

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Т. Е. Родионова

**ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ
ОБРАБОТКИ ДАННЫХ**

**Учебное пособие
для студентов направления 01.03.04**

Ульяновск
УлГТУ
2020

УДК 004.6(075.8)
ББК 32.973.26-018.2я73
Р60

Рецензенты:

Заведующий кафедрой «Механика и теория управления» УлГУ,
доктор физико-математических наук, профессор А. С. Андреев;
Заведующий кафедрой «Телекоммуникационные технологии и
системы» УлГУ, доктор технических наук, профессор А. А. Смагин

Утверждено редакционно-издательским советом университета
в качестве учебного пособия

Родионова, Т. Е.

Р 60 Информационные технологии обработки данных: учебное пособие для
студентов направления 01.03.04 / Т. Е. Родионова. – Ульяновск :
УлГТУ, 2020. – 113 с.

ISBN 978-5-9795-2017-9

Пособие предназначено для подготовки студентов направления 01.03.04, по дисциплине «Информационные технологии». Составлено в соответствии с учебным планом специальности. Разработано на кафедре прикладной математики и информатики. В пособии излагается история развития информационных технологий, состав персонального компьютера, классификация программного обеспечения и состав систем программирования. Рассмотрены вопросы, касающиеся систем счисления, используемых в компьютерных вычислениях. В учебном пособии приведено большое количество примеров для иллюстрации излагаемого материала. Кроме того, пособие содержит задание на лабораторный практикум по изучаемой дисциплине. Пособие может быть использовано для самостоятельного изучения студентами основ информационных технологий.

Предназначено для студентов вузов дневной формы обучения.

УДК 004.6(075.8)
ББК 32.973.26-018.2я73

ISBN 978-5-9795-2017-9

© Родионова Т. Е., 2020
© Оформление. УлГТУ, 2020

ОГЛАВЛЕНИЕ

Введение	4
Эволюция информационных технологий	5
Двоичное кодирование информации.....	7
Эволюция средств вычислительной техники	12
Счетные устройства	12
Поколения компьютеров	15
Системы счисления	25
Перевод чисел из одной системы счисления в другую	25
Способы описания алгоритма	34
Аппаратное обеспечение компьютера	43
Базовая конфигурация ПК.....	44
Видеоподсистема компьютера.....	50
Классификация программного обеспечения	51
Лабораторный практикум.....	60
Лабораторная работа 1. Работа с системами счисления	60
Лабораторная работа 2. Дополнительный код числа	60
Лабораторная работа 3. Ввод и редактирование в текстовом процессоре Writer (OpenOffice.org).....	61
Лабораторная работа 4. Встраивание объектов в документ	70
Лабораторная работа 5. Подготовка документа к печати.....	76
Лабораторная работа 6. Построение граф-схем циклических процессов	80
Лабораторная работа 7. Основы работы в электронных таблицах.	83
Лабораторная работа 8. Расчеты с использованием условий	92
Лабораторная работа 9. Работа со списками.	104
Список литературы	109
ПРИЛОЖЕНИЕ 1	111
ПРИЛОЖЕНИЕ 2	112

ВВЕДЕНИЕ

Информационные технологии (ИТ) являются составной частью информатики. Средствами ИТ информация структурируется и формируется в виде знаний. Информационная технология базируется на реализации базовых технологических процессов извлечения, передачи, обработки, хранения, представления и использования информации. Основу информационной индустрии составляют базовые информационные технологии, которые строятся на базовых технологических операциях, но кроме этого включают ряд специфических моделей и инструментальных средств [5].

Таким образом, информационные технологии рассматриваются как система, включающая базовые технологические процессы, базовые и специализированные информационные технологии, инструментальную базу.

Информационная технология – совокупность методов и способов получения, обработки, представления информации, направленных на изменение ее состояния, свойств, формы, содержания и осуществляемых в интересах пользователей. Кроме того, информационная технология – это процесс, включающий совокупность способов сбора, хранения, обработки и передачи информации на основе применения средств вычислительной техники.

Цель применения информационной технологии – производство информации для ее анализа человеком и принятия на его основе решения по выполнению какого-либо действия и снижение трудоемкости использования информационных ресурсов.

В настоящее время выделяется несколько основных информационных революций в истории развития человеческого общества. Они связываются со следующими событиями: появление человеческой речи; изобретение письменности; изобретение книгопечатания; изобретение радио, телефона, телевидения; изобретение микропроцессорных технологий и появление персональных компьютеров; создание компьютерных сетей [10, 13].

ЭВОЛЮЦИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

В управлении данными, объединяющем задачи их получения, хранения, обработки, анализа и визуализации, выделяют **шесть поколений**:

1. До изобретения приспособлений и устройств для облегчения счета все операции над числами выполнялись вручную.

2. После изобретения счетных устройств (механические калькуляторы, выполняющие основные арифметические действия, первые электрические вычислительные машины) для ввода и хранения информации использовались перфокарты.

3. В третьем поколении данные хранились на магнитных лентах, и применялась пакетная обработка информации, хранящейся в виде последовательных файлов.

4. Четвертое поколение связано с появлением понятия базы данных и развитием программного обеспечения для оперативного доступа к ее записям.

5. В пятом поколении был осуществлен переход к реляционным базам данных и использованием для хранения больших объемов информации распределенной и клиент-серверной обработки.

6. Современное состояние развития информационных технологий считают шестым поколением. Оно характеризуется развитыми технологиями хранения и обработки различных типов данных (документы, графические, звуковые и видеoinформация). Это позволяет создавать и применять средства хранения для приложений Интернета и Интранета.

В первом поколении развития ИТ (4000 г. до н. э. – 1900 г.) человечество использовало для хранения информации глиняные пластинки, затем появились такие носители как папирус, береста и, наконец, бумага. Это поколение характеризуется разработкой алфавитов для различных языков, появлением книг, библиотек и печатных изданий. Это были большие достижения, но обработка информации в эту эпоху осуществлялась вручную.

Второе поколение (1900–1955 гг.) характеризуется активным использованием перфокарт и перфолент для записи данных. Вся хранимая информация кодировалась двоичными кодами. Активно разрабатывалось оборудование для записи и чтения информации с этих носителей. Одним из лидеров в производстве электромеханического оборудования для перфокарт является компания IBM.

Третье поколение характеризуется появлением магнитных носителей информации – магнитных лент и магнитных дисков. Хотя их объем по современным меркам был небольшой, каждый из них мог заменить десятки тысяч перфокарт. Распространение компьютеров и расширение области их применения подтолкнуло к развитию программного обеспечения для обработки данных. Появились компиляторы с языков программирования высокого уровня. Они позволяли создавать программы для пакетной обработки данных. Вместе с развитием систем программирования выявлялись и их ограничения, и недостатки, которые требовали от разработчиков новых идей в создании программного обеспечения и языков программирования.

Четвертое поколение характеризуется широким применением оперативных баз данных и доступом к информации в интерактивном режиме.

Пятое поколение характеризуется доминированием реляционных баз данных, использованием для их реализации «клиент – серверной» архитектуры. Приложение для доступа к базе данных состоит из двух частей: клиентская часть содержит средства для ввода данных и формирования запроса к базе данных; серверное программное обеспечение осуществляет хранение данных и реализацию клиентских запросов. В данной структуре используется параллельная обработка запросов к базе данных от нескольких пользователей.

Примером **современной базы данных** является всемирная библиотека. Многие ведомственные библиотеки открывают доступ к своим хранилищам в режиме on-line. В этом же режиме публикуется новая научная литература.

ДВОИЧНОЕ КОДИРОВАНИЕ ИНФОРМАЦИИ

Человечество занимается обработкой информации уже несколько тысяч лет. Разрабатываются устройства, позволяющие облегчить этот процесс, избавив нас от рутинных операций. Как уже говорилось ранее, вначале вся обработка данных велась вручную. Сейчас мы активно используем вычислительную технику и разработанное программное обеспечение [11, 14].

Для автоматической обработки данных очень важно унифицировать форму их представления, т.е. использовать системы кодирования информации. Вообще, термин «кодирование» означает представление данных одного типа через данные другого типа. Например, в памяти компьютер вся текстовая информация хранится в числовом виде.

В вычислительной технике для представления данных используется двоичное кодирование. Оно основано на представлении данных с помощью двоичной системы счисления (последовательность двух знаков: 0 и 1). Эти знаки называются двоичными цифрами (binary digit). Для хранения информации в вычислительной технике используется двоичное кодирование, это связано с тем, что современные устройства памяти имеют только два устойчивых состояния и соответственно могут хранить и распознавать не более двух знаков (цифр 0 и 1).

Примеры:

- электромагнитные реле (замкнуто/разомкнуто), широко использовались в конструкциях первых ЭВМ;
- участок поверхности магнитного носителя информации (намагничен/размагничен);
- участок поверхности лазерного диска (отражает/не отражает);
- триггер может устойчиво находиться в одном из двух состояний и широко используется в оперативной памяти компьютера.

Кодирование чисел. Для кодирования целых чисел осуществляется перевод их в двоичную систему счисления. Для представления чисел с дробной частью (действительные значения) используется запись длиной 80 двоичных разрядов. Исходное значение преобразуется в специальную

форму (экспоненциальная форма записи числа). Рассмотрим примеры преобразования чисел в экспоненциальную форму:

$$77,159812 = 0,77159812 * 10^2$$

$$512 = 0,512 * 10^3$$

$$0,0090876 = 0,90876 * 10^{-2}$$

Часть числа, которая располагается за знаком «0,» называется *мантисса*, она содержит значащие цифры числа без ведущих нулей. Степень числа десять называется *порядок*, он отвечает за положение десятичной точки в исходном значении. Мантисса числа вместе со знаком («+» или «-») размещается слева и занимает большую часть отводимых для хранения разрядов, затем записывается порядок числа. Положительное число в знаковом разряде содержит 0 (знак «+»), отрицательное число – 1 (знак «-»).

Кодирование текстовых данных. Для записи в память компьютера вся текстовая информация преобразуется в числовую. Существуют специальные кодовые таблицы, где каждому символу текста поставлен в соответствие двоичный код. Для удобства работы пользователей в литературе приведены не только двоичные коды символов, но и их значения в десятичной, восьмеричной и шестнадцатеричной системах счисления. Эти таблицы использует операционная система для обработки информации. Большинство кодировочных таблиц используют коды длиной восемь разрядов, с их помощью можно хранить значения для $2^8=256$ различных символов текста. Развитие вычислительной техники было связано с появлением различных кодовых таблиц для разных компьютеров и отсутствием единых стандартов. В настоящее время существует несколько наиболее распространенных кодовых таблиц, которые активно используются для кодирования и хранения информации.

Международным стандартом сейчас является кодовая таблица *ASCII* (American Standard Code for Information Interchange – стандартный код информационного обмена США), которая введена институтом стандартизации США (*ANSI – American National Standard Institute*). Кодовая таблица *ASCII* состоит из двух частей: базовой и дополнительной. Базовая таблица содержит управляющие коды, символы английского

алфавита, цифры (от 0 до 9) и другие наиболее часто используемые символы с кодами от 0 до 127. Дополнительная часть таблицы (коды от 128 до 255) содержит графические символы и национальные алфавиты.

Сейчас американский код *ASCII* является международным стандартом, и национальные системы кодирования содержатся в расширенной части таблицы со значениями кодов от 128 по 255.

Сегодня самые распространенные кодовые таблицы:

- ASCII – способ печати специальных знаков, уникальные коды которых представлены цифрами. Это самый распространенный тип кодировки. Он был разработан в 1963 году в США;

- Windows-1251 – стандартная восьмибитная кодировочная таблица для русскоязычной версии операционной системы Windows;

- Unicode. Это 16-битная кодировочная таблица для современных операционных систем. Используется современными пользователями наравне с ASCII.

Для кодирования текстов на русском языке (то есть букв кириллицы) наиболее широко применяются следующие кодовые страницы:

- Windows-1251, она же Microsoft codepage 1251 (CP1251) – в системах Windows;

- Семейство кодовых страниц KOI8;

- Альтернативная кодировка (IBM code page 866) – в системах DOS, а также в текстовых окнах Microsoft Windows;

- MacCyrillic – на компьютерах Macintosh.

Кодирование графических данных. В зависимости от способа формирования изображения компьютерную графику подразделяют на растровую и векторную. Трехмерная (3D) графика считается отдельным типом, в ней сочетаются векторный и растровый способ формирования изображения.

Растровая графика. Графическая информация на экране монитора представляется в виде растрового изображения, которое формируется из определенного количества строк, которые в свою очередь содержат определенное количество точек (пикселей). Поскольку количество точек в

изображении и их координаты известны, то в памяти компьютера хранится информация о каждой точке последовательно. Свойства каждой точки изображения (яркость) задается целым числом. Таблица кодов используемых оттенков формируется при выборе цветовой палитры. Таким образом, растровое изображение может быть записано в виде последовательности числовых кодов. Одним из недостатков растровой графики является то, что при увеличении масштаба изображения становятся видны элементы раstra, которые искажают картинку.

Качество изображения определяется разрешающей способностью монитора, т.е. количеством точек, из которых оно складывается. Чем больше разрешающая способность, то есть чем больше количество строк раstra и точек в строке, тем выше качество изображения. В современных персональных компьютерах обычно используется разрешающая способность экрана: 1280×1024 точки.

Цветное изображение на экране монитора формируется за счет смешивания трех базовых цветов: красного, зеленого и синего. Такая цветовая модель называется RGB-моделью по первым буквам английских названий цветов (Red, Green, Blue).

Векторная графика. Представляет изображение как набор простых геометрических фигур (примитивов): точки, прямые, окружности, прямоугольники, а также сплайны некоторого порядка. Указанным геометрическим объектам присваиваются некоторые атрибуты (толщина линии, тип линии, цвет заполнения и т.д.). Рисунок хранится как набор координат, векторов и других численных значений, характеризующих набор примитивов. Изображение в векторном формате отображается для редактирования без потери качества (в отличие от растрового изображения) и могут легко масштабироваться, поворачиваться, деформироваться. Тем не менее, не каждое изображение может быть представлено в виде набора примитивов. Этот способ представления графической информации используется для схем, масштабируемых шрифтов, деловой графики, создания мультфильмов и видеороликов.

В компьютерной графике для задания и хранения цвета применяются различные формы. В общем случае для задания цвета

используется его числовой код в заданной цветовой системе. Наиболее распространены следующие модели цветопередачи:

– RGB (Red, Green, Blue – красный, зеленый, синий). В этой модели цвета получаются путем добавления к черному. Выбор основных цветов обусловлен особенностями физиологии восприятия цвета сетчаткой человеческого глаза. Цветовая модель RGB широко применяется в компьютерной технике. Монитор содержит три вида светодиодов (светофильтров) для красного, зеленого и синего каналов.

– CMYK (Cyan, Magenta, Yellow, black – голубой, пурпурный, желтый, черный). Данная схема формирования цвета используется как в мониторах, так и в полиграфии.

– HSV (Hue, Saturation, Value – тон, насыщенность, значение). Это цветовая модель, в которой координатами являются цветовой тон, насыщенность (называемая также чистотой цвета) и значение (яркость) цвета. Указанная модель разработана на основании модели RGB.

В настоящее время компьютерная графика является одним из перспективных направлений развития компьютерных технологий.

Кодирование звуковой информации.

Звук представляет собой звуковую волну с непрерывно меняющейся амплитудой и частотой. Чем больше амплитуда сигнала, тем он громче для человека, чем больше частота сигнала, тем выше тон. Для того чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть превращен в последовательность электрических импульсов (двоичных нулей и единиц).

При двоичном кодировании непрерывного звукового сигнала он заменяется последовательностью дискретных уровней сигнала. Качество кодирования зависит от количества измерений уровня сигнала в единицу времени, то есть частоты дискретизации. Чем большее количество измерений производится за 1 секунду (чем больше частота дискретизации), тем точнее процедура двоичного кодирования. Для реализации данного процесса используются дополнительные устройства: аналого-цифрового преобразователя (из аналогового сигнала, разложенного в гармонические

ряды получают цифровой код) и цифро-аналоговый преобразователь (для воспроизведения звука). Качество кодирования зависит от частоты дискретизации, чем больше измерений в секунду производится, тем выше качество звука.

ЭВОЛЮЦИЯ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

СЧЕТНЫЕ УСТРОЙСТВА

Первым приспособлением для облегчения вычислений была специальная доска, названная впоследствии «абак» (V–IV вв. до н.э.). Вычисления на ней проводились перемещением мелких предметов в специальных углублениях доски. В Японии аналогичное устройство называлось «серобаян», в Китае – «суанпань». В Древней Руси аналогом данного приспособления были счеты. С помощью специальных костяшек они позволяли выполнять быстрые и довольно сложные расчеты.

В средние века в Европе появилась технология, использующая для вычислений специальные разграфленные таблицы (счет на линиях). Шотландский математик Джон Непер в 1617 г. разработал метод упрощенного умножения, основанный на применении счетных пластинок, и создал устройство, содержащее костяные пластинки с отпечатанными на них цифрами – в определенных сочетаниях эти пластинки позволяли выполнять умножение.

Настоящим предком современного компьютера считают механические часы. На основе часового механизма были созданы первые счетные устройства. Механизация арифметических операций над числами началась в XVII в. Первое механическое устройство на основе часового механизма было создано в 1623 году. Оно называлось «Суммирующие часы» и выполняло сложение над заданными двумя числами. Автором данной разработки был профессор кафедры восточных языков (Германия) Вильгельм Шикард.

В дальнейших разработках происходило совершенствование предложенного принципа вычислений. И уже в 1642 г. французский

механик Блез Паскаль представил более компактное устройство, которое могло выполнять основные математические действия. Это был первый механический калькулятор. В 1673 г. немецкий математик и философ Г.В. Лейбниц предложил более совершенное устройство, которое реализовывало операцию умножения (как многократное сложение) и деление (как многократное вычитание). На рисунке 1 представлена модель калькулятора Лейбница.

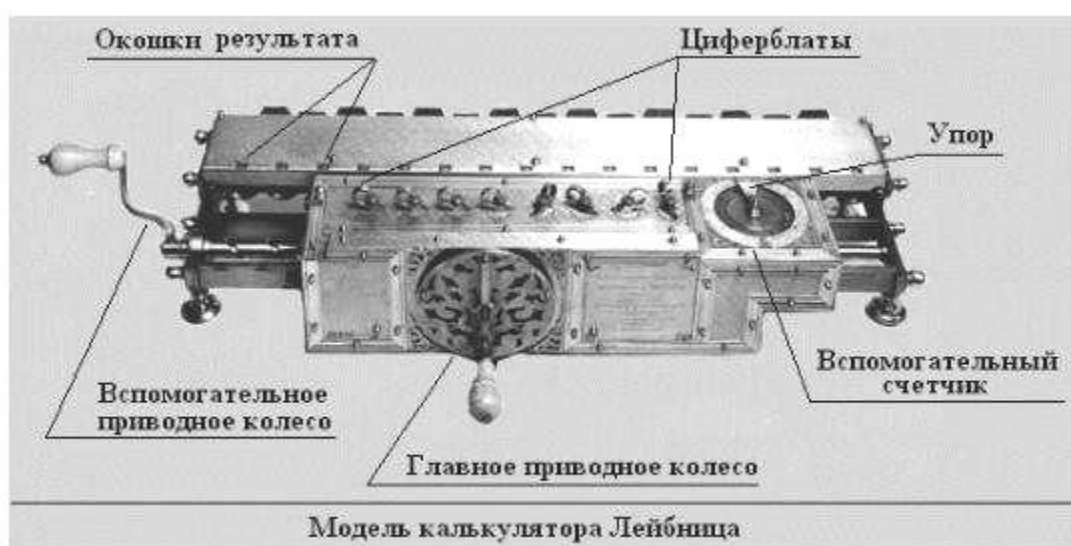


Рис. 1. Модель калькулятора Лейбница

Английскому математику Чарльзу Бэббиджу (1791–1871) принадлежит идея *гибкого управления* вычислительным процессом, которая была реализована в его Аналитической машине. Ч. Бэббиджем впервые был предложен и реализован *принцип разделения информации на команды и данные*. В его Аналитической машине было два вида памяти: «склад» и «мельница». Данные вводились в механическую память «склада» путем установки блоков шестерен, а потом обрабатывались в «мельнице» с использованием команд, которые вводились с перфокарт. В реализации этого проекта следует отметить роль графини Огасты Ады Лавлейс (1815–1852), которая предложила использовать перфокарты для программирования вычислительных операций (1843). Данное устройство можно считать первым в мире механическим компьютером, умеющим

хранить и обрабатывать информацию. Главным преимуществом данной разработки считается реализация автоматических вычислений.

Шведский инженер В. Однер в 1878 году разработал усовершенствованное механическое счетное устройство – арифмометр, который производился серийно на заводе в России.

Кроме совершенствования технических устройств для автоматизации счета развивались и математические основы информатики. Логика Аристотеля описывает правила логических рассуждений и формальные законы, используемые в данном процессе. Аристотель ввел понятие аксиом – утверждений, принимаемых без доказательств и установил основные принципы дедукции, которые позволяли, основываясь на аксиомах, строить безошибочные цепочки суждений. На этом фундаменте была основана Евклидова геометрия.

Рене Декарт использовал системы координат для геометрических построений и доказал, что геометрические действия могут быть заменены алгебраическими операциями над координатами объектов. Разработкой основ алгебры логики занимался Готфрид Лейбниц, ему же принадлежит идея применения двоичной системы счисления для представления данных при проведении автоматических вычислений.

В начале XIX века вышла работа Джорджа Буля «Законы мысли». Булю первому удалось выразить логические предложения в алгебраическом виде. Он постарался свести словесную логику к математической (булева алгебра). Каждое высказывание может быть истинным или ложным; из простых высказываний с помощью логических операций «и», «или», «не», «следует» строятся сложные по принципу «если» – «то».

Американский инженер Клод Шеннон занимался математическим описанием систем управления электрическими сетями. В процессе исследований он доказал, что, можно представить в двоичном виде не только числа, но буквы и знаки. Основным достижением Шеннона в развитии вычислительных устройств можно считать реализацию основных логических операций в виде электрической цепи, состоящей из

включенных и выключенных реле. В этом случае включенное реле означало единицу (истину), а выключенное – ноль (ложь)

Английский математик Алан Тьюринг в 1936 году придумал математическое описание очень простой машины с конечным числом состояний («состояния ума»). Данная машина могла «считывать» («записывать») конечное множество символов (алфавит машины) с бесконечной ленты и при этом изменять свое внутреннее состояние. Предложенная «Машина Тьюринга», которую можно считать моделью компьютера общего назначения, позволила формализовать понятие алгоритма и до сих пор используется во множестве теоретических и практических исследований. Следует отметить вклад А. Тьюринга в теорию искусственного интеллекта.

В 1948 году вышла книга Норберта Винера «Кибернетика, или управление и связь в животном и машине». Кибернетика – наука об общих принципах управления различных систем – технических, биологических, социальных. Предметом кибернетики являются принципы построения и функционирования систем автоматического управления. Кибернетика часто опирается на те же программные и аппаратные средства, что и информатика. Информатика в свою очередь заимствует у кибернетики математическую и логическую базу для развития этих средств.

