотра необходимо щелкнуть правой кнопкой мыши на названии отчета в области переходов и в контекстном меню выбрать режим «Предварительный просмотр» (рисунок 75).



Рисунок 75 – Режим предварительного просмотра

Инструмент «Отчет». Для быстрого создания отчета, т.е. создания одним щелчком мыши можно воспользоваться инструментом *Отчёт*. В этом случае отчёт формируется на базе существующей таблицы или запроса. В созданном отчете будут отображаться все записи таблицы или запроса, на базе которых создается отчёт. Но созданный отчёт можно будет изменить в режиме макета или конструктора. Для создания отчета необходимо выполнить следующее. В области переходов надо выделить таблицу (например, *Студенты*), на основе которой нужно создать отчёт. Затем перейти на вкладку *Создание* и щелкнуть на пиктограмме *Отчет*. На экране будет отображен простой отчет на основе текущей таблицы *Студенты*.

Средство «Пустой отчет» позволяет создавать отчёт с нуля в режиме макета. Для этого следует щелкнуть *Пустой отчёт* в группе *Отчеты* на вкладке *Создание*. В окне редактирования Access 2007 появится *Отчет1* с пустой областью данных, а в правой части окна будет отображаться область «Список полей» существующих таблиц. Щелкнув на знак «+» таблицы (например, *Студенты*), откроется список необходимых полей. Перетащить требуемые поля из этого списка в отчёт, нажав и удерживая левую клавишу мыши. С помощью инструментов из группы «Элементы управления» на вкладке *Формат*, можно доработать отчёт, добавив заголовок, номера страниц, дату и время. При необходимости его можно доработать в режиме конструктора. Сохранить отчёт.

15 Лекция 15. Организация глобальных сетей

15.1 История развития глобальных сетей

Из истории человеческого общества вам должно быть известно, что многие научные открытия и изобретения сильно повлияли на ее ход, на развитие цивилизации. К их числу относятся изобретение парового двигателя, открытие электричества, овладение атомной энергией, изобретение радио и пр. Процессы резкого изменения в характере производства, в быту, к которым приводят важные научные открытия и изобретения, принято называть научно-технической

революцией. Появление и развитие компьютерной техники во второй половине XX века стали важнейшим фактором научно-технической революции.

В этом процессе выделим три этапа.

Первый этап начался с создания первой ЭВМ в 1945 году. Приблизительно в течение 30 лет компьютерами пользовалось сравнительно небольшое число людей, главным образом в научной и производственной областях.

Второй этап начался в середине 70-х годов XX века и связан с появлением и распространением персональных компьютеров (ПК). ПК стали широко использоваться не только в науке и производстве, но и в системе образования, сфере обслуживания, быту. ПК вошли в дом как один из видов бытовой техники наряду с радиоприемниками, телевизорами, магнитофонами.

Третий этап связан с появлением глобальной компьютерной сети Интернет. В результате персональный компьютер, который помещается на письменном столе, стал «окном» в огромный мир информации. Появились новые понятия, такие как «мировое информационное пространство», «киберпространство». Распространение Интернета решает важную социальную проблему информационного неравенства, которое существует между людьми, живущими в разных странах, на разных континентах, а также в крупных городах и на периферии. Именно развитие Интернета дает основание говорить о том, что в истории цивилизации наступает этап информационного общества.

С распространением компьютеров возникает понятие компьютерной грамотности. Это необходимый уровень знаний и умений человека, позволяющий ему использовать компьютер для общественных и личных целей. На первом этапе развития ЭВМ компьютерная грамотность сводилась к умению программировать. Программирование главным образом изучалось в высших учебных заведениях, владели им ученые, инженеры, профессиональные программисты. На втором этапе под общим уровнем компьютерной грамотности стали понимать умение работать на персональном компьютере с прикладными программами, выполнять минимум необходимых действий в среде операционной системы. Компьютерная грамотность на таком уровне становится массовым явлением.

нием благодаря обучению в школе, на многочисленных курсах и самостоятельно. На третьем, современном этапе важным элементом компьютерной грамотности стало умение использовать Интернет. Появилось более широкое понятие — информационная культура. Быстро растет число компьютеров, подключенных к мировой сети. И всё более необходимым становится умение использовать компьютер для общения с другими людьми, для дистанционного обучения, поиска справочной информации, коммерческой деятельности и многого другого.

Понятие глобальной сети — системы объединенных компьютеров, расположенных на больших расстояниях друг от друга, — появилось в процессе развития компьютерных сетей. В 1964 году в США была создана компьютерная система раннего оповещения о приближении ракет противника. Первой глобальной сетью невоенного назначения стала сеть ARPANET в США, введенная в действие в 1969 году. Она имела научное назначение и объединяла компьютеры нескольких университетов страны. В 80-90-х годах прошлого века в разных странах создается множество отраслевых, региональных национальных компьютерных сетей. Их объединение в международную сеть произошло на базе межсетевой среды Интернет. Важным годом в истории Интернета стал 1993 год, когда была создана служба World Wide Web (WWW) — Всемирная информационная сеть (Всемирная паутина). С появлением WWW резко возрос интерес к Интернету, пошел процесс его бурного развития и распространения. Многие люди, говоря об Интернете, подразумевают именно WWW, хотя это только лишь одна из его служб.

Интернет — это сложная аппаратно-программная система.

Постараемся получить ответы на три вопроса: из чего состоит Интернет; как работает; для чего используется? Часто в литературе вместо слова «Интернет» употребляют термин «Сеть» (уважительно с большой буквы). Мы также иногда будем им пользоваться.

15.2 Аппаратные средства Интернета

Основными составляющими любой глобальной сети являются компьютерные узлы и каналы связи. Здесь можно провести аналогию с телефонной сетью: узлами телефонной сети являются АТС — автоматические телефонные станции, которые между собой объединены линиями связи и образуют городскую телефонную сеть. Телефон каждого абонента подключается к определенной АТС. К узлам компьютерной сети подключаются персональные компьютеры пользователей подобно тому, как с телефонными станциями соединяются телефоны абонентов. Причем в роли абонента компьютерной сети может выступать как отдельный человек через свой ПК, так и целая организация через свою локальную сеть. В последнем случае к узлу подключается сервер локальной сети.

Организация, предоставляющая услуги обмена данными с сетевой средой, называется провайдером сетевых услуг. Английское слово *provider* означает «поставщик», «снабженец». Пользователь заключает договор с провайдером на подключение к его узлу и в дальнейшем оплачивает ему предоставляемые услуги (подобно тому, как мы оплачиваем услуги телефонной сети).

Узел содержит один или несколько мощных компьютеров, которые находятся в состоянии постоянного подключения к сети. Информационные услуги обеспечиваются работой программ-серверов, установленных на узловых компьютерах. Каждый узловой компьютер имеет свой постоянный адрес в Интернете; он называется IP-адресом. IP-адрес состоит из четырех десятичных чисел, каждое в диапазоне от 0 до 255, которые записываются через точку. Например, 193.126.7.29 или 128.29.15.124. Такие же IP-адреса получают и компьютеры пользователей Сети, но они действуют лишь во время подключения пользователя к сети, т. е. изменяются в каждом новом сеансе связи, в то время как адреса узловых компьютеров остаются неизменными. Наряду с цифровыми IP-адресами в Интернете действует система символьных адресов, более удобная и

понятная для пользователей. Она называется доменной системой имен (DNS — DomainNameSystem).

