Тема 2. **Технические и программные средства реализации информационных процессов**

Вопросы темы 2

1. Назначение и области применения ЭВМ. Классификация ЭВМ.
2. История развития ЭВМ.
3. Машина фон Неймана. Принцип действия ЭВМ.

Вопрос 1. Назначение и области применения ЭВМ

- Под ЭВМ понимают электронные вычислительные машины, в которых информация кодируется двоичными кодами чисел. Именно эти машины благодаря универсальным возможностям и являются самой массовой вычислительной техникой.
- ЭВМ — основное техническое средство обработки информации.
- По назначению ЭВМ можно разделить на три группы:

1. **Универсальные** (общего назначения) — предназначены для решения различных инженерно-технических задач: экономических, математических, информационных и других, отличающихся сложностью алгоритмов и большим объемом обрабатываемых данных. Характерными чертами этих ЭВМ являются высокая производительность, разнообразие форм обрабатываемых данных (двоичных, десятичных, символьных), разнообразие выполняемых операций (арифметических, логических, специальных), большая емкость оперативной памяти, развитая организация ввода-вывода.

2. **Проблемно-ориентированные** — предназначены для решение более узкого круга задач, связанных обычно с технологическими объектами, регистрацией, накоплением и обработкой небольших объемов данных (управляющие вычислительные комплексы).

3. **Специализированные** — для решения узкого круга задач, чтобы снизить сложность и стоимость этих ЭВМ, сохраняя высокую производительность и надежность работы (программируемые микропроцессоры специального назначения, контроллеры, выполняющие функции управления

Области применения ЭВМ

Академик В.М. Глушков указывал, что существуют три глобальные сферы деятельности человека, которые требуют использования качественно различных типов ЭВМ.

- Первая область применения ЭВМ является традиционной - **использование ЭВМ для автоматизации вычислений**.
- Научно-техническая революция во всех областях науки и техники постоянно выдвигает новые научные, инженерные, экономические задачи, которые требуют проведения крупномасштабных вычислений (задачи проектирования новых образцов техники, моделирования сложных процессов, атомная и космическая техника и др.).

- Вторая сфера применения ЭВМ связана с **использованием их в системах управления**.
- Она родилась примерно в 60-е годы, когда ЭВМ стали интенсивно внедряться в контуры управления автоматических и автоматизированных систем.
- Математическая база этой новой сферы практически отсутствовала, в течение последующих 15-20 лет она была создана.

- Третья область связана с применением ЭВМ **для решения задач искусственного интеллекта**.
- Задачи искусственного интеллекта предполагают получение не точного, а нечеткого (fuzzy) результата на основе использования нечеткой логики, теории вероятностей, теории возможностей и других разделов современной математики.

Классификация ЭВМ

Для технического обеспечения третьего направления нужны качественно новые структуры ЭВМ. Перечисление областей применения ЭВМ показывает, что для решения различных задач нужна соответственно и различная вычислительная техника. Поэтому рынок компьютеров постоянно имеет широкую градацию классов и моделей ЭВМ.

По быстродейтсвию ЭВМ можно разделить на следующие классы:

- **СуперЭВМ** для решения крупномасштабных вычислительных задач, для обслуживания крупнейших информационных банков данных.
- **Большие ЭВМ** для комплектования ведомственных, территориальных и региональных вычислительных центров.
- **Средние ЭВМ** широкого назначения для управления сложными технологическими производственными процессами.
- **Персональные и профессиональные ЭВМ**, позволяющие удовлетворять индивидуальные потребности пользователей. На базе этого класса ЭВМ строятся автоматизированные рабочие места (АРМ) для специалистов различного уровня.
- **Встраиваемые микропроцессоры**, осуществляющие автоматизацию управления отдельными устройствами и механизмами.

Вопрос 2. История развития ЭВМ

- Первое **автоматическое** устройство для вычислений – разработал немецкий профессор Вильгельм Шикард. Он назвал его **«суммирующие часы».**
- В 1642 французский механик Блез Паскаль разработал более компактное суммирующее устройство (**арифмометр**), которое стало первым в мире механическим калькулятором, выпускающимся серийно.
- В 1673 г. немецкий математик и философ Лейбниц создал **механический калькулятор**, который мог выполнить операции умножения и деления путем многократного повторения операций сложения и вычитания.

- Идея программирования вычислительных операций пришла из часовой промышленности. Старинные монастырские башенные часы были настроены так, чтобы в заданное время включать механизм, связанный с системой колоколов – жесткое программирование.
- Идея гибкого программирования механических устройств с помощью перфорированной бумажной ленты впервые была реализована в 1804 г. в ткацком станке Жаккарда.
- Идея **программного управления вычислительными операциями** была реализована выдающимся английским математиком и изобретателем Чарльзом Бэббиджем в его **Аналитической машине**. Особенностью Аналитической машины стало то, что здесь впервые был реализован принцип разделения информации на команды и данные.