ПОКОЛЕНИЯ КОМПЬЮТЕРОВ

Рассмотрим развитие современной вычислительной техники с точки зрения перехода на новую элементную базу. Переход на новые элементы в составе вычислительной техники обязательно влияет на основные технико-эксплуатационные и технико-экономические показатели. Изменяются характеристики устройства, такие как быстродействие, емкость памяти, надежность и стоимость. Стремление к улучшению данных показателей способствует разработке новых элементов и совершенствованию уже существующих [14].

Первым электрическим вычислительным устройством можно считать разработку американского инженера Говарда Эйкена – компьютер

«Марк-1». Он был разработан и создан при поддержке фирмы IBM в 1941 году. Основные элементы данного устройства – электромеханические реле. Созданные по данному принципу электрические схемы были недостаточно быстродействующие и, главное, недостаточно надежные в эксплуатации. Поэтому аналогичные разработки не получили широкого применения. Большинство авторов, исследующих развитие компьютеров, начинают отсчет поколений с устройств на электронных лампах.

Первое поколение компьютеров. Вычислительные машины данного поколения строились на электронных лампах. Характеризовались невысокой скоростью вычислений, малым объемом памяти. Управлялись вручную оператором. Программное обеспечение создавалось для каждой машины индивидуально. Программирование велось в машинных кодах. Область применения этих компьютеров – научные расчеты. Примером компьютера данного поколения является ЭНИАК (электронный вычислительный интегратор и калькулятор). Он разработан в 1946 году в Америке. Его быстродействие по сравнению с «Марк-1» было выше в 1000 раз. В компьютерах первого поколения для задания программы расчета необходимо было переделать подсоединения в электрических схемах и написать в машинных командах последовательность выполняемых действий. В это время выявилась потребность в разработке принципов работы электрических вычислительных устройств. Для хранения создаваемых программ и обрабатываемых данных была необходима собственная память компьютера достаточного размера. К этой работе был привлечен американский математик Джон фон Нейман, который разработал общие принципы функционирования и элементы архитектуры компьютеров, как универсальных вычислительных устройств, которые получили название *принципы фон Неймана*:

1. Принцип двоичного представления данных. Для кодирования хранимой в памяти компьютера информации (данных и команд) используется двоичная система счисления.

2. Принцип программного управления. Программа описывает последовательность действий над данными, состоит из команд, которые

выполняются друг за другом автоматически. Процессор после выполнения очередной команды, загружает из памяти следующую. Специальный регистр процессора (счетчик команд) отвечает за очередность их выполнения. Для реализации данного процесса обработки необходимо, чтобы команды программы хранились в памяти компьютера последовательно друг за другом. Если в программе был необходим безусловные или условный переход, то за это отвечали специальные команды, задающие новый номер ячейки памяти с новой командой. Выборка команд из памяти прекращается после достижения и выполнения команды «*стоп*».

3. Принцип однородности памяти. Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти – число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Это открывает целый ряд возможностей. Например, программа в процессе своего выполнения также может подвергаться переработке, что позволяет задавать в самой программе правила получения некоторых ее частей (так в программе организуется выполнение циклов и подпрограмм).

Более того, команды одной программы могут быть получены как результаты исполнения другой программы. На этом принципе основаны методы трансляции – перевода текста программы с языка программирования высокого уровня на язык конкретной машины.

4. Принцип адресности. Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

Отсюда следует возможность давать имена областям памяти так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

5. Принцип последовательного программного управления. Все команды располагаются в памяти и выполняются последовательно, одна после завершения другой.

6. Принцип условного перехода. Команды из программы не всегда выполняются одна за другой. Возможно присутствие в программе команд условного перехода, которые изменяют последовательность выполнения команд в зависимости от значений данных. (Сам принцип был сформулирован задолго до фон Неймана Адой Лавлейс и Чарльзом Бэббиджем, однако, он логически включен в фоннеймановский набор как дополняющий предыдущий принцип).

В соответствии с принципами фон Неймана компьютер должен состоять из следующих устройств:

- арифметико-логическое устройство (АЛУ), для реализации основных арифметических и логических операций над данными;
- устройство управления (УУ), отвечает за синхронную работу всех составляющих устройств компьютера;
- память. В соответствии с принципом однородности она единая для программ и обрабатываемых данных;
- внешние устройства (ВУ). Используются для ввода информации в компьютер и вывода результатов.

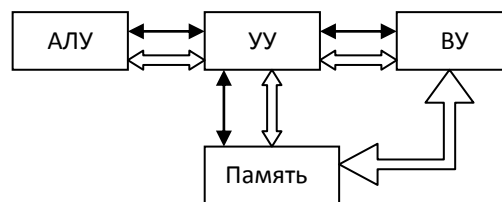


Рис. 2. Структурная схема компьютера

- ⇔ - информационные связи
- - управляющие связи

АЛУ и УУ объединены в единое устройство – процессор (центральное обрабатывающее устройство).

В процессоре реализована внутренняя память – регистры. Все регистры специализированы и хранят соответствующую им информацию. АЛУ получает данные для обработки и команды из соответствующих регистров и в них же помещает результат. Для реализации регистров

используется электронная схема (*триггер*), которая может находится в двух устойчивых состояниях (1 и 0). Набор триггеров с общей системой управления и образует регистр процессора

Память компьютера состоит из ячеек размером 8 двоичных разрядов (1 байт). Таким образом в ячейке может храниться команда или элемент данных для обработки. Процессор должен иметь возможность обращения к каждой ячейке памяти, поэтому каждая ячейка имеет уникальное имя – номер по порядку следования в памяти.

Основные функции памяти компьютера:

- запись в ячейки данных, поступающих из устройств;
- устойчивое хранение информации;
- возможность выдачи хранимой информации по запросу от других устройств, например, процессора.

Очередная обрабатываемая процессором команда содержит в себе информацию о том, из каких ячеек взять текущие данные, какую команду над ними выполнить; по какому адресу записать результат; как изменить содержимое счетчика команд, чтобы знать, откуда взять следующую команду для выполнения.

С развитием и совершенствованием компьютерной техники предложенные принципы *фон Неймана* изменялись и дополнялись, но исходные принципы управления работой компьютера с помощью хранящихся в памяти программ остались неизменными. Подавляющее большинство современных компьютеров построено именно по архитектуре фон Неймана.

В 1951 году был создан первый компьютер, предназначенный для коммерческого использования, в котором были реализованы все принципы архитектуры фон Неймана – УНИВАК (универсальный автоматический компьютер).

В СССР, в Москве и Ульяновске, выпускались большие электронные счетные машины БЭСМ-1, БЭСМ-2 (главный конструктор – Сергей Алексеевич Лебедев) и самая быстродействующая в Европе ЭВМ того времени М-20 (конструкторы – С.А. Лебедев и Ю. А. Базилевский).

Основными недостатками элементной базы данного поколения было то, что электронные лампы энергоемки, громоздки, ненадежны в работе и быстро греются, что требовало создания определенного теплового режима в помещении. В 1948 году был изобретен транзистор. Данное устройство основывается на электрических свойствах полупроводников.

Второе поколение компьютеров. Элементная база – транзисторы. Переход к новой элементной базе позволил существенно уменьшить размеры вычислительных устройств. Транзисторы оказались надежнее в эксплуатации, чем электронные лампы. Также следует сказать о повышении быстродействия новых компьютеров.

В 1954 году началось серийное производство транзисторов, а в 1956 году ученые Массачусетского технологического института создали первый полностью построенный на транзисторах компьютер.

Как уже отмечалось, для компьютеров первого поколения программы писались в машинных кодах. Распространение компьютеров и расширение области их использования потребовало разработки новых языков программирования – более понятных человеку. Первым шагом в этом направлении было создание *ассемблеров*. Впервые для задания команд стали использоваться символьные конструкции. Это делало программу более понятной, а процесс написания новых программ – более быстрым. При запуске ассемблерной программы в процессе обработки символические имена заменялись на соответствующие коды или адреса ячеек с данными.

Развитие компьютеров и программного обеспечения позволило к середине 60-х годов прошлого века использовать автоматическую обработку данных в крупных компаниях, образовательных и научных центрах.

Новые области применения компьютера требовали нового программного обеспечения и создания новых языков, позволяющих решать поставленные перед программистами задачи. В это время появляются языки программирования высокого уровня. Для их использования необходим транслятор – программа преобразующая исходный текст в машинные коды. Появляются трансляторы для языка

программирования Паскаль, Фортран, Алгол, Си и другие. Язык программирования Фортран создавался для математических расчетов, а язык Кобол был специально создан с учетом особенностей обработки финансово-экономической информации. Именно с данным периодом развития вычислительной техники связывают начало развития индустрии программного обеспечения. Появляются первые операционные системы для управления работой компьютера. Быстродействие компьютеров второго поколения достигает нескольких сотен тысяч операций в секунду.

В начале 60-х годов в СССР было освоено серийное производство ЭВМ 2-го поколения на транзисторах: М-220, БЭСМ-3, БЭСМ-4, В 1966 г. в СССР была введена в эксплуатацию быстродействующая ЭВМ БЭСМ-6, которая производилась до 1988 г. Большое быстродействие БЭСМ-6 обусловили мультипрограммный режим работы и конвейерная процедура обработки данных, которые в настоящее время используются во многих современных ЭВМ.

Самой сложной операцией при создании электронных схем была спайка транзисторов. В 1959 году Роберт Нойс изобрел способ, позволяющий создавать на одной пластине кремния транзисторы и все необходимые соединения между ними. Полученные электронные схемы стали называться *интегральными схемами* или *чипами*. Впоследствии Роберт Нойс основал компанию Intel по производству интегральных микросхем. Микросхемы работали значительно быстрее транзисторов и потребляли значительно меньше энергии.

Третье поколение компьютеров. Элементной базой ЭВМ данного поколения ЭВМ являются интегральные схемы (ИС), производство которых в 1961 г. начала фирма Intel (США).

Характерными признаками ЭВМ являются развитая конфигурация внешних устройств с использованием стандартных средств сопряжения, высокое быстродействие и большой объем основной и внешней памяти, развитая операционная система, обеспечивающая работу в мультипрограммном режиме. На основе ЭВМ третьего поколения создавались многопроцессорные, многомашинные и многоканальные вычислительные комплексы. Сначала интегральные схемы состояли из

нескольких элементов, затем развитие технологии позволило разместить на одной плате десятки и сотни транзисторов. Это привело к существенному уменьшению размеров вычислительных устройств. Применение интегральных схем позволило также уменьшить стоимость компьютеров и повысить их надежность.

В 1964 году компания IBM выпустила компьютер IBM System 360, построенный на основе интегральных микросхем. Выпуск этих компьютеров можно считать началом массового производства вычислительной техники.

Данное поколение компьютеров характеризуется началом массового производства. ЭВМ стали использоваться не только крупными компаниями. Это позволило развиваться индустрии программного обеспечения, появлением коммерческих программных продуктов, появлением новых языков программирования. Были спроектированы внешние устройства для расширения возможностей применения компьютера.

Четвертое поколение компьютеров. Основой ЭВМ являются большие интегральные схемы (СБИС) и микропроцессоры, выпуск которых был начат в 1971 г. фирмой Intel (США). Для управления ЭВМ четвертого поколения разрабатываются многозадачные и многопользовательские операционные системы, манипуляторы, устройства речевого ввода и вывода, средства мультимедиа, языки, поддерживающие искусственный интеллект.

Микропроцессор, лежащий в основе архитектуры компьютера данного поколения, – это процессор, выполненный в виде одной микросхемы. В это время появляется микропроцессорные компьютеры, или, персональные компьютеры (ПК), которые характеризуются невысокой стоимостью, небольшими размерами и достаточным быстродействием для решения задач пользователя.

Первой компанией, представившей на рынок персональный компьютер была фирма Apple, она в 1976 представила модель Apple I, а затем в следующем году модель Apple II. Но они не получили такого

широкого спроса и распространения, на который рассчитывали ее создатели.

Удачным коммерческим проектом стал ПК фирмы IBM, которая до этого разрабатывала компьютеры для крупных фирм. В этой разработке был применен принцип «открытой архитектуры», в соответствии с которым узлы нового ПК разрабатывались разными производителями, стандартизировались только способы их соединения. Для создания ПК был использован 16-разрядный микропроцессор Intel-8088. Программное обеспечение для данного компьютера было разработано фирмой Microsoft. Все это позволило существенно снизить цену на ПК и способствовало его широкому распространению и развитию программного обеспечения для данного вида компьютера [13].

Принцип открытой архитектуры заключается в следующем:

- Регламентируются и стандартизируются только *описание принципа действия компьютера и его конфигурация* (определенная совокупность аппаратных средств и соединений между ними). Таким образом, *компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-производителями.*

Компьютер *легко расширяется и модернизируется* за счет наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым *устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.*

Широкое распространение персональных компьютеров фирмы IBM подтолкнуло развитие индустрии комплектующих и дополнительных устройств для IBM PC. Естественно, это привело к их удешевлению и появлению целого семейства IBM-совместимых компьютеров. Владельцы ПК могли сами модифицировать свои компьютеры, используя имеющиеся на рынке узлы и устройства.

Пятое поколение компьютеров. Современные вычислительные системы работают по принципу «управления потоками данных» (в отличие от принципа фон Неймана «управления потоками команд»).

В качестве перспективных направлений дальнейшего развития вычислительной техники можно указать следующие:

- достижение предельной плотности упаковки элементов в СБИС на кремниевой основе;
- производство СБИС на основе арсенида галлия;
- использование криогенной технологии на базе эффекта Джозефсона.

Архитектуры ЭВМ совершенствуются по следующим направлениям:

- системы ЭВМ различной мощности, сбалансированные по архитектуре, что позволит быстро, просто и эффективно использовать огромный потенциал такой системы для обработки различной информации;
- однопроцессорные ПЭВМ с командным управлением на новой быстродействующей элементной базе;
- ЭВМ на нескольких быстрых процессорах с командным управлением, часть которых является универсальными, а другая часть конвейерными или параллельными с небольшим числом процессорных элементов;
- высокопроизводительные многопроцессорные ЭВМ с конвейерной, параллельной или матричной обработкой информации.

Совершенствование интеллектуальных интерфейсов:

- технические и программные средства ввода/вывода различных видов информации;
- общение на проблемно-ориентированном естественном разговорном языке;
- использование текстовых документов, как печатных, так и рукописных, и изображений;
- всемерное развитие известных и новых алгоритмических языков программирования;
- применение языков искусственного интеллекта: Лисп, Пролог, PS, FRL, VALID, OCCAM и др.

СИСТЕМЫ СЧИСЛЕНИЯ

ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

Система счисления – это совокупность правил записи чисел посредством определенного набора цифр. Все существующие системы счисления можно разделить на:

- непозиционные (в этих системах значение цифры не зависит от ее позиции – положения в записи числа);
- позиционные (значение цифры зависит от позиции).

Примером позиционной является арабская системы счисления. В ней используются цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. В позиционных системах счисления количество, обозначаемое цифрой в числе, зависит от ее позиции. Например, число 77 – здесь первая семерка обозначает десятки, а вторая – количество единиц.

В качестве примера непозиционной системы счисления можно привести римский способ записи чисел. В ней для обозначения чисел 1, 5, 10, 50, 100, 500 и 1000 используются соответственно заглавные латинские буквы I, V, X, L, C, D и M соответственно. Число в римской системе счисления представляет собой последовательность стоящих подряд цифр. Для определения значения числа по его римской записи надо выполнить определенные действия.

Методы определения значения числа:

1. Значение числа равно сумме значений его цифр. Например, число 32 в римской системе счисления имеет вид XXXII= $(X+X+X)+(I+I)=30+2=32$.

2. Если слева от большей цифры стоит меньшая, то значение равно разности между большей и меньшей цифрами. При этом, левая цифра может быть меньше правой максимум на один порядок: так, перед L(50) и C(100) из «младших» может стоять только X(10), перед D(500) и M(1000) – только C(100), перед V(5) – только I(1).

Например, число 777 в рассматриваемой системе счисления будет записано в виде

$$DCCVII (D+C+C+V+I+I) = 500+100+100+5+1+1=777.$$

Число 447 будет иметь вид

$$CDXLVII ((D-C) + (L-X) + V+I+I)=(500-100)+(50-10)+5+1+1=447.$$

Таким образом, чтобы оперировать с числами в определенной системе счисления надо знать: основание системы счисления, набор используемых цифр, для позиционных систем счисления еще вес разряда.

В позиционной системе счисления разряды числа нумеруются справа налево, начиная со значения ноль. Если необходимо пронумеровать дробную часть числа – используются отрицательные значения. Например, в следующей таблице приведено число 5762,13 и номера его разрядов.

Число	5	7	6	2	.	1	3
Разряды числа	3	2	1	0		-1	-2

Перевод в десятичную систему счисления

Пусть в обозначении « s^i », i – номер разряда, а s – основание системы счисления. Тогда, обозначив цифры числа как a_i , любое число, записанное в позиционной системе счисления, можем представить в виде:

$$x = a_n s^n + a_{n-1} s^{n-1} + \dots + a_2 s^2 + a_1 s^1 + a_0 s^0 + a_{-1} s^{-1} + \dots$$

Рассмотрим пример формирования десятичного целого числа:

$$2019_{10} = 2 \cdot 10^3 + 0 \cdot 10^2 + 1 \cdot 10^1 + 9 \cdot 10^0$$

Решив полученное арифметическое выражение, мы получаем число 2019 в десятичной системе счисления.

Рассмотрим алгоритм перевода заданного числа из любой заданной системы счисления в десятичную:

1. Пронумеруем разряды исходного числа справа налево;
2. Запишем арифметическое выражение, слагаемые которого получаются как произведения очередной цифры заданного числа на число десять, возведенное в степень, равную номеру разряда;
3. Вычислим результат арифметического выражения.

Примеры:

$$1001111_2 = 79_{10}$$

6	5	4	3	2	1	0
1	0	0	1	1	1	1

$$= 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 79$$

$$11011_2 = 27_{10}$$

4	3	2	1	0
1	1	0	1	1

$$= 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 27$$

Для перевода числа из десятичной в другую систему счисления используется следующий алгоритм:

1. Выполним последовательное деление с **остатком** исходного числа и каждого полученного частного на основание новой системы счисления.

2. Запишем вычисленные остатки и последнее частное, начиная с последнего значения (т.е. в обратном порядке)

Рассмотрим перевод десятичного числа в двоичное. Для этого выполним последовательное деление числа на два и формирование двоичного числа из полученных на каждом шаге остатков. Например, нужно получить из числа 77 его двоичную запись:

$$77 / 2 = 38 \text{ (1 остаток)}$$

$$38 / 2 = 19 \text{ (0 остаток)}$$

$$19 / 2 = 9 \text{ (1 остаток)}$$

$$9 / 2 = 4 \text{ (1 остаток)}$$

$$4 / 2 = 2 \text{ (0 остаток)}$$

$$2 / 2 = 1 \text{ (0 остаток)}$$

$$1 / 2 = 0 \text{ (1 остаток)}$$

Собираем остатки вместе, начиная с конца: 1001101. Это и есть число 77 в двоичном представлении. Проверим:

$$1001101 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 64 + 0 + 0 + 8 + 4 + 0 + 1 = 77$$

Примеры:

1) Перевод числа 49 из десятичной системы счисления в двоичную

$$\begin{array}{r}
 49 \mid 2 \\
 \hline
 48 \mid 24 \mid 2 \\
 \hline
 124 \mid 12 \mid 2 \\
 \hline
 012 \mid 6 \mid 2 \\
 \hline
 063 \mid 2 \\
 \hline
 02 \mid 1 \\
 \hline
 1
 \end{array}$$

Таким образом число 49_{10} в двоичной системе счисления будет выглядеть 110001_2 .

Перевод числа из десятичной системы счисления в двоичную можно выполнить через подбор степеней двойки. Для этого необходимо хорошо знать таблицу степеней двойки (Приложение 2).

$$49 = 32 + 16 + 1 = 2^5 + 2^4 + 2^0$$

$$49_{10} = 110001_2$$

разряд	5	4	3	2	1	0
Значение разряда	1	1	0	0	0	1

2) Перевод числа 49 из десятичной системы счисления в восьмеричную

$$\begin{array}{r}
 49 \mid 8 \\
 \hline
 48 \mid 6 \\
 \hline
 1
 \end{array}$$

В результате число 49_{10} в восьмеричной системе счисления будет выглядеть как 61_8

3) Перевод числа 49 из десятичной системы счисления в шестнадцатеричную

$$\begin{array}{r}
 49 \mid 16 \\
 \hline
 48 \mid 3 \\
 \hline
 1
 \end{array}$$

В результате число 49_{10} в шестнадцатеричной системе счисления будет выглядеть как 31_{16} .

Почему нам необходимо знать представление чисел в двоичной системе счисления? Каждая обрабатываемая компьютером цифра должна быть определенным образом записана на физическом носителе информации. В современных вычислительных системах память изготовлена из элементов, которые могут находиться в двух устойчивых состояниях. Эти состояния соответствуют значениям 0 и 1. Это одна из основных причин, почему двоичной системе счисления уделяется столько внимания.

При работе с компьютерами кроме двоичной системы счисления используют также восьмеричную и шестнадцатеричную, запись в которых более компактна и удобна для человека. С другой стороны, благодаря тому, что 8 и 16 – степени 2, переход между записью в двоичной и одной из этих систем осуществляется без вычислений.

Для перевода числа из двоичной системы счисления (2сс) в шестнадцатеричную (16сс) достаточно заменить каждые четыре двоичных разряда одним шестнадцатеричным числом ($16=2^4$) и наоборот при переводе из шестнадцатеричной системы счисления в двоичную (значения чисел в разных системах счисления приведены в Приложении 1).

Примеры:

1) Дано число 1011011_2 , справа налево выделим в нем по четыре разряда, если необходимо, слева дополним нулями. Получаем две группы чисел и затем по таблице определяем соответствующие им шестнадцатеричные значения.

16сс	5	B
2сс	0101	1011

$$1011011_2 = 5B_{16}$$

2) Дано число 1011110101_2 , справа налево выделим в нем по четыре разряда, если необходимо, слева дополним нулями. Получаем три группы чисел и затем по таблице определяем соответствующие им шестнадцатеричные значения.

16сс	2	F	5
------	---	---	---

2сс	0010	1111	0101
-----	------	------	------

$$1011110101_2 = 2F5_{16}$$

Аналогично происходит и перевод между двоичной и восьмеричной системой счисления (8сс), учитывая, что один разряд восьмеричной записи соответствует трем разрядам двоичной ($8=2^3$).

Примеры:

1) Дано число 1011011_2 , справа налево выделим в нем по три разряда, если необходимо, слева дополним нулями. Получаем три группы чисел и затем по таблице определяем соответствующие им восьмеричные значения.