Например, IP-адресу 87.242.99.97 сервера методической службы издательства «БИНОМ. Лаборатория знаний» соответствует доменное имя `metodist.Lbz.ru`. Данное имя состоит из трех доменов, разделенных точками. Система доменных имен построена по иерархическому принципу. Первый справа домен (его еще называют суффиксом) — домен верхнего уровня, следующий за ним — домен второго уровня и т. д. Последний (первый слева) — имя компьютера. Домены верхнего уровня бывают географическими (двухбуквенными) или административными (трехбуквенными). Например, российской зоне Интернета принадлежит географический домен `ru`. Еще примеры: `uk` — домен Великобритании; `ca` — домен Канады; `de` — домен Германии; `jp` — домен Японии. Административные домены верхнего уровня чаще всего относятся к американской зоне Интернета: `gov` — правительственная сеть США; `mil` — военная сеть; `edu` — образовательная сеть; `com` — коммерческая сеть. Среди узлов Интернета есть своя иерархия. Например, некоторый узел в Самаре имеет соединение с узлом в Москве, который, в свою очередь, связан с рядом узлов европейской опорной сети. Последние имеют связь с узлами США, Японии и др. И всё-таки структура Интернета — это не дерево, а именно сеть. Как правило, каждый узел имеет связь не с одним, а с множеством других узлов. Поэтому маршруты, по которым поступает информация на некоторый узел, могут быть самыми разными. Этим обеспечивается устойчивость работы Сети: при выходе из строя одного узла информационные потоки к другим узлам не прерываются. Они лишь могут изменить свои маршруты.

15.3 Каналы связи

Существуют самые разные технические способы связи в глобальной сети: телефонные линии; электрическая кабельная связь; оптоволоконная кабельная связь; радиосвязь (через радиорелейные линии, спутники связи). Различные ка-

налы связи различаются тремя основными свойствами: пропускной способностью, помехоустойчивостью, стоимостью.

По параметру стоимости самыми дорогими являются оптоволоконные линии, самыми дешевыми — телефонные. Однако с уменьшением цены снижается и качество работы линии: уменьшается пропускная способность, сильнее влияют помехи. Практически не подвержены помехам оптоволоконные линии. Пропускная способность — это максимальная скорость передачи информации по каналу. Обычно она выражается в килобитах в секунду (Кбит/с) или в мегабитах в секунду (Мбит/с). Пропускная способность телефонных линий — десятки и сотни Кбит/с; пропускная способность оптоволоконных линий и линий радиосвязи измеряется десятками и сотнями Мбит/с.

На протяжении многих лет большинство пользователей Сети подключались к узлу через коммутируемые (т. е. переключаемые) телефонные линии. Такое подключение производится с помощью специального устройства, которое называется модемом. Слово «модем» — это объединение сокращений двух слов: «модуля- тор» — «демодулятор». Модем устанавливается как на компьютере пользователя, так и на узловом компьютере. Модем выполняет преобразование дискретного сигнала (выдаваемого компьютером) в непрерывный (аналоговый) сигнал (используемый в телефонной связи) и обратное преобразование. Основной характеристикой модема является предельная скорость передачи данных. В разных моделях она колеблется в диапазоне от 1200 до 56 000 бит/с. Кабельная связь обычно используется на небольших расстояниях (между разными провайдерами в одном городе). На больших расстояниях выгоднее использовать радиосвязь. Всё большее число пользователей в наше время переходят от коммутируемых низкоскоростных подключений к высокоскоростным некоммутируемым линиям связи.

15.4 Программное обеспечение Интернета

Работа Сети поддерживается определенным программным обеспечением (ПО). Это ПО функционирует на серверах и на персональных компьютерах

пользователей. Как вам известно из курса информатики основной школы, основой всего программного обеспечения компьютера является операционная система, которая организует работу всех других программ. Программное обеспечение узловых компьютеров очень разнообразно. Условно его можно разделить на базовое (системное) и прикладное. Базовое ПО обеспечивает поддержку работы сети по протоколу TCP/IP — стандартному набору протоколов Интернета, т. е. оно решает проблемы рассылки и приема информации. Прикладное ПО занимается обслуживанием разнообразных информационных услуг Сети, которые принято называть службами Интернета. Служба объединяет серверы и клиентские программы, обменивающиеся данными по некоторым прикладным протоколам. Для каждой службы существует своя сервер-программа: для электронной почты, для телеконференций, для WWW и пр. Узловой компьютер выполняет функцию сервера определенной службы Интернета, если на нем работает сервер-программа этой службы. Один и тот же компьютер в разное время может выполнять функции сервера различных услуг; всё зависит от того, какая сервер-программа на нем в данный момент выполняется.

На ПК пользователей сети обслуживанием различных информационных услуг занимаются программы-клиенты. Примерами популярных клиентов являются: Outlook Express — клиент электронной почты, Internet Explorer — клиент службы WWW (браузер). Во время работы пользователя с определенной службой Интернета между его программой-клиентом и соответствующей программой-сервером на узле устанавливается связь. Каждая из этих программ выполняет свою часть работы в предоставлении данной информационной услуги. Такой способ работы Сети называется технологией «клиент — сервер».

15.5 Как работает Интернет

В Интернете используется пакетная технология передачи информации. Чтобы в этом лучше разобраться, представьте себе следующую ситуацию. Вам нужно переслать товарищу в другой город какой-то многостраничный документ (например, распечатку романа, который вы сочинили). Полностью в кон-

верт весь ваш роман не помещается, а посылать бандеролью вы не хотите — слишком долго будет идти. Тогда вы делите весь документ на части по 4 листа, вкладываете каждую часть в почтовый конверт, на каждом конверте пишете адрес и всю эту пачку конвертов опускаете в почтовый ящик. Например, если ваш роман занимает 100 страниц, то вам придется отправить 25 конвертов. Вы даже можете опустить конверты в разные почтовые ящики на разных узлах связи (для интереса, чтобы узнать, какие дойдут быстрее). Но поскольку на них указан один и тот же адрес, все конверты должны дойти до вашего товарища. А еще, чтобы товарищу было удобно собрать роман целиком, на конвертах желательно указать порядковые номера.

Аналогично работает пакетная передача информации в Интернете. За ее работу отвечает протокол TCP/IP, о котором уже говорилось раньше. Пора разобраться, что же обозначают эти загадочные буквы.

Фактически речь идет о двух протоколах. Первый — TCP-протокол расшифровывается так: Transmission Control Protocol — протокол управления передачей. Именно согласно этому протоколу всякое сообщение, которое нужно передать по Сети, разбивается на части. Эти части называются TCP-пакетами. Для доставки пакеты передаются протоколу IP, который к каждому пакету дописывает IP-адрес его доставки и еще некоторую служебную информацию. Таким образом, TCP-пакет — это аналог конверта с «кусочком» романа и адресом получателя. Каждый такой пакет будет самостоятельно перемещаться по сети независимо от других, но все они вместе соберутся у адресата. Далее, согласно протоколу TCP, происходит обратный процесс: из отдельных пакетов собирается исходное сообщение. Здесь, очевидно, необходимы те самые порядковые номера на конвертах; аналогичные номера содержатся и в TCP-пакетах. Если какой-то из пакетов не дошел или был испорчен при транспортировке, его передача будет запрошена повторно. Согласно протоколу TCP, передаваемое сообщение разбивается на пакеты на отправляющем сервере и восстанавливается в исходном виде на принимающем сервере.

Назначение IP-протокола (Internet Protocol) — доставка каждого отдельного пакета до места назначения. Пакеты передаются, как эстафетные палочки, от одного узла к другому. Причем маршруты для разных пакетов из одного и того же сообщения могут оказаться разными.

Вопрос о маршруте решается отдельно для каждого пакета. Всё зависит от того, куда его выгоднее передать в момент обработки. Если на каком-то участке Сети произошел «обрыв», то передача пакетов пойдет в обход этого участка. Таким образом, в любой момент времени по любому каналу Сети перемещается «вперемешку» множество пакетов из самых разных сообщений. Использование всякого канала связи стоит денег: междугородние, а тем более международные, телефонные разговоры достаточно дороги. Если бы, работая в Сети, вы в течение всего сеанса связи монопольно занимали международный канал, то расходы вас быстро разорили бы. Однако, согласно описанной технологии, канал вы делите с сотнями (а может — тысячами) других пользователей, и поэтому на вашу долю приходится лишь небольшая часть расходов.