- Будущему развитию вычислительной техники предшествовали **два математических первоисточника**:
- - **1666** г. – Готфрид Лейбниц предложил **двоичную систему счисления**, занимаясь исследованиями философской концепции единства и борьбы противоположностей;
- - создание в **первой половине 19 – го** века выдающимся английским ученым Джорджем Булем **логической алгебры**. Впоследствии она стала называться булевой алгеброй. Операции Буля ориентировались на работу только с двумя сущностями: **истина и ложь**. При создании ЭВМ из системы Буля были использованы четыре основные операции: **И**(пересечение), **ИЛИ** (объединение), **НЕ**(обращение) и **ИСКЛЮЧАЮЩЕЕ ИЛИ**. Эти операции лежат в **основе работы** всех видов процессоров современных компьютеров.

- В **сороковых годах XX века** появились вычислительные машины, в которых использовались электронные элементы- радиолампы. Первой действующей ЭВМ была американская машина (1945г.) **ENIAC** – **E**lectronic **N**umerical **I**ntegrator **A**nd **C**alculator. Авторы – американские ученые **Эккерт и Моучли**., весила 30 т., содержала 18000 электронных ламп, выполняла 5000 опер. в секунду.
- В 1945г. к работе был привлечен американский ученый дж. **Фон Нейман**. Им были сформулированы **принципы построения ЭВМ**, которые лежат в основе работы и современных ЭВМ.

1945-1960 гг – это ЭВМ ***первого поколения*** (на лампах), они использовались в основном для научных расчетов. Программы набирались с помощью проводов. *Первой отечественной* ЭВМ была **МЭСМ** (малая электронная счетная машина), разработанная в 1947-1951 гг под руководством С.А. Лебедева. В 1952г. была введена в эксплуатацию **БЭСМ** (большая электронная счетная машина). В 1955г. начался выпуск малой ЭВМ «**Урал 1**».

В **1949**г. был создан полупроводниковый прибор – **транзисторы**. С освоением их промышленного производства они полностью вытеснили из ЭВМ электронные лампы. ЭВМ на их базе стали в **сотни раз меньше** ламповых ЭВМ такой же производительности. Скорость ЭВМ возросла до **сотен тысяч операций в сек**. Это стали ЭВМ **второго поколения** (конец 50-х – середина 60-х). Отечественные ЭВМ этого поколения: **БЭСМ-4**, **М-220**, **Минск-32**.

- **Третье поколение** (конец 60-х – начало 70-х) характеризуется появлением в качестве элементной базы процессора **интегральных полупроводниковых схем** (вместо отдельных транзисторов), которые называются **чипами**.
- Переход к интегральным схемам позволил довести скорость ЭВМ до миллиона операций в секунду.
- Переход к **машинам 4-го поколения** – ЭВМ на **больших интегральных схемах (БИС)** – происходил во второй половине **70-х** годов и завершился к **80-м** годам. Теперь на одном кристалле размером 1 $см^3$ стали размещаться сотни тысяч электронных элементов. Скорость ЭВМ составляет до 10^9 опер./сек.
- Наиболее крупным достижением, связанным с применением БИС, стало **создание микропроцессоров** (разработчик-работник фирмы Intel Маршиан Эдвард Хофф создал интегральную схему, аналогичную по своим функциям центральному процессору большой ЭВМ). На

- С **конца 80-х** годов в истории развития ВТ наступила пора **5-го поколения** ЭВМ. Их структура и архитектура отличаются от классической фон-неймановской.
- Машина состоит из **нескольких блоков**.

1. **Блок общения** обеспечивает интерфейс между пользователем и ЭВМ на естественном языке.
2. **Блок знаний** хранит знания, накопленные человечеством в различных предметных областях, которые постоянно расширяются и пополняются.
3. **Блок-решатель** организует подготовку программы решения задачи на основании знаний, получаемых из базы знаний и исходных данных, полученных из блока общения.
4. **Ядро** вычислительной системы составляет **ЭВМ высокой производительности (супер-ЭВМ)**.

Вопрос 3. Машина фон Неймана. Принцип действия ЭВМ

- Большинство современных ЭВМ строится на базе **принципов**, сформулированных американским ученым, одним из отцов кибернетики **Дж. Фон Нейманом:**
- **1.** Основными блоками фон-неймановской машины являются: **Устройство Ууправления(УУ), Арифметико-логическое устрйоство (АЛУ), память, устройства ввода-вывода**;
- **2.** Информация кодируется в двоичной форме и разделяется на **единицы**, называемые **словами**;

3. Алгоритм представляется в форме **последовательности управляющих слов**, называемых командами. Совокупность команд, представляющая алгоритм, называется программой;

4. Программы и данные хранятся в **одной и той же памяти**.

5. УУ и АЛУ обычно объединяются в одно, называемое **центральным процессором**. Они определяют действия путем считывания команд из оперативной памяти. Обработка информации, предписанная алгоритмом, сводится к последовательному выполнению команд.