8сс	1	3	3
2сс	001	011	011

$$1011011_2 = 133_8$$

2) Дано число 1011110101_2 , справа налево выделим в нем по четыре разряда, если необходимо, слева дополним нулями. Получаем три группы чисел и затем по таблице определяем соответствующие им шестнадцатеричные значения.

8сс	1	3	6	5
2сс	001	011	110	101

$$1011110101_2 = 1365_8$$

Дополнительный код числа

Очень часто в вычислениях используются не только положительные, но и отрицательные числа. Для представления чисел со знаком в вычислительной технике выделяется старший разряд числа в качестве знакового. Принято считать, что 0 в знаковом разряде означает положительное значение, а 1 – отрицательное». Рассмотрим однобайтовое представление числа. Седьмой бит – это знаковый разряд, остальные разряды хранят само число.

Разряды восьмибитного двоичного числа							
7	6	5	4	3	2	1	0
Знаковый разряд	Значение числа						

Выполнение арифметических операций над числами с разными знаками представляется для аппаратной части довольно сложной процедурой. Так как при этом необходимо определять знак результата. Использование в расчетах дополнительного кода для отрицательного числа позволяет выполнить операции алгебраического суммирования и вычитания на обычном сумматоре.

Прямой код представляет собой одинаковое представление значимой части числа для положительных и отрицательных чисел и отличается только знаковым битом.

Обратный код для положительных чисел имеет тот же вид, что и прямой код, а для отрицательных чисел образуется из прямого кода положительного числа путем инвертирования всех значащих разрядов прямого кода.

Для получения дополнительного кода для отрицательных чисел необходимо получить сначала его двоичное представление как для положительного числа, затем получить его обратный код и далее прибавить к нему единицу.

Такое представление чисел приводит к асимметрии диапазонов представления чисел относительно нуля. Так, в однобайтовом представлении диапазон чисел с учетом знака следующий:

$$-128 \leq x \leq +127.$$

Если оба числа имеют одинаковую размерность (одинаковое количество разрядов для хранения), то алгебраическая сумма будет получена по правилам двоичного сложения (включая знаковый разряд), если отбросить возможный перенос из старшего разряда.

Примеры.

$$1) 10_{10} + 25_{10} = 35_{10}$$

Рассмотрим поразрядное выполнение этой операции в байтовом представлении.

	0	0	0	0	1	0	1	0
+	0	0	0	1	1	0	0	1
	0	0	1	0	0	0	1	1

$$2) 10_{10} - 5_{10} = 5_{10}$$

Рассмотрим поразрядное выполнение этой операции в байтовом представлении.

$$10_{10} = 00001010_2$$

Для числа «-5» получим дополнительный код.

$$5_{10} = 00000101_2$$

Инвертируем полученное представление для получения обратного кода, получим

$$11111010$$

К обратному коду прибавим единицу, получим

	1	1	1	1	1	0	1	0
+								1
	1	1	1	1	1	0	1	1

Дополнительный код для числа «-5» получился 11111011_2

Выполним операцию сложения

	0	0	0	0	1	0	1	0
+	1	1	1	1	1	0	1	1
	0	0	0	0	0	1	0	1

Получено значение $101_2 = 5_{10}$.

$$3) -10_{10} + 5_{10} = -5_{10}$$

Рассмотрим поразрядное выполнение этой операции в байтовом представлении.

$$5_{10} = 00000101_2$$

Для числа -10 получим дополнительный код.

$$10_{10} = 00001010_2$$

Инвертируем полученное представление для получения обратного кода, получим

$$11110101$$

К обратному коду прибавим единицу, получим

	1	1	1	1	0	1	0	1
+								1
	1	1	1	1	0	1	1	0

Дополнительный код для числа -10 получился 11110110_2

Выполним операцию сложения

	1	1	1	1	0	1	1	0
+	0	0	0	0	0	1	0	1
	1	1	1	1	1	0	1	1

Полученное значение 11111011_2 находится в дополнительном коде, так как самый левый разряд равен единице. Переведем это число в прямой код. Для этого вычтем из полученного представления единицу и затем инвертируем разряды.

	1	1	1	1	1	0	1	1
-	0	0	0	0	0	0	0	1
Обратный код	1	1	1	1	1	0	1	0
Прямой код	0	0	0	0	0	1	0	1

Получено число $101_2 = 5_{10}$

СПОСОБЫ ОПИСАНИЯ АЛГОРИТМА

При изучении информатики немало внимания уделяется изучению алгоритмов и их видам. Не зная основных сведений о них, нельзя написать программу или проанализировать ее работу [1, 2].

Алгоритм – это четко заданная последовательность действий, приводящая к достижению результата. При составлении алгоритма необходимо разделить исходную задачу на последовательность простых действий, выполняющихся друг за другом. Выделяют следующие основные свойства алгоритма:

- **Определенность (детерминированность)** состоит в том, что любой алгоритм позволяет получить результат при заданных корректных исходных данных;
- **Дискретность** означает, что любой алгоритм состоит из последовательности простых этапов, каждый из которых выполняет определенное действие;
- **Конечность** означает, что при любых исходных данных алгоритм достигнет конечной вершины за конечное число шагов;
- **Результативность** означает, что после выполнения ряда действий будет достигнут определенный, ожидаемый результат.
- **Массовость** означает, что данный алгоритм может быть использован для решения задач определенного типа.

Существует несколько способов записи алгоритма:

- На естественном языке;
- Псевдокод;
- Граф-схемы алгоритма.

Запись алгоритма на естественном языке означает подробное описание последовательности выполняемых действий. Обычно используется при описании простых алгоритмов или на этапе разбиения сложной задачи на простые подзадачи. Например, алгоритм решения квадратного уравнения в словесной форме может выглядеть так:

Пример 1. Алгоритм решения квадратного уравнения.

- Записать вид квадратного уравнения;
- Определить исходные данные;
- Выбрать метод решения квадратного уравнения (например, через дискриминант);
- Вычислить дискриминант;
- Если дискриминант меньше нуля, то считаем, что решение не существует;
- Если дискриминант равен нулю, то существует единственный корень уравнения;
- Если дискриминант больше нуля, то существуют два корня уравнения.

Пример 2. Алгоритм нахождения наибольшего общего делителя.

1. Дано два числа;
2. Если числа равны, то ответ равен введенному числу, конец алгоритма, в противном случае продолжаем алгоритм;
3. Определяем большее из чисел;
4. Заменяем большее из чисел разностью большего и меньшего;
5. Повторяем алгоритм с шага 2.

Запись алгоритма на псевдокоде не содержит строгих правил для записи последовательности команд, как для языков программирования. Однако имеются некоторые четко определенные конструкции (служебные слова) которые облегчают переход от записи алгоритма на псевдокоде к программе на выбранном языке программирования. Запись алгоритма на псевдокоде обычно используется на этапе проектирования алгоритма решения поставленной задачи и позволяет описать ее более свободно и понятно. Используемые в псевдокоде ключевые слова выделяются в тексте (использование прописных букв или подчеркивание).

Пример 3. Запись алгоритма нахождения наибольшего общего делителя (Алгоритм Евклида) на псевдокоде.

1. **НАЧАЛО ФУНКЦИИ**
2. **ВВОД** чисел a и b
3. **ПОВТОРЯТЬ**
4. **ЕСЛИ** $a=b$ **ТО ВЫПОЛНИТЬ** Результат= a **ВОЗВРАТ** Результата
5. **ИНАЧЕ ЕСЛИ** $a>b$ **ТО** $a=a-b$ **ИНАЧЕ** $b=b-a$
6. **КОНЕЦ ПОВТОРЕНИЯ**
7. **КОНЕЦ ФУНКЦИИ**

Самым распространенным способом записи алгоритма считается граф-схема (или блок-схема). Для обозначения каждой операции в граф-схеме используются специальные геометрические фигуры, зафиксированные в стандарте [8].

Граф-схемой называется графическое изображение логической структуры алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций. Перечень блоков, их наименование, отображаемые ими функции, форма и размеры определяются стандартами.

Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий. В таблице 1 приведены наиболее часто употребляемые при изображении блок-схемы символы.

Линейным называется такой вычислительный процесс, при котором все этапы решения задачи выполняются в естественном порядке следования записи этих этапов.

Ветвящимся называется такой вычислительный процесс, в котором выбор направления обработки информации зависит от исходных или промежуточных данных (от результатов проверки выполнения какого-либо логического условия).

Таблица 1. Обозначения в блок-схемах

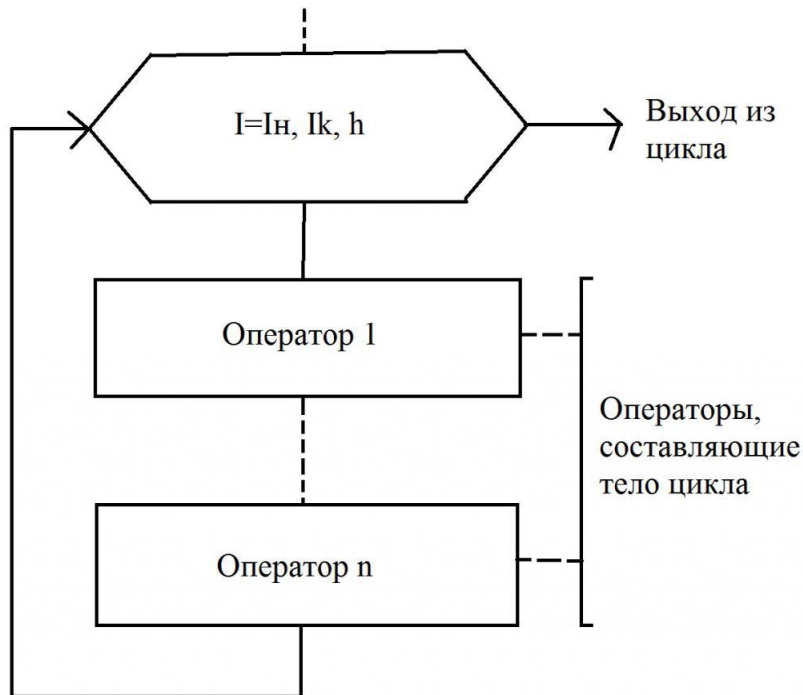
Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Цикл с заданным числом повторений
Предопределенный процесс		Вычисления по подпрограмме
Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму

Циклом называется многократно повторяющаяся последовательность действий. Вычислительный процесс, содержащий один или несколько циклов, называется **циклическим**. По количеству выполнения циклы делятся на циклы с определенным (заранее заданным) числом повторений и циклы с неопределенным числом повторений. По структуре циклы можно разделить на цикл с предусловием (условие проверяется в начале каждой итерации) и цикл с постусловием (условие проверяется в конце каждой итерации).

Основные разновидности циклов представлены ниже:

1. **Цикл со счетчиком**. Такой тип алгоритмов показывает, что заранее известно количество повторений данного цикла. Задана переменная (I – счетчик цикла), которая считает число шагов (итераций). Перед выполнением цикла нужно присвоить начальное значение счетчику (I_н) – это может быть любое число, оно зависит от конкретного алгоритма. В начале каждой итерации проверяется условие. Если оно истинно, начнет

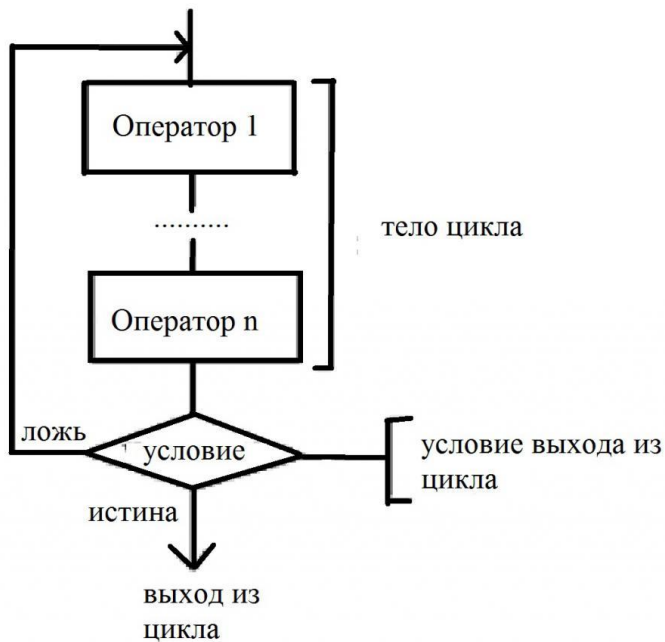
выполняться определенная группа команд, которые составляют тело цикла. В конце каждой итерации счетчик меняется на величину шага (значение h). Цикл закончится, когда значение параметра цикла будет меньше, чем конечное значение (I_k).



2. Цикл с предусловием. В этом виде цикла также непосредственно перед началом выполнения тела осуществляется проверка условия. В случае истинности условия происходит переход к началу тела цикла. В цикле необходимо изменять параметра цикла или переменной, которая влияет на поставленное условие. Если не придерживаться этого правила, получим «зацикливание». Выход из цикла происходит, когда условие ложно. В блок-схемах алгоритмов допускается осуществлять проверку не истинности, а ложности начального условия. Оба варианта правильные, их использование зависит от того, какой конкретно удобнее использовать для решения той или иной задачи. Особенность данного вида цикла – тело может не выполниться ни разу. Ниже приведена блок-схема, которая описывает все эти действия:

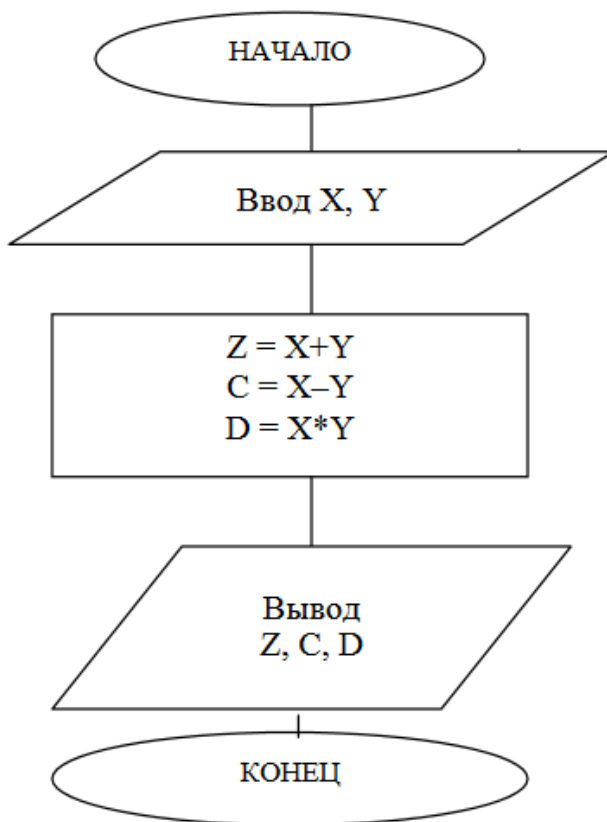


3. Цикл с постусловием. В этом типе цикла условие для выхода проверяется после тела цикла. Особенность данного вида цикла – тело цикла обязательно выполнится хотя бы один раз. В этом виде цикла также может использоваться счетчик (параметр цикла).

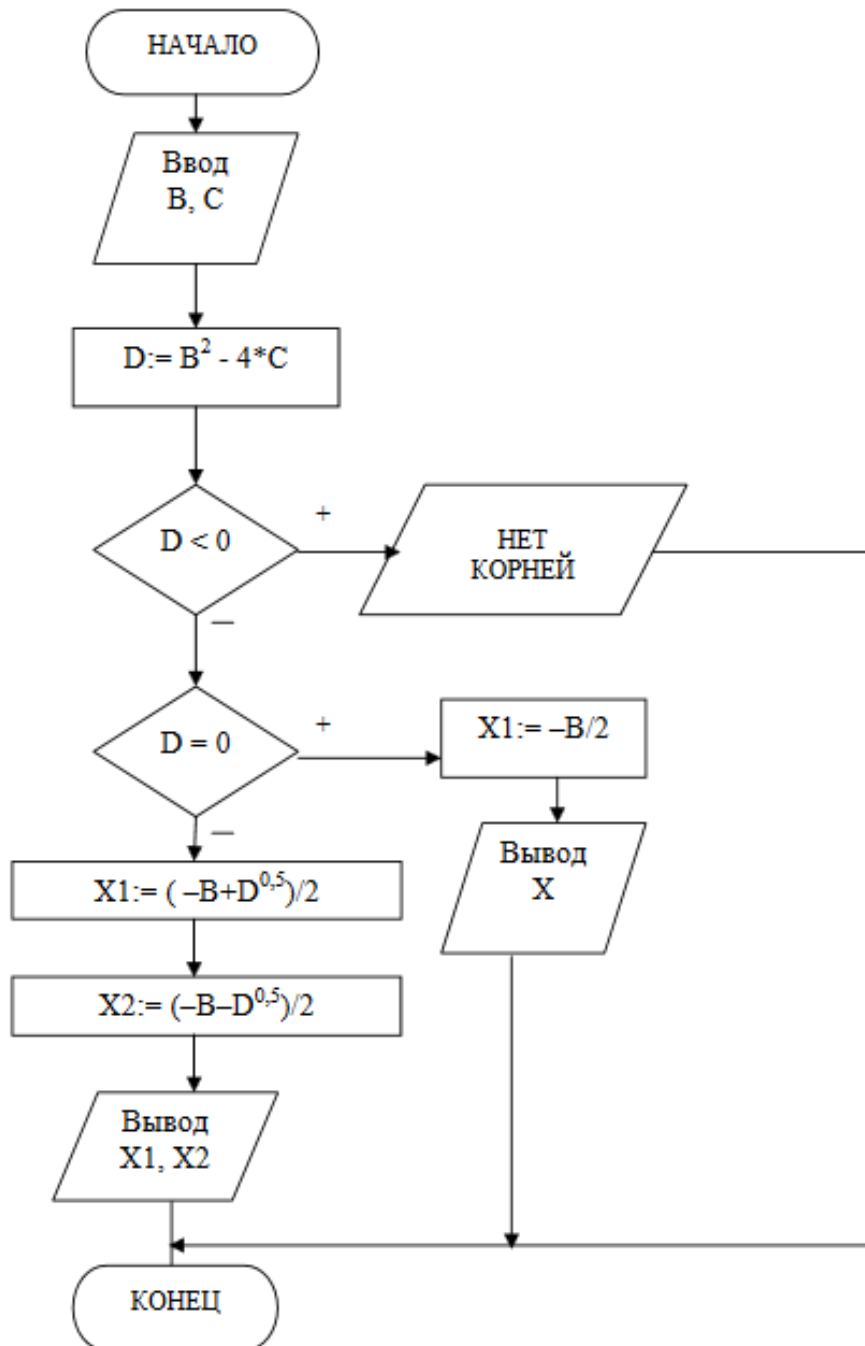


Циклы могут быть вложены один в другой, такие конструкции называют – вложенными. При этом те конструкции, которые охватывают иные циклы, называют «внешними». А те, которые входят в конструкцию внешних – «внутренними». При выполнении каждого шага внешнего цикла происходит полная прокрутка внутреннего.

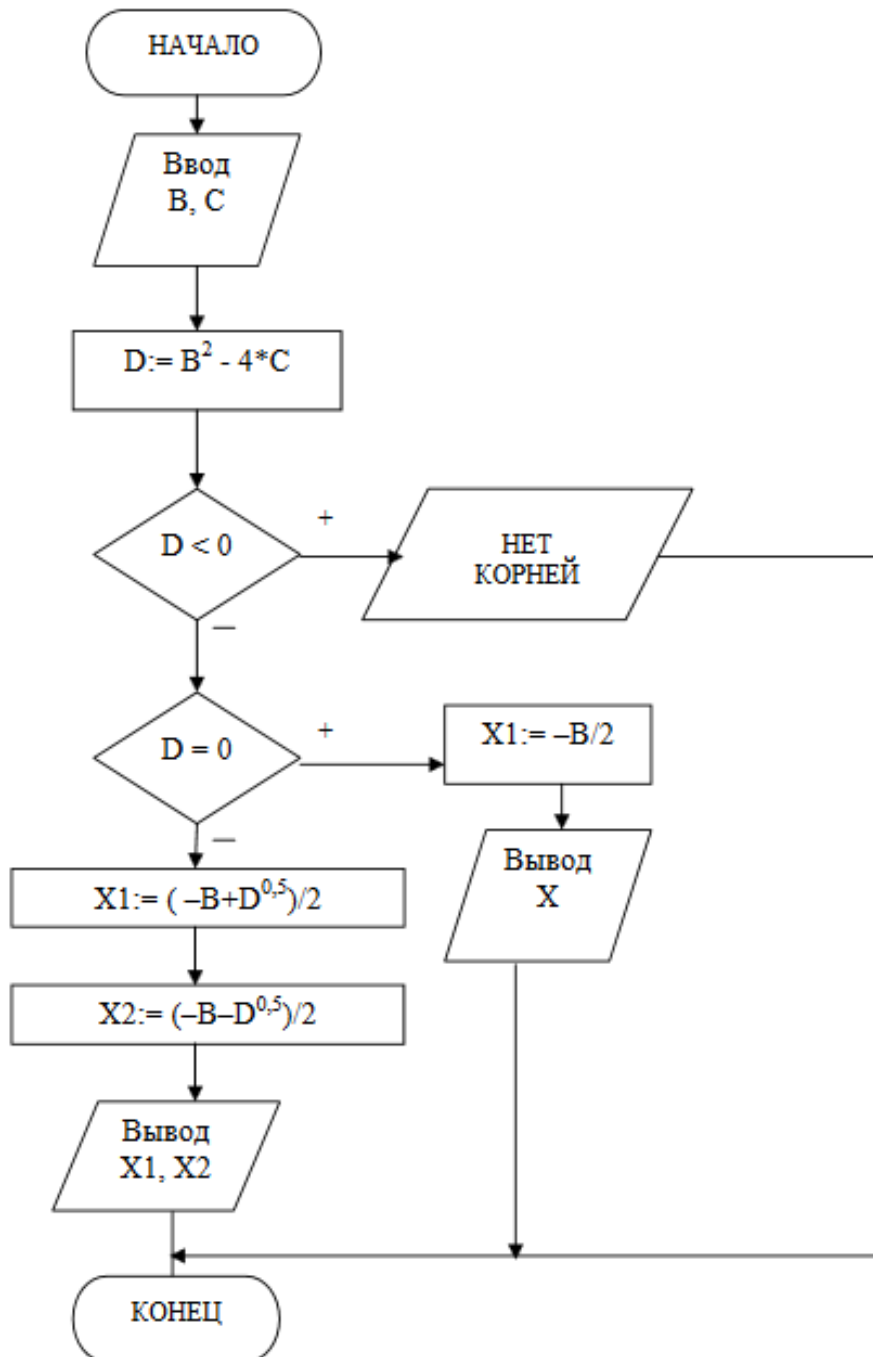
Пример линейного алгоритма. Составить блок-схему алгоритма для вычисления суммы, разности и произведения двух введенных переменных X и Y.