16 Лекция 16. Поиск информации с использованием компьютера. Электронная почта

16.1 Поиск информации с использованием компьютера.

Windows предлагает несколько возможностей выполнения поиска файлов и папок. Средство «Помощник по поиску» обеспечивает наиболее быстрый способ поиска файлов. Рекомендуется использовать средство «Помощник по поиску», если выполняется поиск файлов одного из наиболее употребительных типов, если известно полное имя или часть имени файла или папки, поиск которых необходимо выполнить, или если известно время последнего изменения файла.

Если известна только часть имени, можно использовать подстановочные знаки для поиска всех файлов или папок, содержащих эту часть имени. Напри-

мер, по запросу «*письмо.*» будут найдены файлы «Поздравительное письмо.doc», «Специальное письмо.doc» и «Специальное письмо.txt».

Поиск информации - задача, которую человечество решает уже многие столетия. По мере роста объема информационных ресурсов, потенциально доступных одному человеку, были выработаны все более изощренные и совершенные поисковые средства и приемы, позволяющие найти необходимый документ. Обширные возможности для работы с большими массивами информации дают поисковые сервисы Internet.

При наличии первичных сведений по теме поиска, документы можно разыскивать в поисковых системах. При этом следует различать приемы простого, расширенного, контекстного и специального поиска.

Под простым поиском понимается поиск Web-ресурсов по одному или нескольким ключевым словам. Недостаток простого поиска заключается в том, что обычно он выдает слишком много документов, среди которых трудно выбрать наиболее подходящие. При использовании расширенного поиска ключевые слова связывают между собой операторами логических отношений. Расширенный поиск применяют в тех случаях, когда приемы простого поиска дают слишком много результатов. С помощью логических отношений поисковое задание формируют так, чтобы более точно детализировать задание и ограничить область отбора, например по дате публикации или типу данных. Контекстный поиск – это поиск по точной фразе. Он удобен для реферативного поиска информации, но доступен далеко не во всех поисковых системах. Прежде всего, чтобы обеспечивать такую возможность, система должна работать не только с индексированными файлами, но и с полноценными образами Web-страниц. Эта операция достаточно медленная, и ее выполняют не все поисковые системы. Специальный поиск применяют при розыске Web-страниц, содержащих ссылки на заданные адреса URL, содержащих заданные данные в служебных полях, например в поле заголовка и т.п.

Расширенный поиск. Кроме средства простого поиска обычно поисковые службы предоставляют средства расширенного поиска. Эти средства позволяют

более точно формулировать поисковое задание, но требуют определенного опыта и работают заметно медленнее. В большинстве поисковых систем команды расширенного поиска формируются с помощью логических команд. Удобство использования логических команд в частности связано с тем, что команды простого поиска у многих поисковых систем реализованы по-разному. Каждая система стремится сделать средства простого поиска наиболее удобными, а средства расширенного поиска – наиболее стандартными. Тем не менее, для обозначения логических операторов в различных поисковых системах используются разные обозначения. Поэтому желательно перед осуществлением расширенного поиска желательно изучить синтаксис поисковых запросов выбранной поисковой системы. Рассмотрим подробнее операторы логических отношений (логические команды).

Логическая оператор OR (ИЛИ) служит для формирования поискового запроса, если искомый текст должен содержать хотя бы один из терминов, соединенных данным оператором. Этот оператор в различных поисковых системах может обозначаться одним из следующих способов: | ; OR; ИЛИ. Например, результат запроса «Чёрное OR море» - будет представлен списком ссылок на документы, в которых есть слово «Чёрное», или слово «море», или оба этих слова вместе. В некоторых поисковых системах, как отмечалось выше, по умолчанию ключевые слова в запросе связаны именно этим логическим отношением. С помощью логического оператора AND (И) осуществляется поиск документов, содержащих все термины, соединенные данным оператором. Этот оператор может обозначаться одним из следующих способов: +; AND; &; И. Например, по запросу – «Чёрное AND море» - будут найдены документы, в которых содержатся слова «черное» и «море».

Логической оператор NOT (НЕ) позволяет производить поиск документов, в тексте которых отсутствуют термины, следующие за данным оператором. Этот оператор может обозначаться одним из следующих способов: not; !; ~; НЕ. Например, по запросу – «Чёрное NOT море», результат - документы, в которых есть слово «Чёрное» и нет слова «море».

С помощью логических операций можно создавать достаточно сложные запросы. Запрос из нескольких слов, перемежающихся операторами, будет истолкован в соответствии с их приоритетом. Операторы AND и NOT традиционно имеют более высокий приоритет, поэтому запрос из нескольких слов при обработке сначала группируется по операторам AND и NOT, и лишь потом по операторам OR. Например, по запросу «Чёрное AND море OR Крым» будут найдены документы, либо содержащие обязательно слова: «Чёрное» и «море», либо слово «Крым», либо все три слова.

Изменить порядок группировки можно использованием скобок. Оператор, стоящий в скобках, будет выполняться в первую очередь. Использование скобок позволяет строить вложенные запросы и передавать их операторам в качестве аргументов. Так по запросу «Чёрное AND (море OR Крым)» будут найдены документы, в которых обязательно содержится слово «Чёрное» и одно из двух слов «море» или «Крым». С помощью вложенных запросов можно значительно ограничивать область отбора, освобождая результирующий список от ненужных ссылок. Так, например, если нас интересует информация об отдыхе на юге на море, но исключительно на российском побережье, то можно попробовать использовать примерно такой запрос – «отдых AND ((Азовское OR Чёрное) AND море) NOT (Крым OR Турция OR Болгария)».

Чтобы найти файл или папку на компьютере нажмите кнопку *Пуск*, выберите пункты *Найти* и *Файлы и папки*. Выберите ссылку *Все файлы и папки*. Если ссылка *Все файлы и папки* не выведена на экране, возможно, был изменен способ поиска, используемый по умолчанию. Выберите ссылку *Изменить параметры*. Выберите ссылку *Изменить способ поиска в файлах и папках*. Нажмите кнопку *Стандартный*, а затем — кнопку *ОК*. Выберите ссылку *Все файлы и папки*. Введите часть имени или полное имя файла или папки или введите слово или фразу, содержащиеся в этом файле. Если об объекте поиска имеются определенные сведения или нужно уменьшить диапазон поиска, выберите один или несколько из следующих параметров. В поле *Поиск* выберите диск, папку или сетевой ресурс, в котором требуется

выполнить поиск. Нажмите кнопку *Дополнительные параметры* для задания дополнительных условий поиска. Нажмите кнопку *Найти*.

16.2 Использование подстановочных знаков

Подстановочный знак — это вводимый с клавиатуры знак, например, звездочка (*) или вопросительный знак (?), который можно использовать для представления одного или нескольких других знаков при поиске файлов, папок принтеров, компьютеров или людей. Подстановочные знаки часто используются вместо одного или нескольких знаков, когда нужный знак неизвестен либо для того, чтобы не вводить имя полностью.