- Команда состоит из **кода операции** и нескольких **адресных полей**, содержащих адреса расположения операндов команды. Общий вид команды:

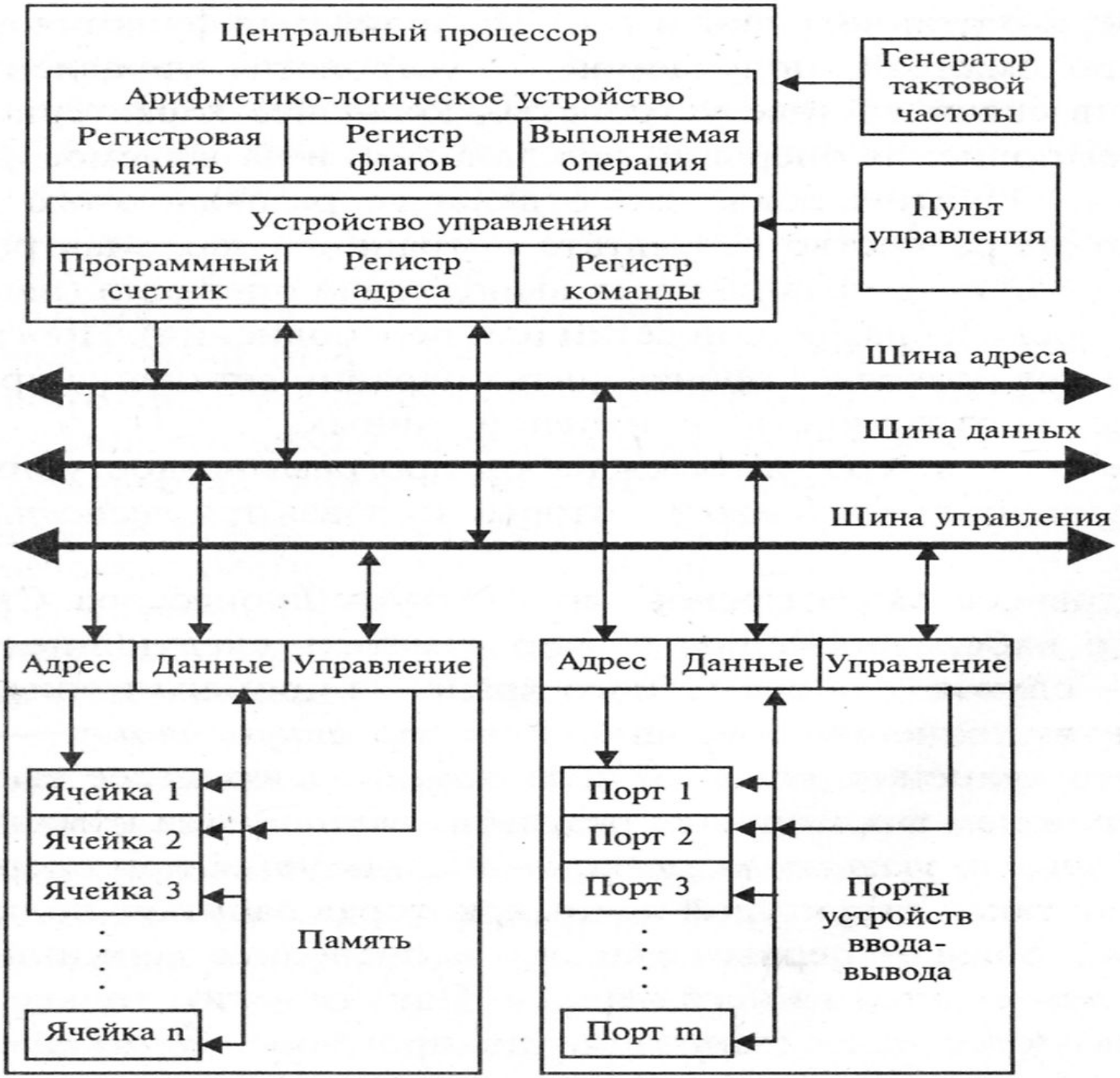
$$\mathbf{К\ А_1\ А_2\ А_3,}$$

- где **К** – двоичный код операции, содержащий указание на операцию (сложение, вычитание, сдвиг и т.д.),
- $\mathbf{А_1}$ -двоичный адрес **первого операнда**, $\mathbf{А_2}$ - двоичный адрес **второго операнда**, $\mathbf{А_3}$ - двоичный адрес для размещения **полученного результата**.

- Например, для 4-х разрядного процессора эта команда может иметь вид:

0001 0011 1100 1001

- *Способ* вычисления адреса по информации, содержащейся в *адресном поле команды*, называется *режимом адресации*.
- *Множество команд*, адресованных в данной ЭВМ, образует ее *систему команд*.
- Система команд, режим адресации, формат данных, набор программно-доступных регистров составляют понятие *«архитектура » ЭВМ*.