Пример разветвленного алгоритма. Составить блок-схему алгоритма поиска решения квадратного уравнения. Знаками + и – на блок-схеме обозначается истинность и ложность условия соответственно.



Пример циклического алгоритма. Составить блок-схему циклического алгоритма (цикл с условием) для подсчета суммы ряда $S=x+x^2+x^3+x^4+\dots+x^N$



АППАРАТНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

Рассмотрим устройство настольного персонального компьютера на примере ПК фирмы IBM (International Business Machines Corporation). Аппаратное обеспечение компьютера обозначают английским термином «Hardware».

Архитектурой компьютера называется его описание на некотором общем уровне. Архитектурой определяются принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативной памяти, внешних запоминающих и периферийных устройств. Классическую концепцию (принципы работы и принципиальная схема) построения компьютера предложил математик Джон фон Нейман [6, 7].

Перечислим самые распространенные архитектуры вычислительных машин:

– *Классическая архитектура* (архитектура фон Неймана) содержит только одно арифметико-логическое устройство, и одно устройство управления. Это самый распространенный пример однопроцессорного компьютера. К данной классификации относятся персональные компьютеры с общей шиной. Общая шина представляет собой многопроводниковое соединение, которое оснащено специальными гнездами для подключения необходимых электронных схем. Совокупность проводов общей шины можно разделить на шину адреса, шину данных и шину управления.

– *Многопроцессорная архитектура*. Данное вычислительное устройство содержит несколько отдельных процессоров с общей оперативной памятью и позволяет распараллеливать решаемые задачи за счет нескольких потоков данных и команд.

– *Многомашинная вычислительная система*. Данная архитектура также содержит несколько отдельных процессоров, но в отличие от предыдущей они не имеют общей оперативной памяти. У каждого процессора в данной системе своя локальная память. Собранные в такую систему компьютеры могут иметь классическую архитектуру.

– *Архитектура с параллельными процессорами.* В данной системе есть несколько арифметико-логических устройств, работающих под общим управлением (одно общее устройство управления). Такая конфигурация наиболее выгодна с точки зрения быстродействия для однотипных задач, которые необходимо выполнить на разных наборах данных.

БАЗОВАЯ КОНФИГУРАЦИЯ ПК

Рассмотрим базовую конфигурацию персонального компьютера (минимальный набор устройств, обеспечивающих его функционирование). Состав оборудования персонального компьютера можно гибко изменять и дополнять по мере необходимости. В типовую базовую архитектуру входят четыре устройства:

- системный блок;
- монитор;
- клавиатура;
- мышь.

Системный блок – это основное устройство компьютера, внутри которого установлены наиболее важные компоненты. Устройства, находящиеся внутри системного блока, называют внутренними, а устройства, подключаемые к нему снаружи, называют внешними. Внешние дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных, также называют периферийными. Внутри системного блока размещаются следующие узлы:

- материнская плата;
- жесткий диск;
- корпуса внешних запоминающих устройств (CD, DVD приводы);
- блок питания.

Материнская плата (системная плата) – это самая большая электронная плата в компьютере. Электронная плата представляет собой плоский кусок пластика, на котором укреплены электронные компоненты и различные разъемы. На материнской плате располагаются:

- микропроцессор;
- оперативное запоминающее устройство (ОЗУ);
- постоянное запоминающее устройство (ПЗУ);
- специальная память;
- контроллеры устройств;
- шина;
- разъемы (слоты).

Микропроцессор конструктивно представляет собой интегральную микросхему. Основные внутренние схемы процессора – арифметико-логическое устройство, внутренняя память (так называемые *регистры*), кэш-память (сверхоперативная память) и схемы управления всеми операциями и внешними шинами. Состав регистров процессора и их назначение называют архитектурой процессора. Разные регистры процессора имеют разное назначение. Регистры общего назначения используются для операций с данными. Адресные регистры содержат адреса, по которым процессор находит данные в памяти.

Рассмотрим основные задачи процессора:

- выполнение арифметических или логических операций, заложенных в схеме процессора в соответствии с выполняемой в данное время программой;
- оперативное управление устройствами вычислительной системы.

Процессор выполняет заложенные в нем функции практически на порядок быстрее, чем идет обмен информацией с остальными устройствами материнской платы. Во время обработки данных процессору необходимо получать их из оперативной памяти. Чтобы повысить эффективность работы процессора за счет сокращения обращений к ОЗУ внутри процессора создана буферная область – *кэш-память*. В нее записывается фрагмент обрабатываемой информации из ОЗУ и когда процессору нужны данные, он сначала обращается в кэш-память, и только если там нужных данных нет, происходит его обращение в оперативную память.

Нередко кэш-память распределяют по нескольким уровням. Кэш первого уровня выполняется в том же кристалле, что и сам процессор, и имеет объем порядка десятков Кбайт. Кэш второго уровня находится либо в кристалле процессора, либо в том же узле, что и процессор, хотя и выполняется на отдельном кристалле. Кэш-память первого и второго уровня работает на частоте, согласованной с частотой ядра процессора.

Кэш-память третьего уровня выполняют на быстродействующих микросхемах типа **SRAM** и размещают на материнской плате вблизи процессора. Ее объемы могут достигать нескольких Мбайт, но работает она на частоте материнской платы.

С другими устройствами, и в первую очередь с оперативной памятью, процессор связан группами проводников, которые называются шинами. Основных шин три:

- шина данных,
- адресная шина,
- командная шина.

Адресная шина. Данные, которые передаются по этой шине, трактуются как адреса ячеек оперативной памяти. Именно из этой шины процессор считывает адреса команд, которые необходимо выполнить, а также данные, с которыми оперируют команды. В современных процессорах адресная шина 32-разрядная, то есть она состоит из 32 параллельных проводников.

Шина данных. По этой шине происходит копирование данных из оперативной памяти в регистры процессора и наоборот. В компьютере с 64-разрядной шиной данных на обработку поступает сразу 8 байт данных.

Командная шина. По этой шине из оперативной памяти поступают команды, выполняемые процессором. Команды представлены в виде байтов.

Характеристики процессора:

– Производительность (количество элементарных операций, выполняемых им за одну секунду) определяет быстродействие компьютера

в целом. В свою очередь, производительность процессора зависит от двух других его характеристик – тактовой частоты и разрядности.

– Тактовая частота определяет число тактов работы процессора в секунду. Такты измеряются микросекундами, в течение которых может быть выполнена элементарная операция, например, сложение двух чисел. Тактовая частота измеряется в мегагерцах (1 МГц – миллион тактов в секунду).

– Разрядность процессора определяет размер данных в байтах, которые процессор может обработать за один такт. Эту характеристику часто называют машинным словом. Процессор в зависимости от его типа может иметь одновременный доступ к 32, 64 битам.

На материнской плате также может находиться математический сопроцессор. Он используется для ускорения выполнения операций над двоичными числами с плавающей запятой, над двоично-кодированными десятичными числами, для вычисления тригонометрических функций. Математический сопроцессор имеет свою систему команд и работает параллельно с основным микропроцессором, но под управлением последнего.

Процессор во время работы может напрямую обращаться к оперативной памяти (ОЗУ) и к постоянному запоминающему устройству (ПЗУ). ОЗУ, ПЗУ и кэш-память образуют внутреннюю (основную) память компьютера. Внешняя память – это устройства, предназначенные для длительного хранения информации компьютера (жесткий диск, флеш-карты).

Оперативная память – самое быстродействующее устройство для хранения данных. Она энергозависима, т.е. при выключении компьютера вся информация в ней очищается. ОЗУ хранит данные для активных на данный момент приложений. Главное достоинство оперативной памяти – ее высокое быстродействие и возможность прямого обращения к каждой адресуемой группе из восьми ячеек памяти отдельно (прямой адресный доступ к ячейке).

Оперативная память состоит из ячеек. В каждой ячейке может храниться 1 байт данных. У каждой ячейки есть свой адрес.

Оперативная память в компьютере размещается на стандартных панельках, называемых *модулями*. Модули оперативной памяти вставляют в соответствующие разъемы на материнской плате. Модули отличаются по размеру и количеству контактов.

Постоянное запоминающее устройство (ПЗУ) используется для хранения специальных программ, которые записываются в нее на этапе производства. Этот комплекс программ называется BIOS – базовая система ввода-вывода. В ПЗУ находятся:

- тестовые программы, проверяющие при каждом включении компьютера правильность работы его блоков;
- программы для управления основными периферийными устройствами – дисководом, монитором, клавиатурой;
- информация о том, где на диске расположена операционная система.

BIOS призвана обеспечивать выполнение основных аппаратных функций ПК вне зависимости от того, какого типа операционная система на нем установлена.

Энергонезависимая память CMOS. CMOS-память представляет собой микросхему, расположенную на системной плате компьютера. Эта память имеет низкое энергопотребление. В CMOS-памяти записаны:

- конфигурация аппаратных средств компьютера;
- количество и тип накопителей на магнитных дисках;
- тип видеоадаптера;
- информация о наличии сопроцессора.

CMOS-память питается от небольшой батарейки и поэтому ее содержимое не стирается даже после выключения питания компьютера. Микросхема CMOS-памяти также содержит обыкновенные электронные часы. Благодаря им в любой момент можно узнать текущую дату и время.

Генератор тактовых импульсов генерирует последовательность электрических импульсов. Промежуток времени между соседними

импульсами определяет время одного такта работы машины или просто такт работы компьютера. Частота ГТИ – одна из основных характеристик персонального компьютера и во многом определяет скорость его работы, так как каждая операция в машине выполняется за определенное количество тактов.

На материнской плате находятся электронные схемы (**контроллеры или адаптеры**), которые предназначены для обмена данными с соответствующим внешним устройством. На материнской плате находятся контроллер клавиатуры, контроллер жесткого диска и других внешних устройств.

Блок питания преобразует электропитание сети в постоянный ток низкого напряжения, который подается на микросхемы компьютера.

Жесткий диск (HDD – HardDiskDrive). Это устройство для долговременного хранения больших объемов данных и программ. Технически представляет собой устройство записи и хранения информации, основанное на принципе магнитной записи. Информация записывается на круглые диски, покрытые слоем ферромагнитного материала и расположенные на одной оси привода. Чтение и запись данных осуществляется с помощью считывающих головок,двигающихся над поверхностью дисков. Чтобы на пластины можно было записывать информацию, их поверхность разбивают на дорожки – концентрические кольцевые области. Все дорожки на всех пластинах жесткого диска, находящиеся на одинаковом расстоянии от центра, называются цилиндр. В свою очередь, дорожки делятся на равные отрезки, называемые секторами. Из объединения соседних секторов в группу образуется кластер.

К основным параметрам жестких дисков относятся емкость и производительность. Емкость – максимальное количество информации, которое может хранить жесткий диск. Производительность жесткого диска определяется скоростью передачи данных и временем произвольного доступа. Время произвольного доступа – это усредненное количество времени, которое требуется головке чтения-записи для позиционирования на произвольном участке пластины. Чем меньше данный показатель, тем

быстрее диск способен отдавать запрошенную информацию. Сегодня все жесткие диски имеют очень высокий показатель скорости передачи данных (до 30-60 Мбайт/с), и потому их производительность в первую очередь зависит от характеристик интерфейса, с помощью которого они связаны с материнской платой.

ВИДЕОПОДСИСТЕМА КОМПЬЮТЕРА

Монитор – это устройство визуального отображения информации. Он подключается к видеокарте, установленной в слот расширения системной платы. По принципу действия все мониторы делят на следующие группы: электронно-лучевые (CRT), жидкокристаллические (LCD), плазменные PDP (Plasms Display Panel), электролюминесцентные ELD (Electro Luminiscent Display), полевые эмиссионные FED (Field Emission Display), вакуумные флуоресцентные VFD (Vacuum Fluorescent Display), эмиссионные диоды LED (Light Emitting Diode).

Основными потребительскими параметрами мониторов являются: размер и шаг маски экрана, максимальная частота регенерации изображения, класс защиты.

Размер экрана монитора. Эта характеристика измеряется между противоположными углами экрана по диагонали в дюймах. Часть мониторов оснащена маской из вертикальных проволочек, что усиливает яркость и насыщенность изображения. Чем меньше шаг между отверстиями или щелями (*шаг маски*), тем четче и точнее полученное изображение. Шаг маски измеряют в долях миллиметра.

Частота регенерации (обновления) изображения показывает, сколько раз в течение секунды монитор может полностью сменить изображение (поэтому ее также называют *частотой кадров*). Частоту регенерации изображения измеряют в герцах (Гц). Чем она выше, тем четче и устойчивее изображение, тем меньше утомление глаз, тем больше времени можно работать с компьютером непрерывно.

Класс защиты монитора определяется стандартом, которому соответствует монитор с точки зрения требований техники безопасности.

Видеокарта (видеоадаптер). Она выполняет операции, связанные с управлением экраном. Это отдельная печатная плата (ее называют дочерней платой) помещаемая в разъем материнской платы. Основное назначение видеоадаптера – обработка графической информации.

За время существования персональных компьютеров сменилось несколько стандартов видеоадаптеров: MDA (монохромный); CGA (4 цвета); EGA (16 цветов); VGA (256 цветов). В настоящее время применяются видеоадаптеры SVGA, обеспечивающие по выбору воспроизведение до 16,7 миллионов оттенков с возможностью произвольного выбора разрешения экрана из стандартного ряда значений (640×480, 800×600, 1024×768, 1152×864; 1280×1024 точек и далее).

Разрешение экрана определяет количество точек экрана по вертикали и горизонтали и, следовательно, размер точки экрана. Чем выше данный показатель, тем меньше размер отдельного элемента изображения и, следовательно, выше качество отображаемой информации.

Цветовое разрешение (глубина цвета) определяет количество различных оттенков, которые может принимать отдельная точка экрана. Максимально возможное цветовое разрешение зависит от свойств видеоадаптера и, в первую очередь, от количества установленной на нем видеопамати. Кроме того, оно зависит и от установленного разрешения экрана.

КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программное обеспечение (ПО) – это совокупность программ для обработки информации с помощью компьютера. К этой категории также относят программы, необходимые для эксплуатации других программ и документацию, описывающую функционирование системы обработки данных [4, 12].

В компьютерном сленге часто используется слово «софт» от английского слова software, которое в этом смысле впервые применил в статье в American Mathematical Monthly математик из Принстонского университета Джон Тьюки в 1958 году.

Рассмотрим несколько типов классификации ПО.

1. **По назначению** программное обеспечение можно разделить на системное, прикладное и инструментальное.

2. **По способу распространения и использования** можно разделить на несвободное (закрытое), открытое и свободное.

Системное программное обеспечение – это комплекс программ, которые обеспечивают эффективное управление компонентами вычислительной системы, такими как процессор, оперативная память, каналы ввода-вывода, сетевое оборудование, выступая как «межслойный интерфейс», с одной стороны которого аппаратура, а с другой приложения пользователя. Основное назначение системного ПО – обеспечить удобный интерфейс пользователя и взаимосвязанную работу других программ, а также эффективное управление основными ресурсами вычислительной системы.

Прикладное программное обеспечение (англ. Application software) – это программное обеспечение, предназначенное для решения конкретных задач пользователя. К нему можно отнести:

- отдельные прикладные программы и пакеты прикладных программ, предназначенные для решения различных задач пользователей;
- автоматизированные системы, созданные на основе этих прикладных программ.

Инструментальное программное обеспечение – программное обеспечение, предназначенное для использования в ходе проектирования, разработки и сопровождения программ. Инструментальное программное обеспечение предназначено для разработки новых программ и программных комплексов.

Рассмотрим состав системного программного обеспечения. К этой категории относятся: программы BIOS, операционная система, сервисное программное обеспечение.

BIOS (базовая система ввода-вывода) – это встроенное программное обеспечение. Вообще, встроенное ПО (built-in software)

представляет собой компьютерные программы, которые заложены в компьютер и автоматически выполняются при его включении. BIOS обычно хранится в чипе флэш-памяти на материнской плате, поставляется вместе с материнской платой и позволяет производить изначальную настройку компьютера и подготовить его к запуску операционной системы. При включении компьютера микропроцессору необходимо взять первую инструкцию для работы. Ее нельзя получить из операционной системы, поскольку последняя находится на жестком диске. В BIOS хранится инструкция по обращению к операционной системе и ее загрузке в оперативную память. Вот еще некоторые стандартные задачи, которые решает BIOS:

- Самотестирование компьютера при включении питания для всех аппаратных компонентов системы с целью проверки правильного функционирования каждого из них;
- Загрузка операционной системы с жесткого диска;
- Выполнение ряда операций низкого уровня, с помощью которых операционная система связывается с аппаратными средствами компьютера. Эти операции предусматривают управление таким оборудованием, как клавиатура, экран, последовательные и параллельные порты;
- Настройка и конфигурирование аппаратных средств и системных ресурсов;
- Обслуживание аппаратных прерываний от системных устройств (таймера, клавиатуры, дисков);
- Обработка базовых функций программных обращений (сервисов) к системным устройствам.

Операционная система (ОС) – комплекс программ, создающих интерфейс пользователя, а также обеспечивающий управление ресурсами системы, выполняющие загрузку и выполнение прикладных программ.

К основным функциям ОС относятся:

- Загрузка пользовательских программ в оперативную память и их выполнение;
- Обеспечение доступа к устройствам ввода и вывода информации;

- Эффективное управление оперативной памятью, разделение доступной памяти между работающими программами;
- Обеспечение доступа к жесткому диску и другим внешним носителям информации;
- Реализация дружественного пользовательского интерфейса;
- Обеспечение параллельного выполнения задач;
- Обмен данными между активными процессами, синхронизация их работы;
- Защита самой системы, а также пользовательских данных и программ от действий пользователей (злонамеренных или по незнанию) или приложений;
- Разграничение прав доступа и многопользовательский режим работы (аутентификация, авторизация).

Рассмотрим область применения операционных систем. Есть виды вычислительной техники, для которых операционные системы не нужны. Например, встроенные микрокомпьютеры содержатся сегодня во многих бытовых приборах, автомобилях. В данных устройствах после их включения выполняется одна последовательность действий, нет обмена данными и нет необходимости сохранения промежуточных результатов и текущего состояния системы.

Можно назвать следующие причины, по которым устройствам требуется операционная система:

- Необходимо сохранение данных в процессе работы и обмен ими между различными задачами. Для этого используется файловая система – составная часть любой ОС. Современные ОС, кроме того, предоставляют возможность непосредственно «связать» вывод одной программы с вводом другой, выполняя межпрограммный интерфейс;
- В процессе работы необходимо связываться с устройствами ввода и вывода информации. В современных ОС для этого используются системные библиотеки, имеющие набор стандартных функций ввода и вывода информации со стандартных устройств;

– Необходимо разделение прав доступа к информации между различными типами пользователей для защиты данных от несанкционированного доступа или случайного уничтожения;

– Необходимо разделение времени при исполнении нескольких программ на одном компьютере (даже содержащем лишь один процессор). При этом специальный компонент, называемый планировщиком, делит процессорное время на короткие отрезки и предоставляет их поочередно различным исполняющимся программам (процессам);

– Необходимо управление процессами выполнения отдельных программ. Для этого используются специальные утилиты, которые являются частью операционной системы.

Многозадачность и распределение полномочий в современных операционных системах требуют определенной иерархии компонентов самой операционной системы. В составе ОС можно выделить следующие основные группы компонентов:

- ядро, содержащее планировщик;
- драйверы устройств, непосредственно управляющие оборудованием;
- файловая система;
- системные библиотеки;
- оболочка с утилитами.

Большинство программ, как системных, так и прикладных, исполняются в непривилегированном («пользовательском») режиме работы процессора и получают доступ к оборудованию только посредством системных вызовов. Ядро исполняется в привилегированном режиме. Ядро – центральная часть операционной системы, управляющая выполнением процессов, ресурсами вычислительной системы и предоставляющая процессам координированный доступ к этим ресурсам. Основными ресурсами являются процессорное время, оперативная память и устройства ввода-вывода.

Сервисное программное обеспечение – это совокупность программных средств, расширяющих и дополняющих возможности операционных систем.

По функциональному назначению служебные программы можно разделить на следующие группы:

- программы-менеджеры для улучшения пользовательского интерфейса;
- защита данных от случайного удаления и несанкционированного доступа;
- программы резервного копирования, используемые для восстановления информации;
- программы обслуживания дисков (форматирование, очистка от временных файлов, дефрагментирование и т.п.):
- средства архивации и разархивации;
- антивирусные средства.

По способу организации и реализации сервисные средства могут быть представлены: оболочками, утилитами и автономными программами. Утилиты и автономные программы имеют узкоспециализированное назначение и выполняют каждая свою функцию. Оболочки предоставляют пользователю качественно новый интерфейс и освобождают его от детального знания операции и команд ОС.

Примером автономных программ являются программы технического обслуживания. Это совокупность программно-аппаратных средств для диагностики и обнаружения ошибок в процессе работы компьютера или вычислительной системы в целом. К ним относятся:

- средства диагностики и тестового контроля правильности работы компьютера и ее отдельных частей, в том числе автоматического поиска ошибок и неисправностей;
- специальные программы диагностики и контроля вычислительной среды информационной системы в целом, в том числе программно-аппаратный контроль, осуществляющий автоматическую

проверку работоспособности системы обработки данных перед началом работы вычислительной системы в очередную производственную смену.

Инструментальное программное обеспечение. Оно используется для создания новых приложений на определенном языке программирования. Язык программирования – это формализованный язык описания алгоритмов, используемых для решения различных задач на компьютере.

Со времени создания первых компьютеров человечество придумало уже более восьми с половиной тысяч языков программирования. Современные прикладные и системные программы, в том числе операционные системы и системы программирования, в основном разрабатываются на языках высокого уровня, которые обеспечивают удобство и высокую производительность работы программиста. Интегрированные среды программирования включают весь набор средств для комплексного их применения на всех технологических этапах разработки программ. Основное назначение такого инструментария состоит в том, чтобы с его помощью повысить производительность и эффективность труда программистов.

В состав системы программирования обязательно входит компилятор, который переводит исходную программу с языка программирования высокого уровня в некоторую промежуточную форму, называемую *объектным кодом*. После компиляции в объектный код выполняется еще один этап, во время которого из объектных кодов отдельных частей программы собирается вся программа в кодах целиком. На этом же этапе, называемом *редактированием связей* или *сборкой*, происходит подключение стандартных подпрограмм, если они нужны, и преобразование оставшихся имен в адреса.

Рассмотрим типовой состав системы программирования для языка высокого уровня:

– *Текстовый редактор*. Программа для ввода и редактирования исходного текста программы.

– *Компилятор*. Программа, преобразующая исходный текст программы в выполняемый код. По способу генерации кода компиляторы делятся на обычные и оптимизирующие. Оптимизирующие компиляторы работают гораздо дольше обычных, но генерируют код, оптимизированный по скорости или по объему.

– *Компоновщик или редактор связей (linker)*. Программа, собирающая из совокупности объектных файлов и библиотек исполняемую программу.

– *Отладчик (debugger)*. Программа, позволяющая контролировать код и данные программы во время ее выполнения с целью обнаружения ошибок.