Таблица 2 – Подстановочные знаки

Подстановочный знак	Использование
Звездочка (*)	<p>Звездочку можно использовать для замены любых знаков, включая пустой. Если при поиске файла не удастся вспомнить его имя полностью, но известно, что оно начинается на «gloss», введите следующее: <code>gloss*</code></p> <p>Будут найдены файлы всех типов, имена которых начинаются на «gloss», включая <code>Glossary.txt</code>, <code>Glossary.doc</code> и <code>Glossy.doc</code>. Чтобы задать поиск файла конкретного типа, введите следующее: <code>gloss*.doc</code></p> <p>Будет выполнен поиск всех файлов, имена которых начинаются на «gloss», с расширением <code>.doc</code>, например <code>Glossary.doc</code> и <code>Glossy.doc</code>.</p>
Вопросительный знак (?)	<p>Вопросительный знак можно использовать для замены одного знака в имени. Например, если ввести <code>gloss?.doc</code>, будет найден файл <code>Glossy.doc</code> или <code>Gloss1.doc</code>, но не <code>Glossary.doc</code>.</p>

Использование круглых скобок для управления порядком исполнения задания на поиск разрешается большинством крупнейших поисковых систем.

Таблица 3 – Расширенный поиск yandex.ru

Оператор	Описание
«	Обнаруживает точные слова в кавычках или фразы
	Найти любое из слов. Достаточно поставить между словами символ , и вы получите страницы, где содержится хоть одно из слов запроса.
~	Исключает страницы, содержащие слово или фразу.
()	Вы можете строить сколь угодно сложные конструкции, подставляя в каждом из операторов вместо отдельного слова целые выражения. Чтобы Яндекс при этом правильно понимал вас, заключайте выражения в круглые скобки.
&	Ограничить поиск страницами, где слова запроса находятся в пределах предложения
&&	Если вам нужны документы, где присутствуют заданные слова — неважно, на каком расстоянии и в каком порядке — соедините их оператором
!	Слова с большой и маленькой буквы считаются разными формами одного слова, поэтому все равно, какой регистр использовать в запросе. Исключением является оператор точной формы. Это полезно, если искомое имя собственное совпадает с распространенным словосочетанием, например, группа !Черный кофе. Все слова, которые вы приводите в запросе, по умолчанию ищутся с учетом морфологии. Чтобы отключить ее, используйте оператор ! перед словом (без пробела).
/	Вы можете указать максимально допустимое расстояние между двумя любыми словами запроса, поставив после первого слова символ /, сразу за которым идет число, означающее расстояние.
*	Замена части слова. Журналист*
?	Замена любого символа. Журналистик?

Таблица 4 – Расширенный поиск rambler.ru

Оператор	Описание
+	Используется для включения общих слов
«	Слова запроса, заключенного в двойные кавычки, ищутся в документах именно в том порядке и в тех формах, в которых они встретились в запросе. Таким образом, двойные кавычки можно использовать и просто для поиска слова в заданной форме (по умолчанию слова находятся во всех формах).
NOT	<p>Оператор NOT позволяет сформировать запрос, которому отвечают документы, удовлетворяющие левой части запроса и не удовлетворяющие правой. Так, результатом поиска по запросу собака NOT кошка будут все документы, в которых есть слово «собака» и нет слова «кошка».</p> <p>Это особенно полезно в случаях, если искомая словоформа является одновременно формой другого слова: Женя - имя собственное, а также деепричастие. Если мы ищем именно человека по имени Женя, в запросе можно написать Женя NOT женить.</p>
&&	Два запроса, соединенные оператором &&, образуют сложный запрос, которому удовлетворяют только те документы, которые одновременно удовлетворяют обоим этим запросам. Иными словами, по запросу собака && кошка найдутся только те документы, которые содержат и слово «собака», и слово «кошка».
	Сложному запросу, состоящему из двух запросов, соединенных оператором , удовлетворяют все документы, удовлетворяющие хотя бы одному из этих двух запросов. По запросу <i>собака кошка</i> найдутся документы, в которых есть хотя бы одно из двух слов - слово «собака» или слово «кошка» (или оба эти слова вместе).
()	Использование скобок позволяет строить вложенные запросы и передавать их операторам в качестве аргументов, а также перекрывать приоритеты операторов, принятые по умолчанию.

16.3 Организация специального поиска

С помощью средств специального поиска можно осуществлять поиск документов: содержащих искомые данные в различных полях (заголовке, ключевых словах, описании); расположенных на каком-либо сайте; содержащих те или иные ссылки и т.п. Рассмотрим подробнее некоторые возможности специального поиска. Многие поисковые системы позволяют разыскивать Web-документы по тексту, содержащемуся в заголовках. Поиск по заголовкам существенно уменьшает количество найденных ссылок, но очень точно выводит на нужные материалы. Ведь каждая Web-страница может иметь заголовок, если её автор не поленился его создать. И заголовок Web-страницы обычно точно характеризует тему материала, который содержится на ней. Например, если вам нужна информация о дистанционном обучении, то целесообразно искать страницы, в которых это сочетание присутствует в заголовке. Таким образом, в отличие от простого запроса мы отсекаем те документы, где эти слова не являются значимыми, т.е. не определяют тему статьи. Оператором или командой такого поиска является `title`, Этот оператор может обозначаться одним из следующих способов: `title:`, `t:`, `title =`, `$title` и т.п. После оператора следуют ключевые слова. В некоторых поисковых системах ключевые слова следует заключать в скобки, в других они пишутся без скобок. Например, в Индекс команда поиска в заголовке записывается так: `$title (очное обучение)`.

Поиск по сайту. С помощью поисковых систем можно осуществлять поиск информации не во всем Web пространстве, а на каком то конкретном сайте (если конечно последний проиндексирован поисковой системой). Соответствующий оператор может обозначаться следующим одним из следующих способов: `url=`, `url:`, `u:`, `#url=`«. Далее следует адрес Web-узла, некоторые системы требуют заключения адреса в кавычки. Если в запросе просто записать данный оператор с адресом какого-либо Web-узла, то будет получен список документов, проиндексированных поисковой системой на данном сайте. Но этот оператор можно комбинировать с другими, тем самым, осуществляя поиск информации

по всем правилам построения запросов на данном сайте. Поисковые системы могут предлагать другие возможности специального поиска: поиск по тексту ссылок, поиск в описании документа, поиск в списке ключевых слов Web-страниц, поиск по подписям к рисункам и т.п. Следует знать, что синтаксис поисковых запросов, да и состав доступных операторов, отличается в различных поисковых указателях. Поэтому перед осуществлением поиска в той или иной поисковой системе, следует изучить страницу справки по поиску в данной системе.

Таблица 5 – Специальный поиск aport.ru

Оператор	Описание
title=	Указанное после равенства слово или конструкция в круглых скобках должны искаться в заголовках документов (перед круглыми скобками знак равенства можно опускать). По запросу title=(папа или мама) будут найдены документы, содержащие в заголовке слово папа или слово мама, или оба слова одновременно.
anchor=	Указанное после равенства слово или конструкция в круглых скобках должны искаться в тексте ссылок. По запросу anchor=(ненавижу Интернет) будут найдены документы, в тексте ссылок на которые, встречаются оба слова: ненавидеть и Интернет.
text=	Указанное после равенства слово или конструкция в круглых скобках должны искаться только в обычном тексте. По умолчанию слова запроса ищутся как в тексте, так и во всех указанных выше полях. Чтобы искать только по тексту, выдаваемому в основное окно браузера, следует использовать данный оператор. По запросу text=(оглавление или содержание) будут найдены документы, в которых любое из указанных слов встречается в пределах основного текста документа.

16.4 Электронная почта

Электронная почта – одна из наиболее распространенных и популярных функций компьютерных сетей, обеспечивающая обмен сообщениями между пользователями сети. Порядок использования электронной почты во многом сходен с обычной почтой. Роль почтовых отделений играют узлы сети Интернет – почтовые серверы, на которых абонентам организуются специальные почтовые ящики. При пересылке сообщений по электронной почте необходимо указывать адрес получателя в сети Интернет. Он состоит из: имени пользователя, символа @, имени почтового сервера. Например: sasha_007@mail.ru

По электронной почте можно пересылать не только текстовые сообщения, но и готовые файлы, созданные в любых других программах. Работать с электронной почтой можно при помощи почтовой программы (почтового клиента), установленной на компьютере пользователя или при помощи браузера, с помощью web-интерфейса.