Центральный процессор
Арифметико-логическое устройство
Регистровая память
Регистр флагов
Выполняемая операция
Устройство управления
Программный счетчик
Регистр адреса
Регистр команды
Генератор тактовой частоты
Пульт управления
Шина адреса
Шина данных
Шина управления
Адрес
Данные
Управление
Ячейка 1
Ячейка 2
Ячейка 3
Ячейка n
Память
Адрес
Данные
Управление
Порт 1
Порт 2
Порт 3
Порт m
Порты устройств ввода-вывода

- **Арифметико – Логическое Устройство** выполняет арифметические и логические операции.
- *Выполняемая операция* определяется *микрокомандой*, получаемой от **Устройства управления**.
- **АЛУ** имеет ***устройство***, *характеризующее результат выполнения* операции над данными и называемое *флаговым регистром*. Отдельные разряды этого регистра указывают на равенство результата нулю, на знак результата операции (+или-), на правильность выполнения операции (наличие переноса или переполнения). Программный анализ флага позволяет производить операции ветвления программы в зависимости от конкретных значений данных.
- **АЛУ** имеется набор программно-доступных быстродействующих ячеек памяти, которые

- Среди обязательного набора регистров можно отметить следующие:

1. ***Регистр данных*** – служит для временного хранения промежуточных результатов при выполнении операций.
2. **Регистр- аккумулятор** – регистр временного хранения, который используется в процессе вычислений (например, в нем формируется результат выполнения команды умножения).
3. **Регистр указатель стека** – используется при операциях со стеком, стеки используются для организации подпрограмм.
4. **Индексные, указательные и базовые** регистры - используются для хранения и вычисления адресов операндов в памяти.
5. **Регистры-счетчики** используются для организации циклических участков в программах.
6. **Регистры общего назначения** - имеются во многих ЭВМ, их назначение определяет программист при написании программы.
7. **Внутренние системные регистры**, не доступные программно и используемые во время внутренних пересылок информации при выполнении команд.

- **Управляющее Устройство** (УУ) вырабатывает распределенную во времени и пространстве последовательность внутренних и внешних управляющих сигналов, обеспечивающих выборку и выполнение команд.
- На этапе выборки УУ анализирует команду, выбранную из программной памяти.
- На этапе выполнения команды в соответствии с типом реализуемой операции УУ формирует требуемый набор микрокоманд низкого уровня (пересылка данных, сдвиг данных, установка и анализ признаков, запоминание результатов и др) для АЛУ и других устройств.
- В простейшем случае УУ имеет в своем составе три устройства:

1. **регистр команды** - содержит код команды во время ее выполнения;
2. **программный счетчик** - в нем содержится адрес очередной подлежащей выполнению команды;
3. **регистр адреса**, в котором вычисляются адреса операндов, находящихся в памяти.

- **Память** – устройство, предназначенное для запоминания, хранения и выборки программ и данных.
- Память состоит из конечного числа ячеек, каждая из которых имеет свой уникальный номер или адрес.
- Доступ к ячейке осуществляется указанием ее адреса.
- Память способна выполнять два вида операций над данными – **чтение** с сохранением содержимого и **запись** нового значения со стиранием предыдущего.
- Память должна иметь большую емкость. С другой стороны, память должна обладать достаточным быстродействием. Чем больше емкость памяти, тем медленнее доступ к ней.

- **Периферийные устройства (ПУ)**. Различают **ПУ** двух типов: устройства ***внешней памяти***, предназначенное для долговременного хранения данных большого объема и программ, и ***коммуникационные устройства,*** предназначенные для связи ЭВМ с внешним миром (с пользователем, другими ЭВМ и т.д.).

- Обмен данными с внешним устройством осуществляется через **порты ввода-вывода**. (В переводе с англ. Port – ворота).
- По аналогии с ячейками памяти порты можно рассматривать как **ячейки**, через которые можно записать в **ПУ**, или, наоборот – прочитать из него.
- Так же, как и ячейки памяти, порты имеют **адреса** портов ввода-вывода.

- **Система шин**. Совокупность трех шин называют системной шиной.
- Это **шина данных**, по которой осуществляется обмен информацией между блоками ЭВМ, **шина адреса**, используемая для передачи адресов и **шина управления** для передачи управляющих сигналов.
- Состав и назначение шин, и правило их использования, виды передаваемых по шине сигналов и другие характеристики шины могут существенно различаться у разных видов ЭВМ.
- **Шириной шины** называется количество линий (проводников), входящих в состав шины. Ширина шины адреса определяет размер адресного пространства ЭВМ.

Функционирование ЭВМ с шинной структурой можно описать следующим обобщенным алгоритмом

Функционирование любой фон-неймановской ЭВМ представляет собой последовательность достаточно простых действий, выполняемых с очень большой скоростью

- **Система команд**. Все разнообразие решаемых на ЭВМ задач реализуется с помощью небольшого набора очень простых команд.
- Система команд у типичной ЭВМ включает в себя всего **60-150** базовых команд. Все команды в основном служат для выполнения очень простых действий, таких, как прочитать, запомнить, сложить, сдвинуть, сравнить и т.д.
- Интеллектуальность ЭВМ достигается за счет того, что ЭВМ способна выполнять эти команды с огромной, не достижимой для человека скоростью.