– *Профилировщик (profiler)*. Программа, предназначенная для оценки эффективности выполнения различных участков кода. Профилировщики встраивают в программу специальный код, позволяющий подсчитать частоту вызова различных подпрограмм, время их работы с учетом и без учета вызываемых подпрограмм и большое количество других параметров.

– *Библиотека (library)*. Это набор стандартных данных и подпрограмм. Библиотеки по способу использования делятся на статические (static library) и динамически связываемые (dynamic link library – DLL). Статические используются только при компоновке программы, при которой из них извлекаются используемые модули. Динамические библиотеки постоянно находятся в памяти, вследствие чего их модулями могут пользоваться различные программы одновременно.

В настоящее время многие интегрированные среды поддерживают популярную технологию визуального программирования – так называемую технологию *быстрой разработки приложений* (Rapid Application Development – RAD). Эта технология позволяет быстро создавать и изменять графический интерфейс пользователя. Поскольку программирование графического интерфейса пользователя является достаточно объемной и сложной задачей, то путем сокращения затрат на его создание уменьшаются затраты на разработку приложения в целом.

По способу распространения можно выделить следующие виды программного обеспечения:

- **Коммерческое программное обеспечение.** Это программное обеспечение, созданное коммерческой организацией и распространяемое на платной основе.

- **Условно-бесплатное программное обеспечение (Shareware).** Это программное обеспечение, имеющее ограничения на его использование. Такая программа предоставляется пользователям бесплатно на определенное время. Пользователь может использовать программу, тестировать ее, осваивать ее возможности, но по истечении этого срока он обязан либо купить ее, либо удалить со своего компьютера.

- **Бесплатное программное обеспечение (Freeware).** Это бесплатное программное обеспечение, распространяемое без исходных кодов.

- **Свободное программное обеспечение (Free Software).** Это программное обеспечение, которое распространяется вместе со своими исходными кодами. В отношении свободного ПО пользователь обладает «четырьмя свободами», которые сформулировал Ричард Столлман в 1970-х годах: **запускать, изучать, распространять и улучшать** программу.

- **Шпионское программное обеспечение (Spyware).** Это программное обеспечение обычно распространяется вместе с другим полезным ПО и занимается сбором информации на компьютере пользователя и отсылкой ее создателю.

- **Adware** – вид программного обеспечения, при использовании которого пользователю принудительно показывается реклама. Как правило, средства, полученные от рекламы, идут на дальнейшее развитие этого ПО.

- **Careware.** Это вид условно-бесплатного программного обеспечения shareware. Автор данного вида ПО требует, чтобы оплата за него шла на благотворительность.

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

ЛАБОРАТОРНАЯ РАБОТА 1. РАБОТА С СИСТЕМАМИ СЧИСЛЕНИЯ

Задано число в десятичной системе счисления. Перевести его в двоичную, восьмиричную и шестнадцатиричную системы счисления. Проверить правильность выполнения действий переводом всех полученных чисел обратно в десятичную систему счисления.

Задано число в двоичной системе счисления, перевести его в восьмиричную и шестнадцатеричную системы счисления. Проверить правильность выполнения действий переводом всех чисел в десятичную систему счисления.

$$\text{ЧИСЛО} = \text{Номер варианта} * 2 + 210$$

ЛАБОРАТОРНАЯ РАБОТА 2. ДОПОЛНИТЕЛЬНЫЙ КОД ЧИСЛА

Задано отрицательное число (ЧИСЛО) в десятичной системе счисления. Получить его обратный и дополнительный код в байтовом представлении. Выполнить над числами в двоичной системе счисления следующие действия:

250 - ЧИСЛО

29 - ЧИСЛО

Предварительно перевести числа 250 и 29 в двоичную систему счисления.

Проверить правильность выполнения действий переводом результата в десятичную систему счисления.

$$\text{ЧИСЛО} = \text{Номер варианта} + 210$$

ЛАБОРАТОРНАЯ РАБОТА 3. ВВОД И РЕДАКТИРОВАНИЕ В ТЕКСТОВОМ ПРОЦЕССОРЕ WRITER (OPENOFFICE.ORG)

Загрузите текстовый процессор Writer (OpenOffice.org), используя Пуск / Программы / Writer (OpenOffice.org).

Загрузите *справочную систему*. Ознакомьтесь со справочной системой: ее структура такая же, как у справки к операционной системе Windows.

Рассмотрите элементы окна Writer. С помощью меню *Вид* установите *режим отображения* – *Разметка страницы*, убедитесь в наличии *линеек* (при необходимости подключите их, используя меню *Вид*). Опробуйте другие режимы отображения как с помощью меню *Вид*, так и с помощью кнопок в левой нижней части окна: назначение каждой кнопки можно определить по всплывающей подсказке. Проверьте наличие двух панелей – *Стандартной* и *Форматирования*. Если установлены другие панели, отключите их (*Вид / Панели инструментов*). Установите *масштаб* изображения 100% (*Вид / Масштаб / 100% / ОК* или поле *Масштаб* с раскрывающимся списком в стандартной панели инструментов).

Параметры страницы настраиваются с помощью меню *Формат / Страница*, где могут быть установлены размер листа и его ориентация, поля и другие параметры. Используя соответствующие вкладки диалогового окна *Параметры страницы*, установите размер бумаги А4, ориентация – книжная, поля – по 20 мм со всех сторон, затем нажмите ОК.

Набор текста

В панели форматирования проверьте установки: *стиль* – Обычный, *шрифт (гарнитура)* – TimesNewRoman, *размер* шрифта – 10. В рабочей области введите приведенный ниже текст, используя соответствующие кнопки выравнивания по центру (для обращения), по ширине (для основной части текста), по правому краю (фамилия). Печать точки при русской раскладке – нажатием крайней правой клавиши в нижнем ряду

(знак «?»), запятой – эта же клавиша при нажатой **Shift**. Для пропуска строки используйте клавишу ввода **Enter**.

Для *удаления* ошибочно введенного текста используйте клавиши удаления **Backspace** или **Delete**. Как правило, текстовый процессор работает в *режиме вставки*, когда вновь вводимый в текст символ раздвигает уже введенные символы. В *режиме замены* вновь вводимые символы замещают ранее введенные. Для включения или отключения режима замены используется двойной щелчок по индикатору **ЗАМ** в строке состояния.

Writer позволяет автоматизировать некоторые операции при наборе текста [3, 9]]. Многократно повторяющиеся слова можно сохранить в виде *автотекста*, и они будут вводиться затем по первым символам или специальному обозначению. Добавьте, например, в автотекст слово «Информатика». Для этого выполните следующие действия: напечатайте в тексте слово «Информатика» и скопируйте его в буфер обмена, затем выберите меню *Правка \ Автотекст*, введите из буфера обмена в панель автотекста – скопированное слово «Информатика», убедитесь что помечена категория *стандартная* (или, в зависимости от версии, *Мой автотекст*), щелкните на кнопку *автотекст*, выберите во всплывающем меню – *создать текст*, затем щелкните по кнопке *добавить*. Теперь начните ввод этого слова в набираемом тексте – компьютер предложит автоматически вставить соответствующее слово; для вставки достаточно нажать клавишу **Enter**.

Подобным образом Writer исправляет и некоторые типичные ошибки: производит *автозамену*. Например, неправильно введенное слово «бюллетень» будет автоматически заменено на правильное – «бюллетень». Настроить автозамену можно используя меню *Сервис / Параметры автозамены / Заменить*.

Настройте замену символов **ИИИ** на текст «Иванов Иван Иванович». В меню *автозамены* просмотрите список таких исправлений, внести в него новые слова.

Скопируйте в новый файл текст (не обращая внимание на границы абзацев, они будут изменены далее):

Уважаемый Иван Петрович!

Извещаю Вас, что 10 марта в 10.00 в конференц-зале открытого акционерного общества «Интеграл» состоится очередное собрание кредиторов с повесткой дня:

Утверждение изменений к плану внешнего управления ОАО «Интеграл».

Разное.

При себе необходимо иметь:

паспорт или удостоверение личности

доверенность

акт сверки счетов Вашего предприятия с ОАО «Интеграл»

Внешний управляющий ОАО «Интеграл»

А.А.Александров

При редактировании документа иногда надо ясно видеть, где нажата клавиша **Enter**, сколько раз нажата клавиша пробела и т.п. – то есть увидеть специальные непечатаемые символы. Для этого используется кнопка *Непечатаемые символы* в стандартной панели инструментов (¶). Нажмите эту кнопку и просмотрите обозначения и расстановку таких символов. Отключите кнопку *Непечатаемые символы*.

Форматирование текста

Выделение фрагмента текста, как уже отмечалось, производится протаскиванием мышью. Для *выделения отдельной строки* делаются щелчки слева от строки, *отдельного слова* – двойной щелчок по первой букве этого слова. *Выделить абзац* целиком можно щелчками слева от начала абзаца. Опробуйте все эти методы выделения фрагментов.

Выделенный фрагмент может быть *перенесен* или *скопирован* через буфер обмена как в любое место активного документа, так и в другой документ. Используется любой из ранее рассмотренных методов: кнопки *Вырезать*, *Копировать*, *Вставить*; соответствующие команды

контекстного меню или меню *Правка*; сочетания клавиш клавиатуры: **Ctrl+X** – вырезать, **Ctrl+C** – скопировать в буфер, **Ctrl+V** – вставить из буфера. При переносе фрагмента на небольшое расстояние более удобен другой способ: поместите указатель мыши в выделенную область (вид указателя – стрелка) и перетащите фрагмент левой кнопкой мыши в нужное место (указатель изменит вид – теперь это стрелка с прямоугольником внизу). Подобным образом фрагмент можно скопировать: те же действия производятся при нажатой клавише **Ctrl**.

Выделенный фрагмент может быть *удален* нажатием клавиши удаления **Delete**. Скопируйте путем перетаскивания фрагмент второго предложения «в 10.00» в любое другое место этой же строки. Удалите скопированный фрагмент.

Выделите весь текст. Измените размер шрифта на 12.

Для установки левой и правой *границ текста* используйте маркеры на линейке. Для установки *абзацного отступа* (отступа первой строки в абзаце) можно нажать клавишу табуляции **Tab**. Величину отступов можно отрегулировать соответствующим маркером на горизонтальной линейке. Также для точной настройки размера абзацного отступа можно использовать меню *Формат / Абзац* вкладка – *Отступы и интервалы*. Установите отступ первой строки абзаца в 2 см.

Установите межстрочный интервал, используя меню *Формат / Абзац* вкладка – *Отступы и интервалы*, на значение *полуторный*. Просмотрите, какие еще виды междустрочных интервалов Вам доступны. Задать междустрочный интервал можно также соответствующими кнопками на панели инструментов *форматирование*. Найдите эти кнопки, выделите абзац текста и опробуйте разные значения междустрочного интервала.

Просмотрите разные способы выравнивания текста в абзаце: по левому краю, по правому краю, по центру, по ширине, используя соответствующие кнопки на панели форматирования или меню настроек абзаца.

Для точной настройки параметров абзаца можно использовать меню *Формат / Абзац* вкладка – *Отступы и интервалы*. Найдите на вкладке

настройку выравнивания текста, его стиль (основной текст); отступа установленные для указанного абзаца:

- *Слева*. Это отступ всего абзаца влево от границы текста;
- *Справа*. Это отступ всего абзаца вправо от границы текста;
- *Перед*. Это отступ всего абзаца вниз от предыдущего абзаца;
- *После*. Это отступ всего абзаца вверх от последующего абзаца.

Выделите абзац, начинающийся со слов «Извещаю Вас» и установите отступ слева – 1 см, отступ справа – 1 см, интервал перед абзацем – 1 см, интервал после абзаца – 1 см. Посмотрите, как изменился текст.

Выделите в виде фрагмента первую строку и оформите другим шрифтом и размером шрифта, используя панели инструментов и контекстное меню. Задайте для данной строки разреженный шрифт, для этого используйте меню *Формат / Символ*. Просмотрите все возможные настройки, отраженные на вкладках этого меню и найдите настройку междусимвольного интервала. Установите *разреженный* на 3 пункта (пт.) шрифт. Посмотрите, как изменилось написание строки. Попробуйте установить уплотненный тип шрифта.

Для **отмены** ошибочно выполненных действий используйте кнопку *Отменить* в стандартной панели инструментов: она отменяет последнее выполненное действие. При необходимости отмены нескольких действий они могут быть выделены в списке, открываемом с помощью стрелки справа от кнопки *Отменить* и отменены однократным нажатием этой кнопки. Ошибочно отмененная операция может быть восстановлена с помощью кнопки *Вернуть*. Перенесите первое предложение документа в любое место перетаскиванием мышью. Отмените это действие с помощью кнопки *Отменить*.

Операции поиска и замены

Предположим, что в названии организации сделана ошибка: ОАО называется не «Интеграл», а, например, «Интервал». Как внести изменения, не перепечатывая каждый раз ошибочно набранное слово?

Воспользуйтесь меню *Правка / Заменить*. В диалоговом окне наберите в поле *Найти* слово «Интеграл», а в поле *Заменить* – слово «Интервал» (для активизации нужного поля щелкните по нему мышью). Нажмите кнопку *Найти* далее, затем – *Заменить*. Если Вы уверены, что надо заменить все, нажмите соответствующую кнопку.

Упражнения для самостоятельной работы

Просмотрите еще раз настройки меню АБЗАЦ и СИМВОЛ, вызвав их с помощью контекстного меню или из меню окна ФОРМАТ. Опробуйте все возможные в них настройки и научитесь оформлять с помощью них текст.

Создайте новый документ (кнопка *Создать*) с параметрами страницы – формат А4, ориентация – книжная, поля слева и справа по 3 см, сверху и снизу – по 2 см; скопируйте в него абзац текста. Сделайте две копии этого абзаца в этом же документе. В первой копии измените тип шрифта (гарнитуру) и его размер. Установите разреженный интервал между символами. Увеличьте отступы слева и справа. Во второй копии измените одинарный межстрочный интервал на двойной. Установите интервал перед вторым абзацем 1 см, после него – 2 см.

Настройте автотекст на выбранное вами слово.

Настройте автозамену на выбранную вами фразу (например, «ААА» заменяется на «С добрым утром»).

Опробуйте операцию поиска и замены.

Сохраните созданный документ и покажите преподавателю.

Списки

Наиболее просто оформление списка производится с помощью кнопок на панели инструментов форматирования. Наберите каждый раз с новой строки название дней недели и оформите их в виде **нумерованного списка**. Для этого щелкните по кнопке *Нумерованный список* в панели форматирования. После этого для добавления нового элемента в список

достаточно после последнего элемента нажать переход на новую строку (клавиша Enter).

Любой список необходимо завершать двойным нажатием клавиши Enter, иначе установки списка будут распространяться на следующий текст.

Следующий перечень фраз (допустим названия изучаемых дисциплин), оформите в виде **маркированного списка**: выделите соответствующие строчки, и щелкните по кнопке *Маркеры*. Если Вас не устраивает вид маркеров, их можно изменить с помощью меню *Формат / Маркеры и нумерация* (или аналогичного контекстного меню) и выбрать один из имеющихся вариантов. С помощью вкладки **Изображения** можно выбрать и другие маркеры. Отступы списка можно отрегулировать с помощью кнопок *Уменьшить / Увеличить отступ*, которые располагаются на панели инструментов *Списки* (или *Маркеры и нумерация*). Эта панель появляется автоматически при работе со списками, но если она не отображается, ее можно вызвать с помощью меню *Вид / Панели инструментов*.

Используя это же меню, можно организовать и **многоуровневый список**. Для этого в меню *Формат / Список* настраивается вид уровней создаваемого списка, вид их нумерации, отступы и другие параметры. Далее создаете обычный нумерованный список, где вся нумерация получится одного уровня. Далее с помощью кнопок *Уменьшить / Увеличить отступ* (или, в зависимости от версии, *Перейти на уровень выше / Перейти на уровень ниже*) переходим к нужному уровню в каждом пункте.

Для внесения изменений в списки можно использовать панель *Маркеры и нумерация*.

Предположим, что в последовательности автоматически нумерованных абзацев Вам нужен один абзац без нумерации, как в примере:

- 1) первый абзац
- 2) второй абзац
- третий абзац без нумерации

3) четвертый абзац

Это можно сделать следующими способами:

Способ 1: нумеруем все абзацы без исключения; затем помещаем курсор перед третьим абзацем и нажимаем клавишу <Backspace>, для стирания одного символа справа. Абзац теряет свой номер.

Способ 2: нумеруем все абзацы без исключения; затем помещаем курсор перед третьим абзацем и щелкаем по значку Нумерация в панели инструментов. Абзац теряет свой номер и отступ текста.

Способ 3: пронумеруем абзацы 1 и 2 с помощью значка *Нумерация* в панели инструментов. Далее переходим к абзацу 4, выбираем *Формат – Маркеры и нумерация* – вкладка *Настройки* и в счетчике начального значения задаем число 3.

Создайте три списка и в каждом испытайте каждый из способов исключения элемента из общего списка.

Сохраните созданный документ.

Применение стилей

Оформлять каждый абзац можно по отдельности, но для большого документа это занимает много времени. Поэтому все современные офисные пакеты упрощают эту процедуру за счет использования стилей. Стиль абзаца определяет форматирование как самого абзаца (отступы, интервалы, выравнивание и т.д.), так и форматирование знаков в абзаце (шрифт, размер и т.д.).

Чтобы задать стиль абзаца, установите на него курсор и щелкните мышкой на кнопке *Стили и Форматирование* в панели инструментов или выбрать в меню *Формат / Стили*. Далее выбирается из списка нужный стиль.

Для изменения уже существующего стиля необходимо выбрать его в окне кнопки *Стили и Форматирование*, щелкнуть правой кнопкой мыши и выбрать меню *Изменить*.

Для создания своего стиля абзаца выберите в тексте сформированный по Вашим требованиям текст. Откройте меню стили и форматирование и щелкните по кнопке *Создать стиль из выделенного*.

Скопируйте в файл несколько абзацев текста и, выделяя их по очереди, примените к ним созданный вами стиль.

Упражнения для самостоятельной работы

Создайте в документе три списка: маркированный, нумерованный и многоуровневый. Сохраните документ под именем Списки.

Создайте два собственных стиля оформления абзаца и продемонстрируйте его применение.

ЛАБОРАТОРНАЯ РАБОТА 4. ВСТРАИВАНИЕ ОБЪЕКТОВ В ДОКУМЕНТ

Работа с таблицами

Для встраивания таблицы выберите меню *Таблица / Вставить таблицу*, в диалоговом окне установите количество строк – 6 и столбцов – 4 (при необходимости в дальнейшем их количество можно будет изменить). Более удобный вариант встраивания небольшой таблицы – использование кнопки *Добавить таблицу* в стандартной панели инструментов: размер таблицы задается протаскиванием мыши.

Подготовьте в таблице расписание занятий: 6 дней по 4 пары. Добавьте строку для указания номера пары (курсор установите в любую ячейку таблицы, используйте меню *Таблица / Вставить строку*). Добавьте столбец для указания дня недели: выделите мышью первый столбец таблицы, выберите в меню *Таблица / Разбить ячейки*, задайте значения: число столбцов – 2, число строк – 7. Получили 5 столбцов (выделенный столбец разбит на два). Установите нужную ширину столбцов: это можно сделать по-разному. Щелкните по таблице, и обратите внимание на появление маркеров на горизонтальной линейке. Перетаскивая маркеры на горизонтальной линейке, измените ширину столбцов. Часть столбцов (кроме первого) должна иметь одинаковую ширину: выделите эти столбцы, и в меню *Таблица* воспользуйтесь командой *Выровнять ширину столбцов*.

Введите в таблицу расписание занятий. Для ввода повторяющихся наименований используйте автотекст или копирование. Выберите подходящие размеры и начертания шрифтов, выравнивание текста в ячейках; при этом выделите наименование дней недели и нумерацию пар.

Для оформления таблицы воспользуйтесь панелью инструментов *Таблица*, выберите толщину и тип линии для наружного контура таблицы, для этого щелкните по кнопке *Внешние границы* или воспользуйтесь вызовом меню *Таблица / Свойства таблицы*. Выберите линии для внутренних границ, и установите их, выбирая из списка *Внешние границы* нужное положение линии.

Выделите последний столбец созданной таблицы и добавьте столбец. Выделите появившийся новый столбец, в меню *Таблица* выберите команду *Объединить ячейки*. В полученную объединенную ячейку впишите Вашу фамилию и имя. Отредактируйте их (поверните на 90 градусов), используя панель меню *Формат / Символ*. Расположите фамилию и имя по центру ячейки, используя в контекстном меню строку *Ячейка*. При необходимости уменьшите ширину этой ячейки перетаскиванием мышью за правую границу таблицы. Сохраните документ.

Создание рисунков и схем

Рисунок может быть создан с использованием графического редактора и встроен в документ. Несложные рисунки и схемы часто удобнее подготовить, используя встроенные средства текстового процессора. Подключите панель инструментов для рисования *Вид / Панели инструментов / Рисование* (или кнопка *Рисование* в стандартной панели инструментов).

Изобразите структурную схему вашего факультета. Выберите в панели рисования скругленный прямоугольник, протащите указатель мыши от левого верхнего до правого нижнего угла изображаемого прямоугольника, отпустите кнопку мыши (при этом нарисованная фигура остается выделенной: по ее углам и серединам сторон – квадратики-маркеры), расположите его вверху листа по центру документа. Для редактирования линии контура прямоугольника выделите его и щелкните по кнопке *Тип линии*: выберите нужный тип. Для выполнения надписи выполните щелчок левой кнопкой мыши по фигуре – в ней появится курсор-указатель текста. Наберите название своего факультета. Выделите текст надписи и отформатируйте ее: сцентрируйте, выберите подходящий тип и размер шрифта.

Ниже скругленного прямоугольника по аналогии изобразите два одинаковых овала (можно воспользоваться непосредственно кнопкой *Овал* в панели *Рисование*): в одном введите надпись «Направления и специальности», в другом – «Кафедры». Между скругленным

прямоугольником и каждым из овалов проведите стрелки: нажмите кнопку *Стрелка* и протащите мышь от начала стрелки у скругленного прямоугольника до ее конца у овала. Для форматирования стрелки используйте кнопку *Тип линии*. При рисовании стрелки автоматически появляется панель меню с ее настройками (обычно под панелью форматирования), изучите настройки стрелок.

При ошибочно выполненном действии используйте кнопку *Отменить*; выделенный объект может быть удален нажатием клавиши **Delete**.

Нарисуйте еще две схемы с придуманной вами информацией, используйте как можно больше разных элементов, а также объемные и обычные стрелки. Сохраните созданные схемы.

Методы встраивания объектов, подготовленных в других приложениях

В текст документа можно вставить целиком готовый ранее файл, например, другой текстовый документ. Для этого используется меню *Вставка / Файл*. В новом документе наберите или скопируйте в него абзац текста. Затем введите фразу «*далее идет вставка из другого файла*» и вставьте текст из другого созданного вами файла.