Почтовая программа (клиент электронной почты, почтовый клиент) — программное обеспечение, устанавливаемое на компьютере пользователя, предназначенное для получения, написания, отправки, хранения и обработки сообщений электронной почты пользователя (например, Microsoft Outlook Express). В системе пересылки электронной почты еще необходим почтовый сервер (сервер электронной почты). Почтовый сервер — это компьютерная программа, которая передаёт сообщения от одного компьютера к другому. Почтовые серверы работают на узловых компьютерах Интернета, а почтовые клиенты должны быть у каждого пользователя e-mail.

Существует большое количество WWW-серверов, которые предлагают завести бесплатный почтовый ящик и позволяют работать с почтой, используя только браузер. Чтобы получить бесплатный почтовый ящик на таком сервере, необходимо зарегистрироваться. Для этого нужно заполнить несколько обязательных полей – ввести свой логин, пароль, возраст, пол и т.д. В случае успешной регистрации, за Вами будет закреплен бесплатный почтовый электронный

адрес. Спам – рассылка коммерческой, политической и иной рекламы или иного вида сообщений лицам, не выразившим желания их получать. Старайтесь не рассылать одно письмо сразу большому количеству людей, т.к. многие могут воспринять это письмо как спам (нежелательную корреспонденцию). Спамер – пользователь, рассылающий спам по интернету, локальным сетям, системам сотовой связи, и т. д.

17 Лекция 17. Интернет как глобальная информационная система. Инструменты по созданию сайтов

17.1 Назначение Интернета

Главное назначение Интернета — быть глобальной информационной системой. Системой, дающей пользователю неограниченные возможности как для информационных коммуникаций с другими людьми, так и для получения любой интересующей его информации. Средства обеспечения определенных информационных услуг для пользователей Сети принято называть службами (сервисами) Интернета. Число различных служб в Сети непрерывно растет. Опишем лишь некоторые самые известные службы, разделив их на коммуникационные и информационные. Всякая услуга в Интернете предоставляется с помощью программ-серверов. Серверы делятся на несколько видов в зависимости от типа услуги, которая предоставляется пользователям: web-серверы предоставляют доступ к информации в виде web-страниц, файловые серверы обеспечивают доступ к файлам, почтовые серверы выполняют обмен почтовыми сообщениями, игровые серверы служат для одновременной игры нескольких пользователей и др.

С сервером взаимодействует программа-клиент, работающая на компьютере пользователя. Работа связки «клиент — сервер» подчиняется определенному протоколу: стандарту на представление, обработку, передачу информации средствами данной службы. Если протокол TCP/IP называется базовым прото-

колом Интернета, то протоколы служб можно назвать прикладными протоколами (иногда их называют протоколами второго уровня).

17.2 Коммуникационные службы Интернета

Коммуникационные службы обеспечивают общение между пользователями. Электронная почта — e-mail. Это наиболее старая и одна из самых массовых служб Сети. Ее назначение — поддержка обмена электронными письмами между пользователями. Схема работы электронной почты отражена на рис. 78. Почтовый сервер — это своеобразное «почтовое отделение», куда поступает входящая корреспонденция зарегистрированных на нем пользователей. Эта корреспонденция помещается в почтовые ящики пользователей — специально отведенные разделы на жестком диске. Каждый пользователь получает персональный почтовый адрес, по которому к нему будут поступать письма.

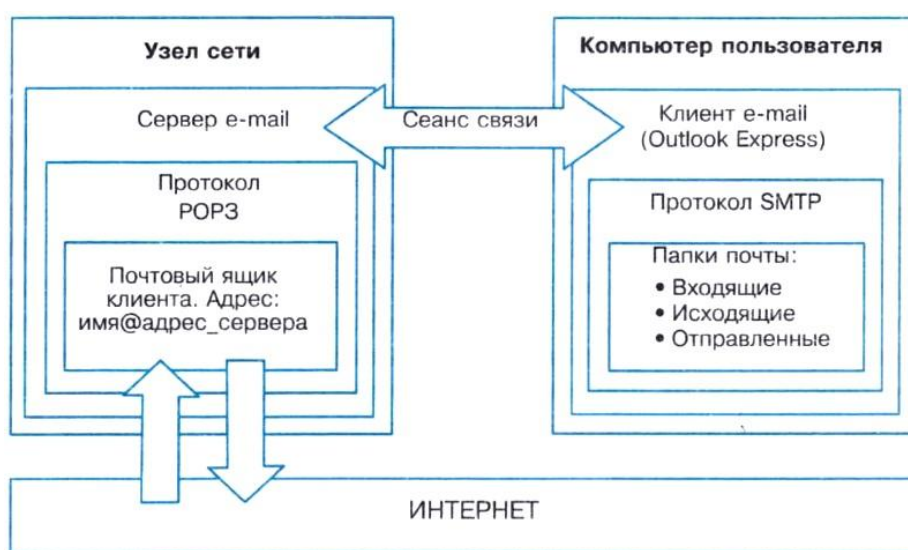


Рисунок 76 – Схема функционирования электронной почты

Каждый почтовый ящик имеет минимум один адрес следующего формата: имя@адрес_сервера. Адрес почтового ящика должен быть уникальным. Для работы с почтой можно использовать web-интерфейс (т. е. работать с обычным браузером) или установить специализированный почтовый клиент. Почтовый клиент — программа, помогающая составлять и посылать электронные сооб-

щения, а также получать и отображать письма на компьютере пользователя. Примеры почтовых клиентов: Outlook Express, Netscape Messenger, TheBat.

На рисунке 76 видно, что сервер и клиент работают по разным протоколам. Безусловно, «язык» у них общий и они «понимают» друг друга. Но их функции отличаются. POP3 (Post Office Protocol — протокол почтового отделения), кроме всего прочего, выполняет функцию защиты информации. Во время сеанса связи он устанавливает личность пользователя, обеспечивает связь с его персональным ящиком. Задача программы-клиента — передать на сервер исходящие письма и принять поступившие. Здесь используется более простой протокол SMTP (Simple Mail Transfer Protocol — простейший протокол передачи почты). По электронной почте можно не только отправлять текст, но и вкладывать в сообщение файлы любого формата.

Службы мгновенного обмена сообщениями (chat). Следующей популярной службой, предназначенной для мгновенного обмена сообщениями между пользователями Интернета, являются системы мгновенного обмена сообщениями. Это, например, программа ICQ («ай-си-кью» — игра слов, подобным образом читается выражение I SeekYou, что означает «я ищу вас»). ICQ позволяет пользователю получить уникальный номер, называемый UIN (Universal Internet Number, универсальный номер Интернета), используемый для вызова и прямого общения. ICQ-клиент можно бесплатно загрузить с сервера www.icq.com. После установки программы на компьютер необходимо зарегистрироваться в системе серверов ICQ и получить UIN. При каждом подключении к Интернету программа ICQ определяет текущий IP-адрес вашего компьютера и по нему определяет UIN. Зная UIN собеседника, можно быстро отправить ему сообщение. Сеть ICQ позволяет осуществлять поиск нужного абонента по целому ряду полей, включая фамилию, имя, адрес электронной почты, место проживания, возраст и т. д. Помимо службы ICQ, в Интернете есть несколько других аналогичных систем.

IP-телефония (Интернет-телефония) — система, позволяющая вести разговор в реальном времени с использованием каналов связи Интернета. Голосовая связь через IP-сеть может осуществляться двумя основными способами.

1) Компьютер — компьютер. Это самый первый и простой способ IP-телефонии. Для его реализации необходимы два компьютера со звуковыми картами, а также системами ввода/вывода звуковой информации (колонки или наушники, микрофон).