- По функциональному назначению команды можно классифицировать следующим образом:
- **1. Команды передачи данных**
- Передача кодов между регистрами внутри процессора, из регистров процессора в память, из памяти в регистры процессора, из одних ячеек памяти в другие и передачи данных между процессором и портами внешних устройств. Отдельную подгруппу составляют команды работы со стеком.

2. Команды обработки данных

- Подразделяются на арифметические (сложить, вычесть, умножить и т.д.), логические (операции И, ИЛИ, НЕ и т.д.) и команды сдвига.
- К арифметическим командам относят также и команды сравнения. Обычно **для сравнения** двух чисел процессор выполняет **операцию вычитания**. По результату вычитания устанавливаются флаги во флаговом регистре процессора.
- Если сравниваемые величины равны, результат вычитания будет нулевым и во флаговом регистре установится флаг нулевого результата. Если вторая из сравниваемых величин больше (A>B)– результат вычитания будет отрицательным и установится флаг отрицательного результата.

3. Команды передачи управления

• Используются для изменения естественного порядка следования команд и организации циклических участков в программах. Простейшей командой передачи управления является команда безусловного перехода (**go to** <**адрес**>), которая загружает адрес перехода, указанный в команде, в программный счетчик **УУ**.

•Команды условного перехода проверяют указанное в команде условие и модифицируют программный счетчик, если условие истинно.

4. Команды для работы с подпрограммами.

- В практике программирования широко используется такой прием, как организация подпрограмм (п/п).
- П/п описывается один раз, а вызываться может из различных мест программы; после выполнения подпрограммы управление должно быть передано команде, следующей за командой вызова п/п, адрес этой команды называется адресом возврата.
- Адрес возврата хранится в одном из регистров процессора.

В ЭВМ для организации стека выделяется **область оперативной памяти**, а для ее адресации и доступа к стеку используется регистр процессора – **указатель стека**. Указатель стека хранит адрес ячейки памяти, содержащей последнее помещенное в стек значение. При записи числа в стек указатель стека модифицируется так, чтобы он указывал на следующую свободную ячейку, и в нее записываются адреса возврата п/п. При извлечении из стека данные считываются из ячейки, на которую указывает указатель стека, затем указатель стека модифицируется так, чтобы указывать на предпоследнее запомненное значение.

Команды вызова п/п **CALL <адрес>** *работают следующим образом*. Когда процессор считывает из памяти команду вызова п/п, программный счетчик увеличивается и показывает на команду, следующую за командой **CALL <адрес>**. При выполнении обращения к п/п процессор сохраняет содержимое программного счетчика в стеке. Адрес, с которого начинается п/п, вычисляется процессором по адресному полю команды вызова п/п CALL <адрес> и помещается в программный счетчик. Процессор приступает к выполнению п/п. Если п/п в процессе своей работы вызывает другую п/п, новое значение адреса возврата будет также включено в стек поверх старого адреса возврата. Для возврата из п/п в основную программу служат команды возврата **RETURN**. Команды возврата из п/п извлекают из стека верхний элемент и помещают его в программный счетчик. Если имели место несколько вложенных вызовов п/п, то возврат произойдет по адресу возврата, сохраненному после последнего вызова.

Вопрос 11. Общие сведения о текстовых, табличных процессорах и базах данных

- В состав пакета Microsoft Office входят следующие приложения Windows:
- WORD - текстовый процессор;
- EXCEL - табличный процессор;
- POWER POINT - презентации, реклама;
- ACCESS - создание и работа с базами данных;
- OUTLOOK - программный комплекс, позволяющий связаться с внешним миром;

Общие сведения о текстовом процессоре Word

Microsoft Word позволяет вводить, редактировать, форматировать, оформлять текст и размещать его на странице. С помощью этой программы можно автоматически исправлять орфографические и грамматические ошибки.

Если при вводе текста делается опечатка, функция автокоррекции автоматически ее исправляет.

Если таблица в документе содержит цифровые данные, то их легко превратить в диаграмму.

Общие сведения о текстовом процессоре Word

- **В процессоре Word** реализованы возможности новейшей технологии связывания и встраивания объектов OLE, которые позволяют включать в документ текстовые фрагменты таблицы, графику, иллюстрации, подготовленные в других приложениях Windows. Встроенные объекты можно редактировать средствами этих приложений.
- **Word –** одна из первых общедоступных программ, которая позволяет выполнять многие операции верстки, свойственные профессиональным издательским системам.

Общие сведения о текстовом процессоре Word

- **Word –** это уникальная коллекция оригинальных технологических решений. Среди таких решений – система готовых шаблонов и стилей оформления, изящные приемы создания и модификации таблиц, функции авто текста и авто коррекции, форматная кисть, пользовательские панели инструментов, макроязык и многие-многие другие.