Предположим, что ранее созданный в графическом редакторе Paint рисунок в файле «Мой рисунок.bmp» или его часть надо вставить в текстовый документ процессора Writer. Графический редактор при этом выступает в роли *сервера* – приложения, в котором готовится объект, а текстовый процессор – *клиента*, который использует объект. По технологии **OLE** объект может быть встроен, как *внедренный* (являющийся частью документа Writer и не поддерживающий связь с сервером; при этом размер документа существенно увеличивается).

В созданном вами документе наберите заголовок «Внедрение объекта». Для вставки всего рисунка в виде файла выберите в меню *Вставка / Объект / Объект OLE*. Используя кнопку *Обзор*, найдите файл со своим рисунком и щелкните ОК.

Далее в этом же документе наберите заголовок «Встраивание рисунка из буфера обмена». В графическом файле «Мой рисунок» выделите часть изображения (вспомните, как это делается) и скопируйте ее в буфер. Для вставки используйте меню *Вставка*. Можно воспользоваться меню *Вставка / Изображение / Из файла*.

Сохраните полученный документ.

Настройка изображения

Встроенный рисунок может быть перемещен: после выделения щелчком мыши он перетаскивается в любое место документа. Для изменения размеров рисунка перетаскивается одна из его границ за соответствующий квадратик (*маркер изменения размера*). Опробуйте эти действия на встроенном рисунке.

Различные методы настройки изображения возможны с использованием панели инструментов Изображения или контекстного меню. Установите эту панель и опробуйте возможности ее инструментов: изменение контрастности и яркости. Наиболее важным инструментом настройки изображения является кнопка, обеспечивающая *обтекание* встроенного в документ рисунка текстом: сверху и снизу рисунка, вокруг рамки, по контуру и т.д. Введите любой текст, наложите на него встроенный ранее рисунок и опробуйте различные варианты обтекания. При необходимости используйте контекстное меню *Обтекание*.

Редактирование формул

Для ввода простых формул бывает достаточно возможностей меню *Стиль / Нижний* или *Верхний индекс* (степень) или *Формат / Символ / Положение*. В новом документе введите формулу $y_i \neq \gamma + \beta x^3$ (Для ввода знака неравенства и греческих букв используйте вставку специальных символов: щелкните мышью по соответствующему символу, затем по кнопке *Вставить*; после ввода всех необходимых символов щелкните *Заккрыть*. Для ввода индекса и степени нажмите клавиши нужных букв на

клавиатуре, выделите букву и установите *Нижний индекс* или *Верхний индекс*. Переменные x и y выделите курсивом.

Используя описанные выше приемы, введите следующие символы и формулы:

$$\S \textcircled{C} \textcircled{R} \div \leq \geq \infty \pi$$

$$c^2 = a^2 + b^2$$

$$y_i = x_i - \beta$$

Для ввода более сложных формул используется специальный **формульный редактор** (меню *Вставка / Объект / Формула*). Изучите меню редактора формул. Если меню шаблонов формул не открылось, вызвать его можно с помощью меню *Вид / Элементы*. Введите зависимость для функции распределения нормального закона

$$F(z) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{(x-m)^2}{2\sigma^2}} dx$$

Для этого загрузите формульный редактор и введите с клавиатуры обычным способом левую часть формулы $F(z) =$. Для ввода дроби перед интегралом воспользуйтесь *Шаблонами дробей и радикалов*, для ввода греческих букв используйте кнопку строчных греческих букв. Подобным образом выберите подходящие шаблоны для интеграла, степеней, символа бесконечность. Для переходов между полями формулы используйте мышь. При необходимости переопределите стиль, размеры символов. Для завершения ввода формулы щелкните вне ее. При необходимости редактирования формулы дважды щелкните по ней.

Введите с помощью редактора еще несколько формул:

$$f(X, \vec{\beta}) = \sum_{j=0}^{p-1} \vec{\beta}_j \cdot \varphi_j(X)$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Автономмерация объектов

К объектам, которые требуют нумерации, относятся: рисунки, формулы и таблицы. Рассмотрим сквозную нумерацию рисунков в тексте. В контекстном меню вставленного рисунка выбираем пункт *Название*. Стилль нумерации и разделитель устанавливаем в соответствии с требованиями по оформлению документа. После этого рисунок автоматически получает очередной номер. Для ссылки в тексте на рисунок пишем ссылку типа «как показано на рисунке », выбираем команду главного меню *Вставить / Перекрестная ссылка*. В диалоговом окне выбираем тип поля *Рисунки* и нужный нам рисунок. После нажатия на кнопку *Вставить*, номер рисунка записывается в позицию курсора. При добавлении или удалении рисунков номера меняются автоматически.

Удобнее всего нумеровать рисунки без обтекания текстом.

Пронумеруйте рисунки и вставьте на них ссылки в тексте.

Упражнения для самостоятельной работы

В свой файл вставьте таблицу с результатами сдачи экзаменов: дисциплина, дата сдачи, оценка. Проведите **сортировку** (упорядочение данных) таблицы по алфавиту дисциплин, по датам сдачи, по полученным оценкам (*Таблица / Сортировка...*). Добавьте в таблицу новую строку «Итого» и под графой с оценками подсчитайте свой суммарный балл, используя функцию суммы (панель инструментов *Таблица / Формула / Сумма*). К ячейке суммы вставьте соответствующее примечание (*Вставка / Примечание*). С подробностями выполнения этих операций разберитесь самостоятельно, используя справку.

ЛАБОРАТОРНАЯ РАБОТА 5. ПОДГОТОВКА ДОКУМЕНТА К ПЕЧАТИ

Проверка правописания

Для автоматической проверки правописания надо установить отключенные ранее флажки автоматической проверки орфографии и грамматики в меню *Сервис / Проверка орфографии*. Найденные слова с орфографическими ошибками подчеркиваются зеленой волнистой линией, с грамматическими – красной.

Программа проводит сравнение слов в набранном тексте со словами из встроенного словаря. В верхней части окна цветом выделено слово, отсутствующее в словаре, в нижней части – предлагаемый вариант для замены (исправления ошибки). При наличии ошибки нажимается кнопка *Заменить*, при отсутствии (слово, которого не оказалось в словаре) – кнопка *Пропустить*. Новое слово может быть добавлено в пользовательский словарь.

Для автоматической расстановки переносов в тексте необходимо в меню *Формат* выбрать *Стили*, в появившемся окне выбрать базовый стиль и вызвать для него контекстное меню *Изменить*, на вкладке *На странице* установить флажок *Автоматический перенос*. Или выполнить аналогичные действия только над отдельным абзацем в меню настройки параметров абзаца.

Создайте новый документ, скопируйте в него 6 абзацев любого текста. В первом и втором абзаце установите автоматические переносы.

Установка колонтитулов

Для установки колонтитулов (повторяющихся надписей в верхней или нижней части страницы) используйте меню *Вставка / Верхний или Нижний Колонтитулы*. С помощью меню *Вставка / Поля*, вставить дату, внести различные изменения. Для завершения создания колонтитулов щелкните по кнопке *Заккрыть*. Там же на вкладке *Формат / Страница* определяются параметры колонтитула.

В верхнем колонтитуле вашего текста напишите свою фамилию, в нижнем колонтитуле вставьте дату.

Создание сносок и концевых сносок

Сноски в текстовом документе могут быть внизу страницы или в конце всего текста (это концевые сноски). Для установки сноски поместите курсор в нужную позицию текста (в конце слова) и выберите *Вставка – Сноска*. Создайте три обычные сноски и три концевые сноски.

Нумерация страниц

Для простановки номеров страниц (если документ из нескольких страниц) воспользуйтесь меню *Вставка / Поля / Номера страниц*. При этом необходимо, чтобы указатель курсора находился в области колонтитула. Вставьте в нижний колонтитул номер страницы.

Использование колонок

Текст можно оформить в несколько колонок. Для этого выделите фрагмент текста, выберите меню *Формат / Колонки* и щелчком пометьте необходимое количество колонок текста, не забудьте установить расстояние между колонками – 1 см. Оформите четвертый абзац в 2 колонки.

Создание оглавления

Вставьте в текст заголовки и подзаголовки (т.е. озаглавьте каждый абзац текста).

Для подготовки оглавления к каждому заголовку раздела, подраздела, пункта и т.п. надо применить стиль заголовка соответствующего уровня (главные заголовки – Заголовок 1, подзаголовки – Заголовок 2 и т.д.). Выделяется заголовок первого раздела, в поле *Стиль* панели инструментов форматирования выбирается Заголовок 1 (при необходимости – переопределяются шрифт и выравнивание), таким же

стилем оформляются заголовки всех разделов. По аналогии для всех заголовков подразделов определяется стиль – Заголовок 2, и т.д.

Установите курсор в то место документа, где должно располагаться оглавление. Выберите *Вставка / Оглавление и указатели*. Выберите вид оглавления. При необходимости смените стиль оглавления, для этого внесите изменения на открытых вкладках. После нажатия ОК оглавление встроено.

Обычно оглавление делается на отдельной странице, поэтому после текста оглавления нужно вставить разрыв страницы с помощью меню *Вставка / Разрыв*.

Регистрация исправлений

Функция обзора исправлений позволяет фиксировать добавление, удаление, изменение текста и его оформления. Это необходимо, когда с документом работают несколько авторов, или когда Вы отслеживаете внесенные ранее исправления. При окончательной правке документа Вы можете взглянуть на каждое внесенное исправление и принять или отклонить его.

Для запуска процедуры фиксации исправлений выбираем *Правка / Изменения* и устанавливаем флажок перед надписью *Запись*.

Добавляемый в документ текст выделяется цветом и подчеркиванием, удаляемый текст остается видимым, но оформлен зачеркиванием.

Для прекращения записи исправлений убираем поставленный флажок.

При работе с документом, в котором фиксировались изменения, есть возможность принять их или отклонить. Для этого выбираем *Правка / Изменения / Принять или отклонить*. Далее просматриваем исправления в диалоговом режиме.

Печать документа

Для предварительного просмотра документа перед выводом на печать щелкните по кнопке *Предварительный просмотр*. Можно выбрать масштаб просмотра, количество одновременно просматриваемых страниц (кнопки *Одна страница* и *Несколько страниц*). Просмотрите созданный вами файл. Для выхода из режима просмотра щелкните по кнопке *Заккрыть*.

Для распечатки документа на принтере используется меню *Файл / Печать*, в диалоговом окне устанавливается тип принтера, выбираются страницы для печати (все, текущая, по номерам, выделенный фрагмент), число копий.

ЛАБОРАТОРНАЯ РАБОТА 6. ПОСТРОЕНИЕ ГРАФ-СХЕМ ЦИКЛИЧЕСКИХ ПРОЦЕССОВ

Построить граф-схему циклического процесса (структура цикла с предусловием) для следующего задания.

Вариант 1. В последовательности целых чисел определить количество положительных чисел и количество элементов, значения которых находятся в интервале от -10 до -20 .

Вариант 2. В последовательности целых чисел определить сумму положительных четных чисел.

Вариант 3. В последовательности чисел вывести на печать TRUE, если количество четных больше, чем нечетных и FALSE – иначе.

Вариант 4. В последовательности целых положительных чисел определить максимальное четное число и его порядковый номер.

Вариант 5. В последовательности вещественных чисел определить наименьшее отрицательное число и его порядковый номер.

Вариант 6. В последовательности целых чисел определить третье положительное число.

Вариант 7. В последовательности вещественных чисел определить наибольшее отрицательное число и его порядковый номер.

Вариант 8. В последовательности чисел выдать на печать TRUE, если значение максимального числа больше числа 10.

Вариант 9. В последовательности чисел подсчитать произведение чисел, кратных 3.

Вариант 10. В последовательности чисел сравнить, что больше, сумма положительных или произведение отрицательных.

Вариант 11. В последовательности чисел подсчитать количество четных и количество чисел кратных трем.

Вариант 12. В последовательности чисел определить предпоследнее отрицательное число.

Вариант 13. Вычислить сумму ряда, общий член которого задан формулой $A_n = (x^n)/n!$.

Вариант 14. При табулировании функции $y=\cos(x+a)$ на отрезке $[1,10]$ с шагом h определить сумму значений y .

Вариант 15. В последовательности целых чисел определить второе положительное число и его номер.

Вариант 16. Подсчитать количество цифр в целом числе X , которое будет введено.

Вариант 17. Вычислить факториал числа X .

Вариант 18. Найти минимальное значение функции $y=\sin(x)$ на отрезке $[1,10]$ с шагом 0.1 .

Вариант 19. Найти максимальное значение функции $y=\sin(x)$ на отрезке $[1,10]$ с шагом 0.1 .

Вариант 20. В последовательности чисел определить положение последнего отрицательного числа.

Построить граф-схему циклического процесса (структура цикла с постусловием) для следующего задания.

Вариант 1. В последовательности чисел сравнить, что больше, сумма положительных или произведение отрицательных.

Вариант 2. В последовательности чисел подсчитать количество четных и количество чисел кратных трем.

Вариант 3. В последовательности целых чисел определить второе положительное число.

Вариант 4. Вычислить факториал числа X .

Вариант 5. Найти минимальное значение функции $y=\sin(x)$ на отрезке $[1,10]$ с шагом 0.1 .

Вариант 6. Найти максимальное значение функции $y=\sin(x)$ на отрезке $[1,10]$ с шагом 0.1 .

Вариант 7. В последовательности чисел определить положение последнего отрицательного числа.

Вариант 8. При табулировании функции $y=\ln(x-a)$ на отрезке $[1,10]$ с шагом h определить сумму значений y .

Вариант 9. В последовательности целых чисел определить количество положительных чисел и количество элементов, значения которых находятся в интервале от -10 до -20 .

Вариант 10. В последовательности целых чисел определить сумму положительных четных чисел.

Вариант 11. В последовательности чисел вывести на печать TRUE, если количество четных больше, чем нечетных и FALSE – иначе.

Вариант 12. В последовательности целых положительных чисел определить максимальное четное число и его порядковый номер.

Вариант 13. Подсчитать количество цифр в целом числе X , которое будет введено.

Вариант 14. В последовательности целых положительных чисел определить максимальное четное число и его порядковый номер.

Вариант 15. В последовательности вещественных чисел определить наименьшее отрицательное число и его порядковый номер.

Вариант 16. В последовательности целых чисел определить третье положительное число.

Вариант 17. В последовательности вещественных чисел определить наибольшее отрицательное число и его порядковый номер.

Вариант 18. В последовательности чисел выдать на печать TRUE, если значение максимального числа больше числа 10.

Вариант 19. В последовательности чисел подсчитать произведение чисел, кратных 3.

Вариант 20. В последовательности целых чисел определить количество положительных чисел и количество элементов, значения которых находятся в интервале от 10 до 20.

ЛАБОРАТОРНАЯ РАБОТА 7. ОСНОВЫ РАБОТЫ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

Рабочая книга

Загрузите табличный процессор Calc. Рассмотрите элементы окна Calc. Большинство из них – это стандартные элементы Windows-приложений: заголовок, меню, панели инструментов – стандартная и форматирования, строка состояния, полосы прокрутки. Под панелями инструментов **строка формул** (ее назначение рассмотрим позже). Рабочая область представляет собой таблицу (**рабочий лист**), состоящую из **столбцов** с именами А, В, С ... и пронумерованных **строк**. На пересечении столбца и строки – **ячейка** с соответствующим **адресом**. Одна из ячеек выделена прямоугольной рамкой – это **активная ячейка** (сразу после загрузки это ячейка с адресом А1). Для активизации другой ячейки щелкните по ней мышью или переместитесь в нее с помощью клавиш со стрелками. Для перемещения по листу можно использовать полосы прокрутки и клавиатуру.

В нижней части окна – **ярлычки** листов (Лист1, Лист2 ...). Щелчок по ярлыку активизирует новый рабочий лист. Совокупность рабочих листов образует **рабочую книгу**, которая представляет собой один файл электронных таблиц. Количество листов может быть изменено пользователем (**Правка / Удалить лист** или **Вставка / Лист**). Вставьте Лист4 в рабочую книгу. Удалите листы Лист4 и Лист3 (при запросе подтвердите удаление). Лист может быть переименован: щелкните правой кнопкой мыши по ярлыку Лист1, в контекстном меню выберите **Переименовать** и введите с помощью клавиатуры новое имя: «Данные».

Для сохранения рабочей книги используйте меню **Файл / Сохранить как**. Для открытия новой рабочей книги используется кнопка **Создать**, для открытия ранее созданной книги – кнопка **Открыть**. Одновременно может быть открыто несколько рабочих книг. Переход между ними (активизация рабочей книги) производится с помощью меню **Окно**.

Ввод и редактирование данных

Проверьте *настройку параметров* процессора в меню *Сервис / Параметры*; на вкладке *Вычисления* – автоматически; на вкладке *Вид / отображение* строки формул и строки состояния, сетки, полосок прокрутки, ярлычков листов; просмотрите другие установленные настройки.

Предположим, что на рабочем листе надо отобразить данные по расчету стоимости компьютеров, продаваемых фирмой. В ячейке A1 введите текст: «Расчет стоимости компьютеров». По окончании ввода нажмите **Enter** или одну из клавиш со стрелкой. Снова активизируйте ячейку A1 (щелкните мышью) и посмотрите в строку формул: слева, в поле имен, стоит имя активной ячейки A1, справа – введенный текст. Обратите внимание на размещение текста в таблице: часть текста, не поместившаяся в ячейке A1, отображается в следующих ячейках слева (тем не менее вся информация считается введенной в ячейку A1, что видно из строки формул). Активизируйте ячейку B1, и введите в нее любые символы: отображение введенного ранее текста пропало; сам текст сохранился в ячейке A1, его можно увидеть в строке формул, вновь щелкнув мышью по A1. Теперь удалите символы, введенные в B1: активизируйте эту ячейку и нажмите клавишу **Delete**.

Во второй строке подготовьте заголовки столбцов таблицы: в ячейку A2 введите «Компьютер», в B2 – «Количество», C2 – «Цена», D2 – «Стоимость», E2 – «Рублей». При необходимости измените ширину столбца: подведите указатель мыши к правой границе наименования столбца A; когда он примет вид двунаправленной стрелки, протащите границу вправо до нужного положения. Пусть фирма имеет компьютеры шести типов, которые условно назовем Тип1, Тип2 и т.д. Введите в ячейке A3 текст «Тип1». Для ввода других типов можно использовать *автозаполнение*: установите указатель мыши на черный квадратик справа внизу активной ячейки; указатель при этом примет вид черного креста; протащите мышью вниз на шесть ячеек (до появления надписи «Тип6»). В ячейки B3.. B8 введите данные о количестве компьютеров соответствующего типа (от 2 до 20), а в ячейки C3.. C8 – цену одного

компьютера в условных единицах (от 300 до 1000). Обратите внимание на различия в выравнивании внутри ячеек текстовых и числовых данных: текст выравнивается по левой границе, числа – по правой (компьютер автоматически распознает, введен текст или число; при необходимости выравнивание можно изменить).

Для **редактирования данных** непосредственно в ячейке щелкните дважды по нужной ячейке, и перемещая вертикальную черту – курсор в нужное положение, внесите соответствующие изменения. Отредактируйте, например, ячейку B2: слово «Количество» измените на «Кол-во». После окончания редактирования нажмите **Enter**. Вместо двойного щелчка для редактирования можно использовать одинарный щелчок в строке формул или нажатие клавиши **F2**. При редактировании часто полезными оказываются кнопки *Отменить* и *Вернуть*.

Для **редактирования блока** ячеек блок выделяется протаскиванием мыши, например, от левой верхней до правой нижней ячейки, входящей в блок (указатель мыши при этом имеет вид белого крестика) или с помощью клавиатуры (**Shift** и клавиши со стрелками). Можно выделить блок с разрывами (то есть, по-существу несколько блоков как один): при протаскивании мышью должна быть одновременно нажата клавиша **Ctrl**: опробуйте такое выделение. В блоке может быть, например, изменен шрифт. Блок, как и ранее, можно удалить, скопировать, перенести. Возможен и перенос путем перетаскивания мышью: выделите любой блок, установите указатель мыши на его границу так, чтобы указатель принял вид стрелки и перетащите в любое другое место листа. При необходимости отмените это действие.

Вставьте под заголовком таблицы две новых строки для ввода даты и курса доллара. Активизируйте любую ячейку второй строки, дважды выберите меню *Вставка / Строки*. В ячейке D2 введите «Дата», а в ячейку E2 введите сегодняшнее число, например, в виде 12/01/20 (12 января 2020 года). После нажатия **Enter** скорее всего дата будет записана в другом формате – в том, который установлен в Calc. Для изменения **формата** (не только даты, но и любых данных) щелкните по ячейке правой кнопкой, в контекстном меню выберите *Формат ячеек*. На вкладке *Число* выберите

формат – *Дата*, справа выберите удобный тип представления даты. Просмотрите другие возможности форматирования ячеек на различных вкладках этого же диалогового окна. В ячейке D3 введите «Курс доллара», а в ячейке E3 – его значение. Для форматирования ячейки E3 щелкните по кнопке *Денежный формат* в панели инструментов форматирования. Отформатируйте теперь данные в ячейках C5 .. C10: выделите блок, вызовите контекстное меню *Формат ячеек*, установите формат – денежный, обозначение - \$ английский (США), щелкните ОК.

Использование формул

Активизируйте ячейку D5. Для расчета стоимости компьютеров Тип1 надо умножить цену одного компьютера на их количество: введите соответствующую формулу в эту ячейку. Ввод формулы начинается со знака = (с клавиатуры или щелчком по знаку = в строке формул), затем щелкните по ячейке C5 (где указана цена компьютера; ее адрес (*ссылка*) будет введен в формулу), нажмите клавишу со знаком умножения * (это удобнее сделать на малой цифровой клавиатуре), затем щелкните по ячейке B5 (количество компьютеров). По окончании ввода формулы щелкните по галочке в строке формул или нажмите клавишу **Enter**. В ячейке D5 получите результат расчета, в строке формул – зависимость, по которой этот результат получен.

Расчет в ячейках D6 .. D10 проводится по аналогичным формулам. Чтобы не вводить вручную это множество однотипных формул используется автозаполнение ячеек формулами. Активизируйте вновь ячейку D5 с введенной ранее формулой, установите указатель мыши на черный квадратик в правом нижнем углу (**маркер автозаполнения**) и протащите мышью до ячейки D10. После отпускания кнопки мыши в ячейках D5 .. D10 – результаты расчетов.

Щелкните по ячейке D6. Обратите внимание на строку формул: введенные в формулу в ячейке D5 адреса изменились; перемещайтесь последовательно до ячейки D10: все ссылки, входящие в формулу, изменяются в соответствии с адресом ячейки, в которой эта формула

находится – это *относительные ссылки* (адреса, изменяющиеся при заполнении ячеек формулами).