2) Компьютер — телефон. Один абонент должен иметь компьютер со звуковой картой и системой ввода/вывода звука, либо специальный IP-телефон, подключаемый к порту USB. Второй абонент имеет обычный телефон, находящийся в любой точке мира. Соединение осуществляется через специальные телефонные шлюзы, предоставляемые провайдерами телефонных услуг в Интернете. Звонок через телефонный шлюз стоит намного дешевле, чем обычный междугородний звонок по схеме «телефон — телефон». Одной из наиболее популярных программ для IP-телефонии в настоящее время является Skype. Она обеспечивает цифровую голосовую связь через Интернет между компьютерами, а также платные услуги для связи с абонентами обычной телефонной сети. При подключении к компьютеру web-камеры Skype поддерживает также и видеосвязь между двумя абонентами. Видеоконференция — это информационная услуга, обеспечивающая одновременную двухстороннюю передачу, обработку, преобразование и представление видео- и звуковой информации на расстояние в режиме реального времени с использованием компьютеров для более чем двух абонентов. Для общения в режиме видеоконференции абоненты должны иметь микрофон, видеокамеру (web-камеру), устройство отображения информации и воспроизведения звука, а также компьютер с необходимым программным обеспечением. Популярной системой поддержки видеоконференций является Vidicor. Различают конференции симметричные, т. е. такие, где все абоненты видят и слышат друг друга, и асимметричные, т. е. такие, где часть участников только видит и слышит, но не участвует постоянно. С помощью видео-

конференций организуются совещания, встречи людей, находящихся в разных точках мира, дистанционное обучение и др.

17.3 Информационные службы Интернета

Информационные службы предоставляют пользователям возможность доступа к определенным информационным ресурсам, хранящимся в Интернете. Такими ресурсами являются либо файлы стандартных форматов, либо разного рода документы (в том числе мультимедийные), которые можно просмотреть, сохранить, распечатать.

Служба передачи файлов. Часто эту службу называют по имени используемого протокола: FTP (File Transfer Protocol — протокол передачи файлов). Со стороны Сети работу службы обеспечивают FTP-серверы, а со стороны пользователей — FTP-клиенты. Назначение FTP-сервера — хранение набора файлов самого разнообразного назначения (обычно в архивированном виде). Чаще всего это программные файлы: средства системного и прикладного программного обеспечения. Но в наборах могут храниться файлы и любых других форматов: графические, звуковые, документы Microsoft Word, Microsoft Excel и др. Вся эта информация образует иерархическую структуру папок (каталогов и подкаталогов). После соединения FTP-клиента с сервером на экране пользователя открывается файловый интерфейс хранилища папок и файлов на сервере (наподобие Проводника Windows). Далее работа происходит так же, как с файловой системой на собственном ПК: папки и файлы можно просматривать, сортировать, копировать на свои диски. Клиент FTP входит в состав программы Internet Explorer и поэтому всегда имеется на ПК, работающем под управлением ОС Microsoft Windows.

World Wide Web (WWW, Всемирная паутина) — самая массовая сегодня информационная служба Интернета. Это огромная, распределенная по всему миру информационная система, содержащая миллионы документов на самые разнообразные темы. Работает эта служба на базе протокола HTTP. О популярности WWW говорят такие данные: с момента создания Интернета (1969 г.) до

появления WWW (1993 г.) к услугам Сети подключились около 2 миллионов пользователей; с появлением WWW за 5–7 лет это число увеличилось приблизительно до 200 миллионов человек. В последнее время Интернет стал отождествляться с WWW. В настоящее время в мире насчитывается более 2 миллиардов пользователей Интернета и World Wide Web.

17.4 Web-2-сервисы

Начиная с 2005 года в Интернете развивается новая методика использования web-технологии, объединяющая в себе информационные и коммуникационные функции, которая получила название Web-2. Основная ее особенность состоит в том, что информационный контент, выкладываемый в Сеть, создают сами пользователи. К сервисам, существующим на основе этой методики, относятся социальные сети (Одноклассники, Вконтакте и др.), блоги, живые журналы, видеохостинги (для размещения видеороликов, например YouTube), фотохостинги (для размещения фотоальбомов), файловые обменники и т. д. Блог (от weblog — интернет-дневник) — web-сайт, основное содержание которого составляют регулярно добавляемые записи владельца (блоггера). Обычно записи отсортированы в обратном хронологическом порядке, т. е. вначале располагается последняя запись. Блоги доступны любым читателям и допускают публичную полемику с блоггером. Живой журнал (ЖЖ, LiveJournal) — исторически первый, созданный в 1999 г., сервис Интернета для ведения блогов. ЖЖ поддерживает общение между блоггерами, позволяет вести коллективные блоги, пополнять «ленту друзей» и другие. Живой журнал — один из вариантов социальных сетей, существующих в Интернете. Наряду с ЖЖ существуют другие платформы ведения блогов.

17.5 Инструменты по созданию сайтов

Создание Web-страниц непосредственно на HTML требует больших усилий, времени и досконального знания синтаксиса языка. Применение

специальных инструментальных программных средств (HTML-редакторов) делает работу по созданию Web-сайтов простой и эффективной.

Инструментами называют все, что помогает в разработке проекта. Таким образом, инструментом может называться даже простейший блокнот, в котором вы можете писать код. Инструмент 1: редактор кода. Прежде всего, это полезный инструмент для тех, кто собирается самостоятельно создавать свой ресурс или вносить какие-то правки в него. Если это так, то вам понадобится написание кода на языке html, как минимум. Для удобной работы с ними нужно какое-то программное обеспечение, в нашем случае редактор, в котором бы этот код подсвечивался, а также появлялись различные подсказки по написанию. В любом случае, вы можете их убрать, если являетесь уже опытным разработчиком. Из таких редакторов можно выделить Notepad++, Brackets, Sublime Text и другие. У каждого редактора есть свои плюсы.

```
115  h3{
116      margin: 0;
117  }
118  .hidden-link{
119      cursor: pointer;
120      color: green;
121      text-decoration: underline;
122      font-weight: bold;
123      font-size: 24px;
124  }
125  img{
126      text-align: center;
127  }
```

Рисунок 77 – Окно редактора

Инструмент 2: среда разработки. Это что-то вроде редактора, но включающее в себя гораздо больше возможностей. Само название говорит об этом. В такой среде есть все необходимое для того, чтобы разработать сайт. Обычно такую программу устанавливают для того, чтобы работать с различными языками программирования. В web-программировании чаще всего используется PHP, именно для работы с ним ставят среду. Например, NetBeans. Это бесплатная программа в различных своих сборках может обеспечить разработчика всем необходимым.

Инструмент 3: PhotoShop или любой другой редактор, поддерживающий работу со слоями. Реализация интернет-проекта начинается с того, что дизайнер

рисует его в программе, вроде Photoshop, для того, чтобы верстальщику не приходилось все придумывать с нуля. Поэтому такой графический редактор, несомненно, тоже является одним из инструментов.

Инструмент 4: локальный сервер. Пожалуй, один из важнейших инструментов, который сильно помогает в создании сайта. Самый известный локальный сервер на Windows – это Denwer. Также есть OpenServer. Благодаря локальному серверу на компьютере вы сможете открывать php-файлы и видеть результат их работы, вы даже можете установить на сайт абсолютно любой движок. Что захотите: WordPress, Joomla, Opencart, modx и проводить эксперименты.

Инструмент 5: движки. Это тоже инструменты для создания сайта, причем бесплатные, в большинстве своем, хотя есть и платные cms. С помощью движков вы можете получить красивый сайт просто так, не потратив ни копейки. Дело в том, что для того же WordPress очень много классных бесплатных шаблонов, так что вы можете получить профессиональный адаптивный шаблон за так, а если купите премиум-тему, то получите еще больше возможностей. Впрочем, сегодня в премиум-шаблоны встроены страницы, на которых вы можете изменить дизайн с помощью визуальных настроек, без всякого вмешательства в код.