Общие сведения о текстовом процессоре Word

Вот некоторые из ряда функций Microsoft Word:

автотекст – для хранения и вставки часто употребляемых слов, фраз или графики;
стили – для хранения и задания сразу целых наборов форматов;
слияние – для создания серийных писем, распечатки конвертов и этикеток;
макросы – для выполнения последовательности часто используемых команд;
“**мастера**” – для создания профессионально

Общие сведения о табличных процессорах

Таблица – форма организации данных по строкам и столбцам.

Электронная таблица – компьютерный эквивалент обычной таблицы.

Табличный процессор – комплекс программ, предназначенных для создания и обработки электронных таблиц.

Электронная таблица – самая распространенная и мощная технология для профессиональной работы с табличными данными. Главное достоинство ЭТ – возможность автоматического пересчета всех данных, связанных формульными зависимостями, при изменении значения любого компонента таблицы.

Общие сведения о табличном процессоре EXCEL

Табличный процессор MS Excel позволяет:

1. Решать математические и экономические задачи: выполнять разнообразные табличные вычисления, строить графики и диаграммы и т.п.;
2. Осуществлять исследование (Что будет, если? Как сделать, чтобы?);
3. Проводить статистический анализ;
4. Реализовать функции базы данных – ввод, поиск, сортировку, фильтрацию (отбор) и анализ данных;

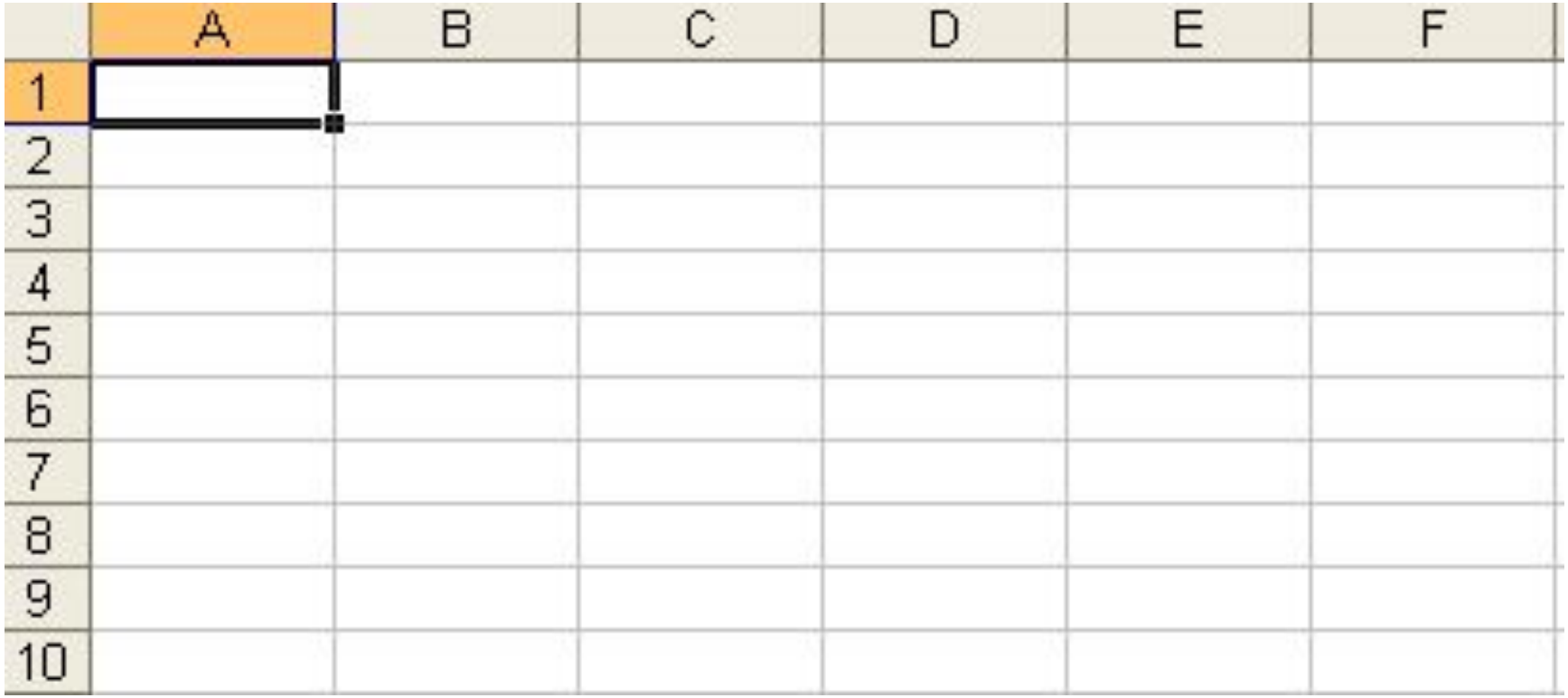
Общие сведения о табличном процессоре EXCEL

5. Устанавливать защиту на отдельные фрагменты таблицы, делать их невидимыми;
6. Наглядно представлять данные в виде диаграмм и графиков;
7. Вводить и редактировать тексты;
8. Осуществлять обмен данными с другими программами, например, вставлять текст, рисунки, таблицы, подготовленные в других приложениях;
9. Осуществлять многотабличные связи.