По аналогии проведите расчет в ячейках E5 .. E10: рассчитайте стоимость техники в рублях. Для этого надо стоимость в долларах умножить на курс доллара. Введите соответствующую формулу, используя ячейки D5 и E3 ($=D5*E3$). После нажатия **Enter** в ячейке E5 получится результат. Попробуйте заполнить формулами остальные ячейки: компьютер выдаст сообщение об ошибках (#ЗНАЧ – недопустимый тип аргумента). Действительно, активизировав ячейку E6, в строке формул можно видеть $=D6*E4$, но в E4 – текст! Кроме того, курс доллара не должен изменяться, соответственно адрес ячейки E3, где он введен, должен оставаться постоянным. *Абсолютная ссылка* – адрес ячейки, не изменяющийся при заполнении ячеек формулами. Выделите блок E6 .. E10 и очистите его. Отредактируйте формулу в ячейке E5: адрес ячейки должен принять вид $\$E\3 (эти значки доллара и указывают на абсолютность, то есть неизменяемость адреса). Нажмите **Enter**. Результат в ячейке E5 не изменился, однако теперь введенную в эту ячейку формулу можно распространить и на другие ячейки: проведите автозаполнение ячеек E6 .. E10. Посмотрите на соответствующие формулы в строке формул.

Внесите изменения в таблицу. Предположим, что количество компьютеров Тип2 увеличилось на 5 штук: введите соответствующее число в ячейку B6. Все результаты немедленно пересчитываются в соответствии с внесенным изменением. Внесите изменение в курс доллара. Убедитесь в автоматическом перерасчете всех данных.

Вставка функций

Для определения *суммы* чисел в строке или столбце таблицы используется кнопка *Автосумма*, находящаяся в стандартной панели инструментов. Для нахождения суммарной стоимости техники активизируйте ячейку E12 и щелкните по кнопке *Автосумма*. В ячейке появится функция суммирования СУММ(E5:E11): компьютер автоматически предлагает просуммировать все данные, находящиеся в

указанном диапазоне. После щелчка по галочке в строке формул или нажатия **Enter** в ячейке появится результат, при этом пустая ячейка E11 будет проигнорирована. При необходимости, используя редактирование формулы, аргумент функции суммирования может быть изменен, однако на практике чаще всего компьютер правильно «угадывает» нужный диапазон суммирования.

Для ввода любой из сотен встроенных в Calc функций используется кнопка *Вставка функций* (**f_x**), которая находится рядом с кнопкой *Автосумма*. Щелкните мышью по ячейке E2, куда была вставлена дата. Дата, вставленная вручную, будет оставаться в дальнейшем без изменений, иногда надо вставлять в документ текущую дату, тогда желательно, чтобы она изменялась автоматически. Вставьте текущую дату с помощью встроенной функции *NOW()*: щелкните по кнопке **f_x** , при этом загрузится окно *мастера функций*, в левой части которого выбирается категория функций – *дата и время*, в правой – функции. В нижней части окна приведено краткое описание выбранной функции. После щелчка ОК под строкой формул слева появляется панель функций, с помощью которой вводятся аргументы функции. Слева внизу – кнопка со знаком вопроса. Щелкните по ней для получения подробной справки по выбранной функции: с помощью *Помощника* выберите получение справки по этой теме, затем – справки по выделенной функции. Просмотрите справку и закройте справочное окно. Функция *NOW* не имеет аргументов, поэтому после щелчка ОК результат ее работы – текущая дата встроена в ячейку. При необходимости изменить вид даты отформатируйте ячейку.

Щелкните мышью по ячейке E13, вычислите в этой ячейке натуральный логарифм от найденной ранее в ячейке E12 суммарной стоимости (предположим, что он понадобился для выполнения некоторого расчета). Щелкните **f_x** , выберите в мастере слева категорию – *математические*, справа – *LN*, в панели функций активно поле ввода аргумента: щелкните по кнопке с красной стрелкой (кнопка *свертывания / развертывания* диалогового окна) для временного удаления панели, затем по ячейке E12 – соответствующая ссылка будет вставлена в окно. После

повторного щелчка по кнопке с красной стрелкой (обратите внимание на изменение ее вида – ввод закончен), затем по ОК, результат – в ячейке E13. Сохраните файл.

Оформление таблицы

Для *центрирования заголовка* таблицы в файле «Моя рабочая книга» выделите ячейки A1 .. E1, и щелкните по кнопке *Объединить и поместить в центре*. Увеличьте размер шрифта заголовка, установите полужирное начертание, измените цвет шрифта и цвет заливки заголовка, используя одноименные кнопки на панели инструментов форматирования. С помощью кнопки *Границы* выделите внешние и внутренние границы (рамки) таблицы линиями разного типа. Оформите таблицу в соответствии с Вашими вкусами. При необходимости используйте инструменты *Рисования*, вставку картинок, отключение изображения сетки и т.п.

Щелкните по ячейке A5 (Тип1) и в контекстном меню выберите *Примечание*. В появившемся окне введите характеристики компьютера Тип1: Intel, установите нужный размер окна. После щелчка вне окна примечание создано: теперь при активизации ячейки A5 это примечание будет появляться в виде всплывающей подсказки. Для компьютера Тип2 введите характеристики компьютера, с которым Вы работаете в компьютерном классе. По аналогии создайте остальные примечания.

Построение диаграммы

Для построения диаграмм используется *мастер диаграмм*, кнопка которого находится на стандартной панели инструментов. Постройте круговую диаграмму распределения количества компьютеров по типам. Выделите диапазон ячеек A4:B10, щелкните по кнопке мастера диаграмм, выберите тип диаграммы – *круговая*, на правой панели укажите один из видов круговой диаграммы. Щелкните по кнопке *Далее*; убедитесь, что данные – в столбцах, диапазон выделен правильно, на вкладке *Ряд* – имя ряда с данными – в ячейке B4, сами данные – в ячейках B5:B10, имена категорий – в первом выделенном столбце; вновь щелкните *Далее*. Введите

заголовок диаграммы: «Количество разных типов компьютеров», на вкладке *Подписи данных* выберите – доля (будут указаны доли разных типов компьютеров в процентах). На вкладке *Легенда* опробуйте удаление и вставку *легенды* – условных обозначений диаграммы, размещение ее в разных местах диаграммы: выберите наиболее удобное размещение легенды. На последнем шаге выбирается место размещения самой диаграммы – на этом же листе, или на отдельном. Разместите эту диаграмму на листе «Данные» и внимательно рассмотрите ее: рамку с квадратиками (маркерами), заголовок, легенду, рисунок, надписи. При необходимости любой из элементов диаграммы может быть изменен.

Теперь постройте гистограмму, отображающую цену и стоимость каждого типа компьютеров. Выделите блок ячеек A4:A10;C4:D10 (для одновременного выделения нескольких блоков (или блока с разрывами) при выделении держите нажатой клавишу **Ctrl**). Вызовите *мастер диаграмм*, выберите тип диаграммы – *гистограмма*, вид – трехмерная (внизу). Обратите внимание на запись выделенного диапазона на втором шаге мастера. На третьем шаге введите заголовок: «Цена и стоимость компьютерной техники», на вкладке *Оси* снимите флажок у оси «у», на вкладке *Линии сетки* – снимите все флажки (чтобы не затенять диаграмму линиями), результат разместите на отдельном листе «Диаграмма1». В рабочую книгу добавился рабочий лист с соответствующим названием, на котором и размещена гистограмма.

Редактирование диаграммы

Раскройте лист «Данные». Для изменения положения круговой диаграммы щелкните в области диаграммы мышью и перетащите ее в нужное положение. Для изменения размеров перетащите границу диаграммы за соответствующий маркер (один из восьми черных квадратиков на границах). Щелкните правой кнопкой по области диаграммы: с помощью контекстного меню можно изменить тип диаграммы, исходные данные, параметры диаграммы (заголовок, легенду, подписи), место ее размещения. Для форматирования области диаграммы

выберите соответствующую команду контекстного меню: поэкспериментируйте с *заливкой* области диаграммы (измените цвет, выберите способ – кнопка *Способы заливки*). Последовательно щелкая правой кнопкой мыши по различным частям диаграммы, можно вызывать соответствующие контекстные меню, позволяющие редактировать легенду, заголовки, ряды данных.

Еще больше возможностей редактирования имеет объемная гистограмма. Перейдите на лист «Диаграмма1», в контекстном меню области диаграммы выберите пункт *Объемный вид*: опробуйте изменение возвышения, поворота диаграммы, перспективы, щелкая по кнопкам увеличения или уменьшения соответствующей характеристики; выберите один из вариантов. Отформатируйте стенки диаграммы (рамка, заливка), основание, измените цену деления *оси* значений (контекстное меню *Оси значений / Формат оси / Шкала*). Сохраните измененный файл.

Упражнения для самостоятельной работы

Для различных наборов данных из таблицы «Данные» опробуйте другие типы диаграмм. Откройте новую рабочую книгу, в первом столбце первого листа введите значения аргумента от 0 до 4 с шагом 0.2 (воспользуйтесь меню *Правка / Заполнить / Прогрессия*), во втором столбце подсчитайте значения функции – синуса от введенных значений аргумента. На другом листе постройте соответствующую синусоиду, выделив оба столбца и выбрав тип диаграммы – точечная. Поэкспериментируйте с различными возможностями форматирования графика.

Откройте еще один новый лист, в первом столбце нового листа подсчитайте косинусы от значений аргумента, введенных в первый лист книги. Для этого щелкните по кнопке вставки функций, выберите косинус, а для вставки аргумента перейдите в нужный лист и щелкните в нем по ссылке на первое значение аргумента; проверьте, чтобы ссылка была относительной. Обратите внимание на запись ссылки в панели формул: указано имя книги, имя листа и адрес ячейки. Щелкните ОК и заполните формулами остальные ячейки.

Логические функции

Логические функции выполняют условное вычисление по формулам и позволяют осуществить реализацию простых алгоритмов принятия решений.

Функции **IF** возвращает одно значение, если заданное условие при вычислении дает значение TRUE (ИСТИНА), и другое значение, если FALSE (ЛОЖЬ). Функция **IF** используется при проверке условий для значений и формул.

Синтаксис функции:

IF (Лог_выражение; Значение_если_истина; Значение_если_ложь)

Предположим, что вам нужно подсчитать в рабочей таблице комиссионные по результатам продаж какого-либо товара. Если продавец продал продукции более, чем на \$100 000, то ставка его комиссионных составляет 7,5%, если меньше, то 5%. Без использования функции вам пришлось бы создавать две разные формулы и правильно употреблять их для каждого значения объема продаж. Ниже приведена функция, которая позволяет рассчитать комиссионные с учетом суммы продаж.

=IF (A1<100000; A1*0,05; A1*0,075), где

IF – функция, которая проверяет, выполняется ли условие, и возвращает одно значение, если оно выполняется, и другое значение, если нет;

A1<100000 – условие;

A1*0,05 – значение, которое возвращается, если условие выполняется;

A1*0,075 – значение, которое возвращается, если условие не выполняется.

Таким образом, если значение ячейки **A1** (сумма продаж) будет меньше 100000 (условие выполняется) программа умножит значение ячейки **A1** на 0,05 (ставка комиссионных), если же сумма продаж будет

больше 100000 (условие не выполняется), программа умножит значение ячейки **A1** на 0,075.

Выполните следующие действия:

1. Откройте окно **Мастер функций**, выберите категорию **Логические функции**. Просмотрите список функций, ознакомьтесь с их описанием.
2. Создайте таблицу 1.

Таблица 1

Норма продаж	150 000,00	
Ставка комиссионных	0,055	
Ставка премиальных	0,075	
Менеджеры	Продажи	Комиссионные
Орлов	149 823,00	
Воробьев	162 023,00	
Скворцов	209 123,00	
Синицын	122 354,00	
Грачев	83 351,00	
Петухов	204 861,00	
Ганиев	150 000,00	
Самойлов	110 500,00	
Дроздов	220 120,00	
Кротов	170 450,00	

3. Отформатируйте таблицу согласно рисунку 1

	А	В	С
1	Норма продаж	\$ 150 000,00	
2	Ставка комиссионных	5,5%	
3	Ставка премиальных	7,5%	
4			
5	Менеджеры	Продажи	Комиссионные
6	Орлов	\$ 155 000,00	\$ 11 625,00
7	Воробьев	\$ 162 023,00	\$ 12 151,73
8	Скворцов	\$ 209 123,00	\$ 15 684,23
9	Синицын	\$ 122 354,00	\$ 6 729,47
10	Грачев	\$ 83 351,00	\$ 4 584,31
11	Петухов	\$ 204 861,00	\$ 15 364,58
12	Ганиев	\$ 150 000,00	\$ 11 250,00
13	Самойлов	\$ 110 500,00	\$ 6 077,50
14	Дроздов	\$ 220 120,00	\$ 16 509,00
15	Кротов	\$ 170 450,00	\$ 12 783,75

Рис. 1

4. Вычислите комиссионные от продаж, если ставка комиссионных составляет 5,5%, ставка премиальных – 7,5%, норма продаж – 150 000 руб. Для этого выполните следующие действия.
5. Активизируйте ячейку **С6** (комиссионные Орлова), вызовите **Мастер функций**, найдите в нем функцию **IF**.
6. В поле **Логическое выражение** введите условие **В6<В1** (продажи Орлова меньше нормы продаж).
7. В поле **Значение если true (истина)** введите выражение **В6*В2** (если продажи Орлова меньше нормы продаж, то для расчета комиссионных необходимо величину продаж умножить на ставку комиссионных).
8. В поле **Значение если false (ложь)** введите выражение **В6*В3** (если продажи Орлова больше нормы продаж, то для расчета комиссионных необходимо величину продаж умножить на ставку премиальных).
9. Проверьте правильность ввода аргументов по рисунку 2.

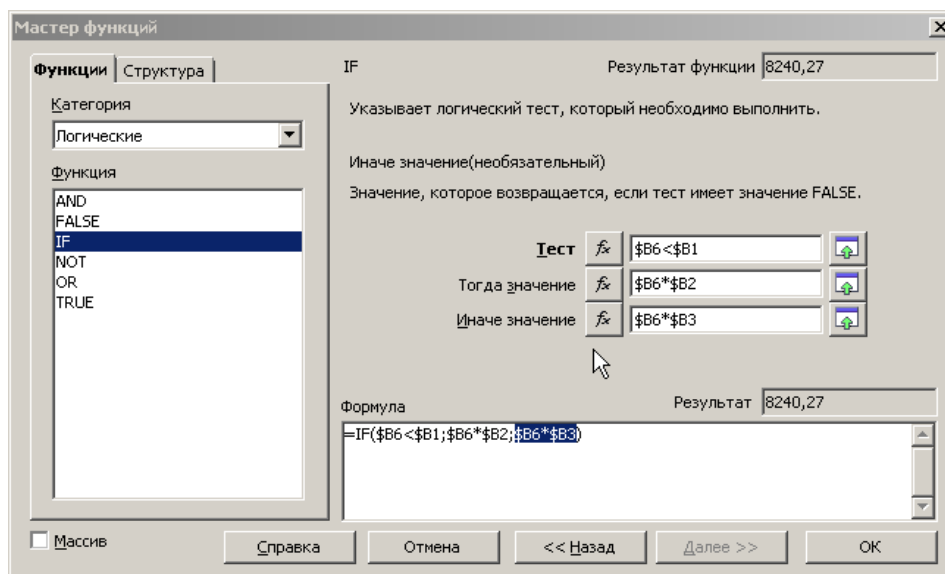


Рис. 2

10. Рассчитайте комиссионные по остальным продавцам. Для этого можно использовать функцию **Автозаполнение**, предварительно предусмотрев абсолютные ссылки.

11. Сравните полученные результаты с рисунком 3.

	А	В	С
1	Норма продаж	\$ 150 000,00	
2	Ставка комиссионных	5,5%	
3	Ставка премиальных	7,5%	
4			
5	Менеджеры	Продажи	Комиссионные
6	Орлов	\$ 149 823,00	\$ 8 240,27
7	Воробьев	\$ 162 023,00	\$ 12 151,73
8	Скворцов	\$ 209 123,00	\$ 15 684,23
9	Синицын	\$ 122 354,00	\$ 6 729,47
10	Грачев	\$ 83 351,00	\$ 4 584,31
11	Петухов	\$ 204 881,00	\$ 15 364,58
12	Ганиев	\$ 150 000,00	\$ 11 250,00
13	Самойлов	\$ 110 500,00	\$ 6 077,50
14	Дроздов	\$ 220 120,00	\$ 16 509,00
15	Кротов	\$ 170 450,00	\$ 12 783,75

Рис. 3

12. Сохраните рабочую книгу.

Математические функции. Функция SUMIF (СУММЕСЛИ)

В программу OpenOffice Calc включены 50 функций данной категории, что вполне достаточно для выполнения сложных математических расчетов.

Функция **SUMIF** суммирует ячейки, заданные критерием.

Синтаксис функции:

SUMIF(Диапазон;Критерий;Диапазон_суммирования).

Диапазон – диапазон вычисляемых ячеек.

Критерий – критерий в форме числа, выражения или текста, определяющего суммируемые ячейки. Например, критерий может быть выражен как 32, «32», «>32», «яблоки».

Диапазон_суммирования – фактические ячейки для суммирования.

Ячейки в **Диапазон_суммирования** суммируются, только если соответствующие им ячейки в аргументе **Диапазон** удовлетворяют критерию. Если **Диапазон_суммирования** опущен, то суммируются ячейки в аргументе **Диапазон**.

Функция **COUNTIF** подсчитывает число ячеек, соответствующих критерию.

Синтаксис функции:

COUNTIF(Диапазон;Критерий).

Диапазон – диапазон вычисляемых ячеек.

Критерий – критерий в форме числа, выражения или текста, определяющего суммируемые ячейки. Например, критерий может быть выражен как «>32», «Бухгалтерия».

Выполните следующие действия:

1. Откройте окно **Мастер функций**, выберите категорию **математических функций**.
2. Просмотрите список функций, ознакомьтесь с их описанием.
3. Создайте таблицу 2. Верхняя левая ячейка таблицы соответствует ячейке **A20**.

Таблица 2

Месяц	Регион	Продажи	Итоги по регионам	
Январь	Север	16350	Север	
Февраль	Юг	14586	Юг	
Январь	Запад	26874	Запад	
Февраль	Восток	8541	Восток	
Март	Север	96574	ВСЕГО	
Январь	Юг	6584		
Февраль	Запад	32584		
Март	Восток	7894	Итоги по месяцам	
Февраль	Север	14578	Январь	
Март	Юг	35941	Февраль	
Март	Запад	74581	Март	
Январь	Восток	5684	ВСЕГО	
	ИТОГО			

4. Отформатируйте таблицу согласно рисунку 4.

	A	B	C	D	E	F
20	Месяц	Регион	Продажи		Итоги по регионам	
21	Январь	Север	16 350,00р.		Север	127 502,00р.
22	Февраль	Юг	14 586,00р.		Юг	57 111,00р.
23	Январь	Запад	26 874,00р.		Запад	134 039,00р.
24	Февраль	Восток	8 541,00р.		Восток	22 119,00р.
25	Март	Север	96 574,00р.		ВСЕГО	340 771,00р.
26	Январь	Юг	6 584,00р.			
27	Февраль	Запад	32 584,00р.			
28	Март	Восток	7 894,00р.		Итоги по месяцам	
29	Февраль	Север	14 578,00р.		Январь	55 492,00р.
30	Март	Юг	35 941,00р.		Февраль	70 289,00р.
31	Март	Запад	74 581,00р.		Март	214 990,00р.
32	Январь	Восток	5 684,00р.		ВСЕГО	340 771,00р.
33		ИТОГО	340 771,00р.			

Рис. 4

5. Вычислите итоговые значения по регионам и по месяцам, а также суммарные значения продаж. Для этого выполните следующие действия.

6. Активизируйте ячейку **F21** (итоги по северу), вызовите **Мастер функций**, найдите в нем функцию **SUMIF**.

7. В поле **Диапазон** укажите диапазон ячеек **B21:B32** (в этом диапазоне ведется поиск критерия «Север»).

8. В поле **Критерий** введите ячейку **E21** (с этим значением происходит сравнение содержимого диапазона **B21:B32**).
9. В поле **Диапазон суммирования** укажите диапазон **C21:C32** (при нахождении в диапазоне **B21:B32** значения, отвечающего критерию, происходит суммирование соответствующих значений из диапазона **C21:C32**).
10. Проверьте правильность ввода аргументов по рисунку 5.

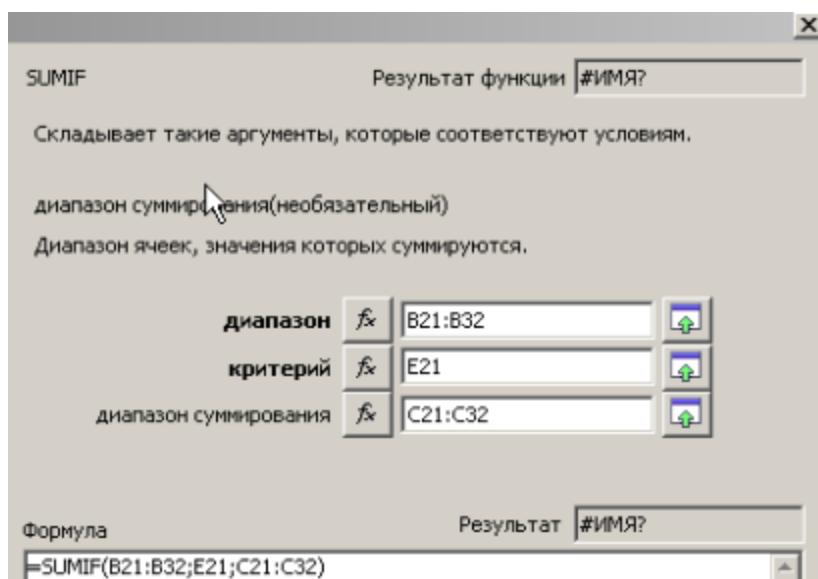


Рис. 5

11. Рассчитайте итоги по остальным регионам и месяцам (используйте Автозаполнение и абсолютные ссылки).
12. Вычислите итоговые значения. Сохраните полученные результаты.
13. Сравните полученные результаты с рисунком 6.

	A	B	C	D	E	F
20	Месяц	Регион	Продажи		Итоги по регионам	
21	Январь	Север	16 350,00р.		Север	127 502,00р.
22	Февраль	Юг	14 586,00р.		Юг	57 111,00р.
23	Январь	Запад	26 874,00р.		Запад	134 039,00р.
24	Февраль	Восток	8 541,00р.		Восток	22 119,00р.
25	Март	Север	96 574,00р.		ВСЕГО	340 771,00р.
26	Январь	Юг	6 584,00р.			
27	Февраль	Запад	32 584,00р.			
28	Март	Восток	7 894,00р.		Итоги по месяцам	
29	Февраль	Север	14 578,00р.		Январь	55 492,00р.
30	Март	Юг	35 941,00р.		Февраль	70 289,00р.
31	Март	Запад	74 581,00р.		Март	214 990,00р.
32	Январь	Восток	5 684,00р.		ВСЕГО	340 771,00р.
33		ИТОГО	340 771,00р.			