В список инструментов по созданию сайтов вполне можно занести фреймворки и различные библиотеки для упрощения работы с кодом. Существуют мощные инструментальные системы разработки сайтов, например Microsoft FrontPage или Macromedia Dreamweaver. Для разработки отдельных Web-страниц можно использовать свободно распространяемые редакторы FrontPage Express (входит в состав Microsoft Internet Explorer) и Netscape Composer (входит в состав Netscape Communicator). Например, использование Netscape Composer для создания и редактирования Web-страниц без программирования на HTML позволяет форматировать текст, создавать различного типа списки, помещать на страницу графические изображения и таблицы, вставлять гиперссылки и так далее. Процесс создания и

редактирования страницы очень нагляден, так как производится в режиме WYSIWYG («What You See Is What You Get» - «Что видишь, то и получишь»).

Упрощенные инструменты: конструкторы сайтов. Конструктор позволяет пойти гораздо более простым путем. При работе в такой программе вам не понадобится никаких начальных знаний в области сайтостроения (хотя все-таки полезно их иметь), все настройки выполняются в визуальном режиме. Вы можете выбрать один из сотен шаблонов, и ваш сайт будет смотреться хорошо, хотя и не будет обладать уникальным оформлением. Некоторые конструкторы являются крупными web-площадками, где вы можете также зарегистрировать себе домен, заказать дополнительные услуги и т.д. Например, Wix, Ucoz. Эти площадки работают на собственных движках. Есть просто программы-конструкторы. Например, DreamViewer или полностью визуальный Adobe Muse. Их называют визуальными или полувизуальными редакторами, где часть кода вы можете писать сами, а часть генерировать с помощью разных кнопок и настроек. Выглядит это примерно так:

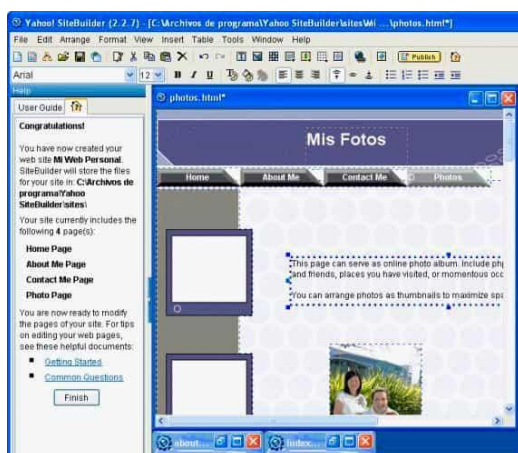


Рисунок 78 – Программа-конструктор

Отличие от web-сервисов в том, что в них вы все-таки выполняете хоть какую-то работу, а также вы получаете в конце лишь файлы будущего сайта, но для реального появления сайта в сети необходимо выполнить все те же операции: зарегистрировать доменное имя, оплатить место на удаленном сервере и разместить нужные файлы. Web-конструкторы берут все это на себя, максимально упрощая для вас дело. Там вы можете создать себе сайт даже полностью бесплатно, но тогда ваши возможности сильно ограничены.

17.6 Основы создания Web-страниц средствами языка разметки гипертекста HTML

Создание простого документа. Основой разработки веб-страниц является язык HTML (Hyper Text Markup Language – язык разметки гипертекстов).

Основной структурной единицей языка HTML является тег (от англ. Tag - дескриптор, маркер). Тег всегда заключен между скобками $\langle \rangle$ и имеет следующий вид: $\langle \text{ТЕГ атрибут 1=ЗНАЧЕНИЕ ... атрибут N=ЗНАЧЕНИЕ} \rangle$

Теги бывают одиночными и контейнерными. Контейнером называется пара: открывающий $\langle \text{ТЕГ} \rangle$ и закрывающий $\langle / \text{ТЕГ} \rangle$. Открывающий тег служит для указания программе-браузеру начала какого-либо объекта или задания свойств объектов помещенных в контейнер. Закрывающий тег служит для указания программе-браузеру о конце объекта или окончания применения свойств, заданных в открывающем теге. Атрибуты тега задают значения свойств данного объекта или объектов помещенных в контейнер. Документ HTML начинается открывающим тегом $\langle \text{HTML} \rangle$ и заканчивается закрывающим тегом $\langle / \text{HTML} \rangle$. Между данной парой контейнерных тегов располагаются две другие основные части HTML документа: заголовок, заключенный в контейнер $\langle \text{HEAD} \rangle \dots \langle / \text{HEAD} \rangle$ и тело документа в контейнере $\langle \text{BODY} \rangle \dots \langle / \text{BODY} \rangle$ (содержание страницы). В разделе описания заголовка можно указать заглавие документа, для этого используется тэг $\langle \text{TITLE} \rangle \dots \langle / \text{TITLE} \rangle$ (имя страницы). Таким образом, структура простого HTML документа выглядит примерно так:

```
 $\langle \text{HTML} \rangle$  {начало страницы}  
 $\langle \text{HEAD} \rangle$  {описание страницы, заголовка}  
 $\langle \text{TITLE} \rangle$  название  $\langle / \text{TITLE} \rangle$  {имя страницы}  
 $\langle / \text{HEAD} \rangle$  {закрытый тэг описания заголовка}  
 $\langle \text{BODY} \rangle$  {содержание страницы}  
текст  
 $\langle / \text{BODY} \rangle$  {закрытый тэг описания страницы}  
 $\langle / \text{HTML} \rangle$  {конец страницы}
```

Форматирование текста.

`<BODY></BODY>` - тег, определяющий границы тела документа.

Атрибуты: `BGCOLOR` – определяет цвет фона документа. По умолчанию «White» (`#FFFFFF`).

`TEXT` – задает цвет текста для всей страницы. Цвет указывается в формате RGB или константами `red`, `green`, `blue` и т.п. По умолчанию «black» (`#000000`).

В таблице 6 приведены названия цветов, определенные в стандарте HTML 4 и соответствующие им RGB-коды. Отметим, что многие современные браузеры выходят за рамки стандартов и поддерживают гораздо больше названий цветов.

Таблица 6 – Названия цветов

Название	Код	Название	Код
Aqua	<code>#00FFFF</code>	navy	<code>#000080</code>
Black	<code>#000000</code>	olive	<code>#808000</code>
Blue	<code>#0000FF</code>	purple	<code>#800080</code>
Fuchsia	<code>#FF00FF</code>	red	<code>#FF0000</code>
Gray	<code>#808080</code>	silver	<code>#C0C0C0</code>
Green	<code>#008000</code>	teal	<code>#008080</code>
Lime	<code>#00FF00</code>	white	<code>#FFFFFF</code>
Maroon	<code>#800000</code>	yellow	<code>#FFFF00</code>

`<P></P>` - абзац. Атрибуты: `ALIGN` – выравнивание. Возможные значения: `RIGHT`-выравнивание текста по правому краю; `CENTER` - по центру; `LEFT` - по левому краю; `JUSTIFY` – по ширине.

`<H1></H1>...<H6></H6>` - заголовки разного уровня (от первого до шестого). Пример: `<H2 ALIGN = CENTER>` Заголовок второго уровня с выравниванием по центру.

`<HR>` - горизонтальная линия. Атрибуты: `ALIGN` – выравнивание.

`WIDTH` – длина линии в процентах от окна браузера или пикселях

SIZE – ширина линии в процентах от окна браузера или пикселях

COLOR – цвет линии.