Общие сведения о табличном процессоре EXCEL

- **Адрес ячейки** – определяется пересечением столбца и строки (A1, F123, AC72);
- **Активная ячейка** – выделенная рамкой, с ней можно производить какие-либо операции;
- **Диапазон (блок) ячеек** – выделенные смежные ячейки, образующие прямоугольный участок таблицы;
- **Адрес диапазона (блока) ячеек** - определяется адресом верхней левой и нижней правой ячейки, разделенных двоеточием (:), B2:C7 → B2, B3, B4, B5, B6, B7, C2, C3, C4, C5, C6, C7.

Общие сведения о табличном процессоре EXCEL



Фрагмент рабочего поля Excel.
Лист книги – рабочее поле, Книга (файл.xls) – содержит заданное количество листов.

Решение системы линейных уравнений

(применяется, в частности, для нахождения коэффициентов уравнений регрессии, используемых для прогнозирования экономических процессов)

$$3x_1 - 2X_2 + X_3 = 8$$
$$X_1 + X_2 + 3X_3 = 9$$
$$2X_1 + X_2 + 3X_3 = 10$$

Алгоритм решения: $\mathbf{Ax = b} \rightarrow \mathbf{x = A^{-1} b}$,

при условии, что определитель матрицы $A \neq 0$.

$$A = \begin{matrix} 3 & -2 & 1 \\ 1 & 1 & 3 \\ 2 & 1 & 3 \end{matrix} \qquad b = \begin{matrix} 8 \\ 9 \\ 10 \end{matrix}$$

Решение системы линейных уравнений

	A	B	C	D	E	F
1	3	-2	1		8	
2	1	1	3		9	
3	2	1	3		10	
4						-7
5						
6						
7						
8						

Набираем в ячейке F4 =МОПРЕД(A1:C3) и нажимаем клавишу **Enter**, получим -7. Кнопка f_x показывает список встроенных ф-ций.

Выделяем блок A6:C8, набираем =МОБР(A1:C3) и нажимаем комбинацию клавиш **Ctrl+Shift+Enter**, получим обр. матрицу:

Решение системы линейных уравнений

	A	B	C	D	E	F
1	3	-2	1		8	
2	1	1	3		9	
3	2	1	3		10	
4						-7
5						
6	-2,8E-17	-1	1		X1=	1
7	-0,42857	-1	1,142857		X2=	-1
8	0,142857	1	-0,71429		X3=	3

Выделяем блок F6:F8, набираем =МУМНОЖ(A6:C8;E1:E3), нажимаем комбинацию клавиш **Ctrl+Shift+Enter**, получим искомые корни системы уравнений. Добавляем комментарии в ячейки E6, E7 и E8.

Решение системы линейных уравнений

Аналогично решаются системы линейных уравнений других размерностей.

Примечание. Ориентация выделяемого блока ячеек (горизонтальная или вертикальная) для искомых корней системы уравнений имеет важное значение.

Функция **МУМНОЖ**(X;Y) означает умножение **X** на **Y**, где **X** и **Y** матрицы, векторы-столбцы или векторы-строки.

В общем случае умножение матрицы **X** на матрицу **Y** подчиняется следующему правилу размерностей:

$$\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$$

$$(\mathbf{m}\times\mathbf{n})\ (\mathbf{m}\times\mathbf{k})\ (\mathbf{k}\times\mathbf{n}),$$

где **m** – количество строк матрицы **Z** и **X**; **n** – количество столбцов матрицы **Z** и **Y**; **k** – количество столбцов матрицы **X** и количество строк матрицы **Y**.

Решение системы линейных уравнений

В нашем примере мы использовали алгоритм

$$\mathbf{x} = \mathbf{A}^{-1} \times \mathbf{b}$$

(3×1) (3×3) (3×1),

т.е. при умножении матрицы $\mathbf{A}^{-1}$ размерности (3×3) на вектор-столбец **b** размерности (3×1) мы получаем вектор-столбец **x** размерности (3×1). Таким образом, мы правильно выделили вертикальный столбец F6:F8 для искомых корней системы уравнений.

Общие сведения о базах данных

Существуют иерархические, сетевые, реляционные и объектно-ориентированные модели баз данных.

1. **Иерархические базы данных** имеют форму перевернутых деревьев с дугами-связями и узлами-элементами данных. Иерархическая структура предполагает неравноправие между данными – одни жестко подчинены другим.

Подобные структуры, безусловно, четко удовлетворяют требованиям многих, но далеко не всех реальных задач.

Общие сведения о базах данных

- **2. Сетевая модель данных.** В сетевых БД наряду с вертикальными реализованы и горизонтальные связи. Однако унаследованы многие недостатки иерархической и главный из них, необходимость четко определять на физическом уровне связи данных и столь же четко следовать этой структуре связей при запросах к базе.