Рис. 6

Задание для самостоятельной работы:

1. Перейдите на новый лист. При отсутствии исходного файла вставьте новый лист, выполнив команду **Вставка**⇒**Лист**, создайте таблицу 3.
2. Отформатируйте таблицу согласно рисунку 7.

Таблица 3

Фамилия	Имя	Отчество	Отдел	Оклад	Премия	Ставки
Андреева	Анна	Семеновна	Бухгалтерия	5730		0,2
Бутаков	Андрей	Викторович	Сбыт	6250		0,15
Горбатов	Иван	Андреевич	Склад	4890		
Ерохин	Иван	Олегович	Склад	5800		
Иванов	Сергей	Александрович	Бухгалтерия	6520		
Крылова	Ольга	Сергеевна	Кадров	6250		
Маметов	Иван	Алексеевич	Сбыт	5730		
Петрова	Мария	Павловна	Кадров	4970		
Чарушин	Семен	Максимович	Склад	5940		
Яровцева	Елена	Викторовна	Бухгалтерия	5730		
			ИТОГО			
Кол-во сотрудников			Суммы окладов			
Бухгалтерия			Бухгалтерия			
Сбыт			Сбыт			
Склад			Склад			
Кадров			Кадров			
			ИТОГО			
>6000						
Иван						

	A	B	C	D	E	F	G
1	Фамилия	Имя	Отчество	Отдел	Оклад	Премия	Ставки
2	Андреева	Анна	Семеновна	Бухгалтерия	5 730,00р.		20%
3	Бутаков	Андрей	Викторович	Сбыт	6 250,00р.		15%
4	Горбатов	Иван	Андреевич	Склад	4 890,00р.		
5	Ерохин	Иван	Олегович	Склад	5 800,00р.		
6	Иванов	Сергей	Александрович	Бухгалтерия	6 520,00р.		
7	Крылова	Ольга	Сергеевна	Кадров	6 250,00р.		
8	Маметов	Иван	Алексеевич	Сбыт	5 730,00р.		
9	Петрова	Мария	Павловна	Кадров	4 970,00р.		
10	Чарушин	Семен	Максимович	Склад	5 940,00р.		
11	Яровцева	Елена	Викторовна	Бухгалтерия	5 730,00р.		
12				ИТОГО			
13							
14	Кол-во сотрудников			Суммы окладов			
15	Бухгалтерия			Бухгалтерия			
16	Сбыт			Сбыт			
17	Склад			Склад			
18	Кадров			Кадров			
19				ИТОГО			
20	>6000						
21	Иван						

Рис. 7

3. Определите общее количество сотрудников по каждому из отделов, используя функцию **COUNTIF**.
4. Подсчитайте количество сотрудников, имеющих оклад больше 6000 р.
5. Определите количество сотрудников с именем «**Иван**».
6. Подсчитайте общую сумму окладов сотрудников каждого отдела (ячейка **ИТОГО**).
7. Рассчитайте величину премии: для сотрудников, имеющих оклад меньше 5000 р., премия равна 15% от оклада, для остальных сотрудников – 20% от оклада, используя функцию **IF**.
8. Сохраните рабочую книгу.

Условное форматирование данных

Применение таких элементов формата ячейки, как свойства шрифта, наличие границы и цвет заливки ячеек, можно поставить в зависимость от ряда условий, например, пока значения в этих ячейках не достигнут некоторых контрольных значений. Для этого в OpenOffice.Calc предусмотрено средство **Условное форматирование данных**.

Например, на рисунке 9 представлена таблица, в которой с помощью условного форматирования заданы различные форматы для значений продаж выше нормы, ниже нормы и равных норме.

Задание:

1. Скопируйте созданную вами таблицу 1 на новый лист.
2. В новой таблице 1 выделите диапазон ячеек **B6:B15**.
3. Выполните команду *Формат / Условное форматирование*.
4. В открывшемся окне задайте условие (значение больше 150 000).
5. Нажмите на кнопку *Формат*.
6. В окне *Формат ячейки* задайте начертание и цвет шрифта (вкладка *Шрифт*) и заливку ячеек (вкладка *Вид*).
7. Нажмите кнопку *А также >>*. Задайте форматы, как показано на рисунке 8. Нажмите кнопку **ОК**.

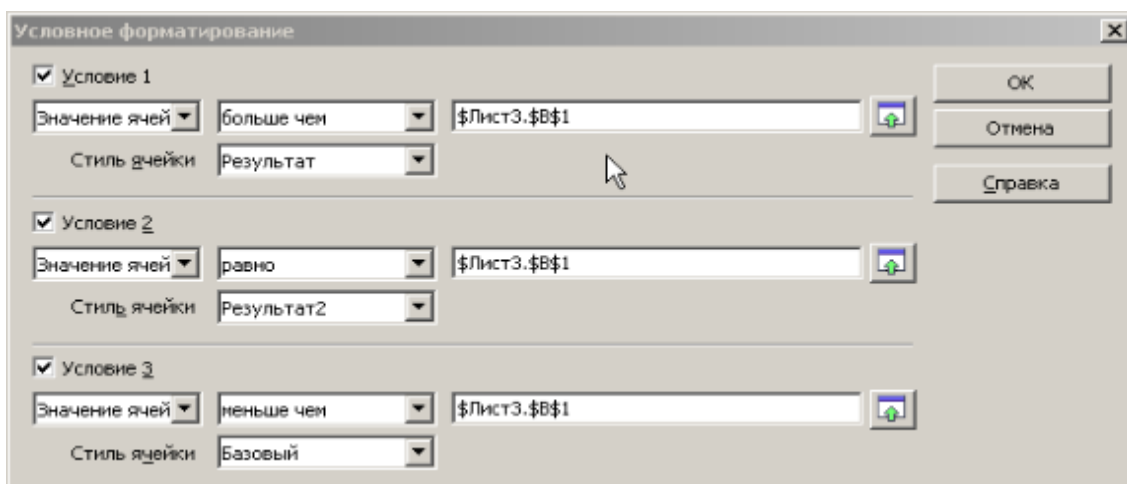


Рис. 8

8. Результат работы представлен на рисунке 9.

	А	В	С
1	Норма продаж	\$ 150 000,00	
2	Ставка комиссионных	5,5%	
3	Ставка премиальных	7,5%	
4			
5	Менеджеры	Продажи	Комиссионные
6	Орлов	\$ 149 823,00	\$ 8 240,27
7	Воробьев	\$ 162 023,00	\$ 12 151,73
8	Скворцов	\$ 209 123,00	\$ 15 684,23
9	Синицын	\$ 122 354,00	\$ 6 729,47
10	Грачев	\$ 83 351,00	\$ 4 584,31
11	Петухов	\$ 204 861,00	\$ 15 364,58
12	Ганиев	\$ 150 000,00	\$ 11 250,00
13	Самойлов	\$ 110 500,00	\$ 6 077,50
14	Дроздов	\$ 220 120,00	\$ 16 509,00
15	Кротов	\$ 170 450,00	\$ 12 783,75

Рис. 9

9. Измените значение продажи для первого менеджера. Например, введите значение выше нормы. Убедитесь, что формат ячейки автоматически изменился.

	А	В	С
1	Норма продаж	\$ 150 000,00	
2	Ставка комиссионных	5,5%	
3	Ставка премиальных	7,5%	
4			
5	Менеджеры	Продажи	Комиссионные
6	Орлов	\$ 155 000,00	\$ 11 625,00
7	Воробьев	\$ 162 023,00	\$ 12 151,73
8	Скворцов	\$ 209 123,00	\$ 15 684,23
9	Синицын	\$ 122 354,00	\$ 6 729,47
10	Грачев	\$ 83 351,00	\$ 4 584,31
11	Петухов	\$ 204 861,00	\$ 15 364,58
12	Ганиев	\$ 150 000,00	\$ 11 250,00
13	Самойлов	\$ 110 500,00	\$ 6 077,50
14	Дроздов	\$ 220 120,00	\$ 16 509,00
15	Кротов	\$ 170 450,00	\$ 12 783,75

Рис. 10

10. Сохраните файл.

Задание для самостоятельной работы:

1. Перейдите на новый лист и создайте таблицу 4.

Таблица 4

Прибыль/убытки				
	Квартал 1	Квартал 2	Квартал 3	Квартал 4
Магазин 1	45000	35000	-1000	-7000
Магазин 2	-50000	0	-35000	4000
Магазин 3	20000	15000	10000	0
Магазин 4	-10000	-24000	2000	8000
Магазин 5	0	-3000	6000	2000

2. С помощью условного форматирования выделите прибыль, меньше 10 000 р., прибыль больше 30 000 р. и убытки больше – 2000 р. Имейте

ввиду, что прибыль – это значения больше нуля, убытки – значения меньше нуля.

3. Отформатируйте таблицу.

4. Примерный результат показан на рисунке 11.

5. Сохраните файл.

	А	В	С	Д	Е
1	Прибыль/убытки				
2		Квартал 1	Квартал 2	Квартал 3	Квартал 4
3	<i>Магазин 1</i>	\$45 000,00	\$35 000,00	-\$1 000,00	-\$7 000,00
4	<i>Магазин 2</i>	-\$50 000,00	\$0,00	-\$35 000,00	\$4 000,00
5	<i>Магазин 3</i>	\$20 000,00	\$15 000,00	\$10 000,00	\$0,00
6	<i>Магазин 4</i>	-\$10 000,00	-\$24 000,00	\$2 000,00	\$8 000,00
7	<i>Магазин 5</i>	\$0,00	-\$3 000,00	\$6 000,00	\$2 000,00

Рис. 11

ЛАБОРАТОРНАЯ РАБОТА 9. РАБОТА СО СПИСКАМИ

Создание списка

Таблица *базы данных* в Calc называется *списком*. Столбцы в списке – это *поля* базы данных, а их заголовки – *имена* соответствующих полей. Каждая строка списка – это *запись* в базе данных. Список создается на отдельном листе. Для автоматической идентификации при последующей обработке входящих в него данных слева оставляется пустой столбец, сверху – пустая строка. Первая строка списка – имена полей; для удобства работы целесообразно выделить эту строку шрифтом, отличным от шрифта записей. Все данные одного столбца должны быть однотипны – иметь один формат и одинаковый шрифт. Некоторые поля могут заполняться по результатам вычислений (*вычисляемые поля*).

Для создания списка с информацией о Ваших товарищах по факультету введите имена полей: в ячейке B2 – Группа, C2 – Фамилия, D2 – Имя, далее – Дата (рождения): в формате даты, Экзамен1, Экзамен2, Экзамен3 – три поля с оценками по результатам последней сессии, средний Балл (вычисляемое поле; установить числовой формат с выводом результата с двумя знаками после запятой). Выделите имена полей жирным шрифтом.

В следующей строке (без пропуска строки) введите данные о себе, а в поле Балл введите соответствующую формулу $=(F3+G3+H3)/3$, или функцию для расчета среднего значения: вставка функций, категория – статистические, аргумент – протащите мышью по ячейкам F3:H3.

Введите данные для 15 студентов.

Переименуйте рабочий лист: Исходный список. Сохраните файл под именем Список.

Сортировка списка

Скопируйте список на два других рабочих листа этой же книги. Переименуйте лист 2: Сортировка по фамилиям. Для сортировки –

упорядочения данных по алфавиту или в порядке возрастания (убывания) числовых значений – щелкните мышью по любой ячейке списка, выберите меню *Данные / Сортировка*, проверьте установку переключателя в группе Идентифицировать столбцы по (подписям). В поле *Сортировать по* выберите в раскрывающемся списке **Группа**, в поле *Затем по* – **Фамилия**, в *последнюю очередь по* – **Имя**. Все три соответствующие переключателя установите в положение по возрастанию. После щелчка ОК будет проведена сортировка списка по возрастанию номеров групп, при совпадении номеров групп – по алфавиту фамилий, при совпадении фамилий – по именам.

Переименуйте лист 3 с другой копией списка: Сортировка по баллу. Опробуйте здесь различные варианты сортировки: по датам, по одному и нескольким экзаменам. Последнюю сортировку выполните по группам и баллам (то есть сортировка по баллам внутри каждой группы).

Поиск с помощью стандартного фильтра

Поиск нужных данных из списка может быть произведен с помощью формы (*Данные / Стандартный фильтр*). В соответствующие поля вводятся условия поиска (**критерии**): для поиска по точному соответствию вводится нужная запись, при поиске по близкому соответствию используются подстановочные символы * (любая последовательность символов) или ? (один неизвестный символ), при поиске по числовому сравнению используются операторы отношения <, <=, > и т.п. Если указаны критерии поиска для нескольких полей, то будут искажаться записи, удовлетворяющие всем условиям одновременно.

Найдите отдельно по каждому критерию: студентов с именами, например, Сергей, Ольга, с фамилиями, начинающимися на А, К, родившихся до 2000 года (введите в поле Дата <1/1/2000 – без пробелов!), со средним баллом, большим 4. Для перехода от одной записи к другой щелкайте по кнопкам *Назад* или *Далее*. При вводе нового критерия не забывайте удалять предыдущий (кнопка *Очистить*).

Опробуйте поиск по нескольким критериям одновременно, например, родились до 2000 года, при этом средний балл больше 3,5.

Автофильтрация данных

Для вывода записей, удовлетворяющих заданным условиям, на рабочий лист используется **фильтрация** данных. Для автофильтрации щелкните мышью по одной из ячеек списка и воспользуйтесь меню *Данные / Фильтр / Автофильтр*. У каждого имени поля появляется кнопка со стрелкой, используемая для отбора данных: после щелчка по стрелке у любого из полей можно выбрать из раскрывающегося списка – *Все записи, Первые 10 записей*, конкретное значение или задать условие фильтрации.

Выберите в раскрывающемся списке поля Группа значение, соответствующее Вашей группе: все остальные данные будут удалены. Для восстановления всех данных в списке поля Группа выберите *Все* или меню *Данные / Фильтр / Отобразить все*. Можно фильтровать список несколько раз. Снова выберите значение, соответствующее Вашей группе, в списке поля Экзамен1 выберите 5: будут выведены данные о студентах Вашей группы, получившие отличные оценки за первый экзамен.

Восстановите список. Для задания более сложных условий отбора используется пункт *Условие*. В раскрывающемся списке поля Балл выберите *Условие*; для вывода данных о студентах со средним баллом от 3.5 до 5 в диалоговом окне введите в верхней строке слева – больше (выбирается из раскрывающегося списка), в правой – 3.5, проверьте установку переключателя **И / ИЛИ** на **И**, в нижней строке слева выберите из списка меньше или равно, справа – 5, щелкните ОК. Восстановите список.

Выберите в Вашей группе фамилии, начинающиеся, например, на К и родившихся до 1/07/2000 (вначале отбор по группе, затем – по фамилиям: больше К, в последнюю очередь – по датам). Восстановите список.

Для исходного списка выберите фамилии студентов со средним баллом от 4 до 5.

В восстановленном списке найдите данные о студентах, родившихся, например, до 2000 года или позже 2002 года. Вновь восстановите исходный список. Для отмены режима автофильтрации вновь выберите *Данные / Фильтр / Автофильтр*.

Расширенная фильтрация

Для работы с расширенным фильтром вначале просмотрите справку по этому вопросу, примеры условий отбора расширенного фильтра.

Алгоритм использования расширенного фильтра

- вначале скопируйте строку имен полей в область рабочего листа ниже списка;

- под скопированной строкой введите условия поиска (эта область листа называется диапазон критериев); условия, введенные в одной строке, выполняются с логическим **И**, в разных строках – с логическим **ИЛИ**. Пример задания критерия поиска по результатам экзамена1 приведен ниже;

	A	B	C	D	E	F	G	H	I
1									
2		группа	Фамилия	Имя	Дата рождения	Экзамен1	Экзамен2	Экзамен3	Средний балл
3		ПМбд-11	Иванов	Илья	03.04.00	5	3	5	4,33
4		ПМбд-11	Серов	Александр	25.09.00	4	4	4	4,00
5		ПМбд-11	Якушев	Виталий	05.08.99	5	5	5	5,00
6		ПМбд-11	Морозова	Виктория	22.05.98	3	4	4	3,67
7		ПМбд-12	Крючков	Илья	02.02.04	5	5	5	5,00
8		ПМбд-12	Аверин	Сергей	12.11.03	4	3	4	3,67
9		ПМбд-12	Юдин	Сергей	03.12.00	3	4	5	4,00
10		ПМбд-12	Петров	Илья	25.07.00	3	5	5	4,33
11									
12		группа	Фамилия	Имя	Дата рождения	Экзамен1	Экзамен2	Экзамен3	Средний балл
13						>=4			
14									

- щелкните по любой ячейке основного списка;

- войдите в меню *Данные / Фильтр / Расширенный фильтр*;

- в диалоговом окне выберите место размещения критерия и отфильтрованного списка. Например,

	A	B	C	D	E	F	G	H	I	J
1										
2		группа	Фамилия	Имя	Дата рождения	Экзамен1	Экзамен2	Экзамен3	Средний балл	
3		ПМбд-11	Иванов	Илья	03.04.00	5	3	5	4,33	
4		ПМбд-11	Серов	Александр	25.09.00	4	4	4	4,00	
5		ПМбд-11	Якушев	Виталий	05.08.99	5	5	5	5,00	
6		ПМбд-11	Морозова	Виктория	22.05.98	3	4	4	3,67	
7		ПМбд-12	Крючков	Илья	02.02.04	5	5	5	5,00	
8		ПМбд-12	Аверин	Сергей	12.11.03	4	3	4	3,67	
9		ПМбд-12	Юдин	Сергей	03.12.00	3	4	5	4,00	
10		ПМбд-12	Петров	Илья	25.07.00	3	5	5	4,33	
11										
12		группа	Фамилия	Имя	Дата рождения	Экзамен1	Экзамен2	Экзамен3	Средний балл	
13						>=4				
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										

Расширенный фильтр

Взять критерии фильтра из
 \$Лист1.\$B\$12:\$I\$13

Параметры

Учитывать регистр Регулярное выражение

Заголовки в первой строке Без повторений

Поместить результат в: Сохранять критерии

\$Лист1.\$B\$15

Диапазон данных: \$Лист1.\$B\$2:\$I\$10 (безымянный)

OK Отмена Справка Ещё ▲

- проверьте правильность ссылок на исходный список;
- укажите ссылку на диапазон критериев, предварительно активизировав щелчком мыши соответствующее поле;
- щелкните ОК.

Последние два задания, выполненные с помощью автофильтра, выполните с использованием расширенного фильтра. Сравните результат.

СПИСОК ЛИТЕРАТУРЫ

1. Афанасьева, Т. В. Алгоритмы и программы : учебное пособие / Т. В. Афанасьева, Ю. Е. Кувайскова, В. А. Фасхутдинова. – Ульяновск : УлГТУ, 2011. – 227 с.
2. Белоцерковская, И. Е. Алгоритмизация. Введение в язык программирования С++ [Электронный ресурс] : учебное пособие / И. Е. Белоцерковская, Н. В. Галина, Л. Ю. Катаева. – Электрон. дан. – Москва, 2016. – 196 с. – Режим доступа: <https://e.lanbook.com/book/100564>
3. Клячкин, В. Н. Компьютерный практикум: офисные технологии / В. Н. Клячкин, Т. Е. Родионова : сборник лабораторных работ по курсу «Информатика» для студентов экономико-математического факультета (1 семестр). – Ульяновск : УлГТУ, 2000. – 27 с.
4. Граничин, О. Н. Информационные технологии в управлении [Электронный ресурс] : учебное пособие / О. Н. Граничин, В. И. Кияев. – Электрон. дан. – Москва, 2016. – 400 с. – Режим доступа: <https://e.lanbook.com/book/100598>.
5. Исакова, А. И. Основы информационных технологий [Электронный ресурс] : учебное пособие / А. И. Исакова. – Электрон. дан. – Москва : ТУСУР, 2016. – 206 с. – Режим доступа: <https://e.lanbook.com/book/110256>.
6. Кадырова, Г. Р. Курс лекций по информатике (В двух частях) Часть 2 : учебное пособие / Г. Р. Кадырова. – Ульяновск : УлГТУ, 2008. – 133 с.
7. Кадырова, Г. Р. Курс лекций по информатике : учебное пособие. В 2 частях. Часть. 1 / Г. Р. Кадырова. – Ульяновск : УлГТУ, 2008. – 100 с.
8. Кадырова, Г. Р. Основы алгоритмизации и программирования : учебное пособие / Г. Р. Кадырова. – Ульяновск : УлГТУ, 2014. – 95 с.
9. Компьютерный практикум. Работа в OpenOffice.org : методические указания к лабораторным работам / сост. Т. Е. Родионова. – Ульяновск : УлГТУ, 2010. – 50 с.
10. Левин, В. И. История информационных технологий [Электронный ресурс] : учебное пособие / В. И. Левин. – Электрон. дан. – Москва, 2016. – 751 с. – Режим доступа: <https://e.lanbook.com/book/100614>.
11. Роганов, Е. А. Практическая информатика [Электронный ресурс] : учебное пособие / Е. А. Роганов. – Электрон. дан. – Москва, 2016. – 640 с. – Режим доступа: <https://e.lanbook.com/book/100373>.

12. Родионова, Т. Е. Технологии программирования : учебное пособие для студентов направления 01.03.04 / Т. Е. Родионова. – Ульяновск : УлГТУ, 2018. – 115 с.

13. Советов, Б. Я. Информационные технологии : учебник для студ. вузов, обучающихся по направлению подготовки дипл. спец. «Информатика и вычислительная техника» и «Информационные системы2 / Б. Я. Советов, В. В. Цехановский. – 4-е изд., стер. – Москва : Высшая школа, 2008. – 263 с.

14. Сырецкий, Г. А. Информатика. Фундаментальный курс : учебник для вузов: в 2 т. / Г. А. Сырецкий. – Санкт-Петербург : БХВ-Петербург, 2007. – Т. 2 : Информационные технологии и системы. – 846 с.

Интернет-источники

1. Федеральный портал Единое окно доступа к образовательным ресурсам <http://window.edu.ru/library>

2. Научная электронная библиотека <http://elibrary.ru/defaultx.asp>

3. Электронно-библиотечная система издательства «Лань» <http://e.lanbook.com/>

4. Электронная библиотека полнотекстовых учебных и научных изданий УлГТУ <http://venec.ulstu.ru/lib/faculty.php>

5. Основы информационных технологий URL: <https://www.intuit.ru/studies/courses/3481/723/info>

ПРИЛОЖЕНИЕ 1

Значение чисел в различных системах счисления

10	2	8	16
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

ПРИЛОЖЕНИЕ 2

Значение степеней числа 2

Степень n	Значение 2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

Учебное издание

Родионова Т. Е.

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
ОБРАБОТКИ ДАННЫХ**

Учебное пособие

Редактор М. В. Теленкова
Подписано в печать 01.06.2020.
Формат 60×84/16. Усл. печ. л. 6,74.
Тираж 50 экз. Заказ 382. ЭИ № 1457.

Ульяновский государственный технический университет
432027, г. Ульяновск, ул. Сев. Венец, д. 32.
ИПК «Венец» УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, д. 32.