Пример:<HR ALIGN = CENTER WIDTH=50% SIZE=6 COLOR= RED>

Дополнительные возможности по форматированию:

 - полужирный текст;

<I></I> - курсивный текст;

<U></U> - эффект подчёркивания;

<STRIKE></ STRIKE >- эффект зачеркивания.

 - используется для указания начертания шрифта в документе.

Атрибуты: SIZE – размер шрифта, COLOR – цвет шрифта, FACE – гарнитура шрифта.

Пример:

<P> Красная строка набрана шрифтом ARIAL 4 размера </P>

Создание списков. В HTML-документе существует четыре основных вида списков: нумерованный, маркированный, вложенные списки, список определений. Фрагмент текста, представляющий список, заключается в тэги:

 упорядоченный список (orderedlist);

 неупорядоченный список (unorderedlist);

< LI>.... вложенные списки;<DL>....</DL> список определений.

Список определений служит для создание списков типа «термин» - «описание». Каждый термин начинается тэгом <DT> , а описание - тэгом <DD>.Каждый элемент списка заключается в тэги (от английского listitem). При выводе на экран элементы списка имеют отступ и начинаются с новой строки.

Тэг может иметь параметры: <OL TYPE=A|a|I|i|1 START=n>

TYPE= определяет вид нумерации, START= задаёт начальное значение первого элемента списка (независимо от типа указывается цифрой).

TYPE=A – маркеры в виде прописных латинских букв;

TYPE=a – маркеры в виде строчных латинских букв;

TYPE=I – маркеры в виде больших римских цифр;

TYPE=i – маркеры в виде маленьких римских цифр;

TYPE=1 – маркеры в виде арабских цифр (по умолчанию).

Тэг может иметь параметр:<ULTYPE=disc|circle|square>

Тип тэга определяет внешний вид маркера как вид по умолчанию (disc), круглый (circle) или квадратный (square).

Тэг может иметь параметры:<OLTYPE=disc|circle|square>

или <OL TYPE=A|a|I|i|1 VALUE=n>

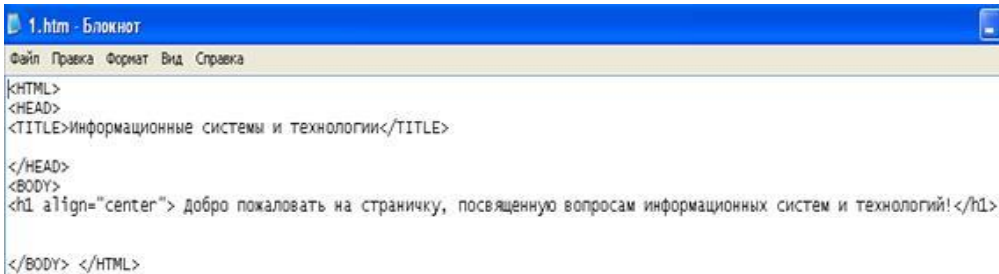
TYPE Вид маркера (см.) или счетчика (см. OL)

VALUE=n Значение для элемента пронумерованного списка (его номер). Все дальнейшие номера элементов списка будут отсчитываться от этого номера.

17.7 Создание документа HTML в текстовом редакторе Блокнот

Для того чтобы создать документ HTML в текстовом редакторе Блокнот необходимо выполнить следующие действия:

1. Запустить текстовый редактор Блокнот (Пуск > Программы > Стандартные > Блокнот).
2. Создать типовую структуру HTML - документа и ввести текст.



```
1.htm - Блокнот
Файл  Правка  Формат  Вид  Справка
<HTML>
<HEAD>
<TITLE>информационные системы и технологии</TITLE>
</HEAD>
<BODY>
<h1 align="center"> Добро пожаловать на страничку, посвященную вопросам информационных систем и технологий!</h1>
</BODY> </HTML>
```

Рисунок 79 – Создание HTML - документа в текстовом редакторе Блокнот

3. Сохранить этот документ. При сохранении в качестве типа файла укажите «все файлы», в качестве имени файла first.html.
4. Закрыть блокнот. Найти файл first.html и открыть его с помощью интернет-браузера.

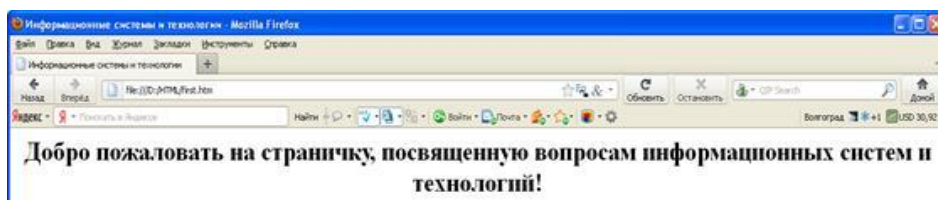


Рисунок 80– Иллюстрация документа в интернет-браузере.

Публикация сайта. Прежде чем разместить свой Web-сайт на сервере в Интернете, его необходимо тщательно протестировать, так как потенциальными посетителями вашего сайта являются десятки миллионов пользователей Интернета. Необходимо просмотреть, как выглядят ваши страницы в наиболее распространенных браузерах. Необходимо проверить:

- нормально ли читается текст на выбранном фоне;
- правильно ли расположены рисунки;
- правильно ли работают гиперссылки.

Для публикации Web-сайта необходимо найти подходящее место на одном из серверов Интернета. Многие провайдеры предоставляют своим клиентам возможность бесплатного размещения Web-сайтов на своих серверах (бесплатный хостинг). Для публикации Web-сайта необходимо получить от провайдера необходимые сведения:

- URL -адрес сайта;
- секретные имя пользователя и пароль, которые необходимы администратору сайта для его редактирования.

Технология публикации Web-сайтов может иметь несколько вариантов:

- для публикации можно воспользоваться инструментальным средством, которое использовалось для создания Web-сайта (например, Netscape Composer);
- организация, предоставившая место на своем сервере, может предложить публиковать сайты через Web-интерфейс с помощью Браузера или с помощью оригинального Диспетчера файлов;
- наибольшие возможности при публикации предоставляют FTP-клиенты.

Список использованных источников

- 1 Семакин, И.Г. Информатика. Базовый уровень: учебник для 10 класса / И.Г. Семакин, Е.К. Хеннер, Т.Ю. Шеина. – 4-е изд. – М.: БИНОМ. Лаборатория знаний, 2015. – 264с.: ил.
- 2 Персональный сайт Иванова А.М. Информатика и информационно-коммуникационные технологии. – Режим доступа: <http://xn----7sbbfb7a7aej.xn--p1ai/informatika/informatika.html>, 2011 – 2018.
- 3 Семакин, И.Г. Информатика и ИКТ. Задачник-практикум. ч. 1/ под ред. И. Г. Семакина, Е. К. Хеннера, М. : БИНОМ. Лаборатория знаний, 2014. – 312 с.: ил.
- 4 Бухтоярова, А.А. Информатика: методические указания к лабораторным работам. в 2ч. / А.А. Бухтоярова, С.С. Цыганкова; Оренбургский гос. ун-т. – Оренбург : ОГУ, 2011. –60с.
- 5 Работа в Microsoft Word 2007. Авторский курс Олега Спиридонова / [Электронный ресурс] / ИНТУИТ национальный открытый университет. – Режим доступа: <https://www.intuit.ru/studies/courses/1134/222/info>
- 6 Работа в Microsoft Excel 2007. Авторский курс Олега Спиридонова / [Электронный ресурс] / ИНТУИТ национальный открытый университет. – Режим доступа: <https://www.intuit.ru/studies/courses/1128/226/info>
- 7 Microsoft Access 2007. Авторский курс лекций Владимира Ткаченко. / [Электронный ресурс]. Режим доступа: // <http://www.lessons-tva.info/edu/inf-access/access.html> – Обучение в интернет, 2007 - 2018.