Общие сведения о базах данных

3. **Реляционная модель. Во-первых**, все данные в модели представляются в виде таблиц. Реляционная модель – единственная из всех обеспечивает единообразие представления данных. И сущности, и связи этих самых сущностей представляются в модели совершенно одинаково – ***таблицами***. Правда, такой подход усложняет понимание смысла хранящейся в базе данных информации, и, как следствие, манипулирование этой информацией.

Общие сведения о базах данных

Избежать трудностей манипулирования позволяет **второй элемент** модели – реляционно-полный язык (отметим, что язык является неотъемлемой частью любой модели данных). Например, язык SQL. Он описывает любой запрос в виде операций с таблицами, а не с их строками. **Третий элемент** реляционной модели требует от реляционной модели поддержания некоторых ограничений целостности. Одно из таких ограничений утверждает, что каждая строка в таблице должна иметь некий уникальный идентификатор, называемый ***первичным***

Общие сведения о базах данных

4. Объектно-ориентированная модель.
Новые области использования вычислительной техники потребовали от баз данных способности хранить и обрабатывать новые объекты – текст, аудио- и видеоинформацию, а также документы. Основные трудности объектно-ориентированного моделирования данных проистекают из того, что такого развитого математического аппарата, на который могла бы опираться общая объектно-ориентированная модель данных, не существует. В большой степени поэтому до

Общие сведения о базах данных

Несмотря на преимущества объектно-ориентированных систем – реализация сложных типов данных, связь с языками программирования и т.п. – на ближайшее время превосходство реляционных БД и соответственно реляционных СУБД гарантировано.

Общие сведения о базах данных

При разработке приложений СУБД используются два подхода: ***проектирование сверху вниз,*** при котором разработка приложения начинается с определения основных функций и задач, и ***проектирование снизу вверх,*** при котором сначала проводится анализ данных и определение их структуры. Сначала определяются задачи и выполняется их группировка (подход «сверху вниз»). Затем для связанных задач собираются все поля данных и начинается процесс формирования объектов (элементы подхода «снизу вверх»).

Общие сведения о базах данных

После определения задач нужно составить для каждой из них описание данных, необходимых для ее решения. Каждый элемент данных, по характеру его использования, может быть классифицирован одним из следующих признаков: входной, выходной, изменяемый, удаляемый и вычисляемый.

Общие сведения о базах данных

Для устранения дублирующихся данных используется процесс, называемый ***нормализацией.*** Он позволяет организовать поля данных в группы таблиц. Лежащая в основе нормализации математическая теория довольно сложна, но для практического применения ее можно сформулировать в виде довольно простых правил.

Правило 1: Каждое поле любой таблицы должно быть уникальным.

Общие сведения о базах данных

Правило 2. Первичные ключи

База данных хорошо спроектирована тогда, когда каждая запись в любой таблице является уникальной. Это означает, что значение некоторого поля (или нескольких полей) не повторяется ни в одной записи в таблице. Такой идентификатор называется ***первичным ключом*** (или просто ключом).

Общие сведения о базах данных

Правило 3. Функциональная зависимость

Если вы определили для каждой таблицы первичный ключ, то можно проверить, включены ли в таблицы все сведения, относящиеся к соответствующим объектам. Кроме того, следует убедиться, что каждое поле функционально зависит от первичного ключа таблицы.

Для каждого значения первичного ключа значения в столбцах данных должны относиться к объекту таблицы и полностью его описывать.

Общие сведения о базах данных

Правило 4. Независимость полей

И наконец, четвертое правило позволяет проверить, не возникнут ли у вас проблемы при изменении данных в таблицах.

Изменение значений любого поля (не входящего в первичный ключ) должно выполняться без воздействия на данные других полей.

Общие сведения о базах данных

Чужие ключи. Если при создании новой таблицы в существующую таблицу включается поле, связывающее старую и новую таблицы, то это «связующее» поле называется ***чужим ключом*** или ***внешним ключом***.

В процессе проектирования нужно внимательно следить за созданием чужих ключей. Задаваемые при создании таблиц в Microsoft Access связи первичных ключей с чужими ключами используются для объединения данных из нескольких таблиц.

Соответствие между таблицей и реляционной таблицей базы данных

Столбец таблицы – поле БД;

Название столбца таблицы – имя поля БД;

Ширина столбца таблицы – длина поля БД;

Тип данных в столбце таблицы – тип данных поля БД (текстовый, числовой, дата и др.);

Строка таблицы – запись БД.

Первичный ключ - выбранное поле в котором совпадения заведомо исключены.

Внешний ключ используется для связывания таблиц.

Примеры ключей БД

1-ая рел. табл. «Студент» Базы данных содержит следующие поля:

(Номер зачетной книжки, Фамилия, Имя, Отчество, Факультет, Курс, Группа, Код вуза)

2-ая рел. табл. «Вуз» Базы данных содержит поля:

(Код вуза, Название, Адрес, Телефон студенческого отдела кадров)

Поле Номер зачетной книжки –первичный ключ для рел. таблицы «Студент».

Поле Код вуза – первичный ключ для рел. таблицы «Вуз» и внешний ключ для рел. таблицы «Студент».