

1. Этапы развития человеческой цивилизации	2
2. Формы обработки информации и каналы ее получения.....	4
3. Сложности понятия информации.....	5
4. Существующие единицы измерения информации (самое большое и самое маленькое), понятие информации – смысловое и в единицах измерения.....	6
5-6. Этапы развития информационных и компьютерных технологий.....	7
7. Специалисты, использующие ЭВМ.....	11
8. Этапы развития языков программирования.....	12
9. Системы диалога.....	13
10. Интеллектуальные программные среды.....	14
11. Современные устройства ввода и вывода.....	15
12. Общие характеристики современных информационных и компьютерных и технологий. Правила работы с информацией.	16
13. Типы информации. Классификация современных компьютеров.....	17
14. Этапы развития современной цивилизации. 4 этапа. Идеи. Революции. Этапы разбиты по билетам.	19
15. Характеристики современного этапа развития общества. Анализ. Капитал, деньги, человек и другое. Формула эффективности развития общества.....	22
16. Аддитивные и позиционные системы счисления. Формы представления любого числа.....	24
17. Распространённые системы счисления.....	26
18. Экономичность систем счисления.....	26
19. Кодирование информации.....	28
20. Понятие бита. Понятие адреса машинного слова. Форматы данных в компьютере.....	29
21. Числа с фиксированной точкой. Представление в машинном слове компьютера.	30

22. Вещественные числа. Представление в машинном слове компьютера. Понятие машинного и математического порядков.	31
23. Измерение информации. Формула Хартли. Связь с вероятностью. Формула Шеннона.	32
24. Алфавитный подход к измерению информации. Кибернетика.	34
25. Алгоритмы Маркова.	35
26. Машина Тьюринга.	37
27. Структура программы на C++	38
28. Этапы обработки программ.	39
29. Типы данных в C++. Спецификаторы типов данных.	40
30. Switch. Структура. Пример задачи.	44
31. Виды циклических процессов.	44
32. Методы решения нелинейных уравнений.	46
33. Операторы циклических процессов.	49

1. Этапы развития человеческой цивилизации

Этапы	Развитие науки	Развитие техники	Ресурсы
Освоение материи -> оседлая жизнь	Осознание единой вещественной природы мира	Орудия труда, рычаг, колесо, крыло	Окружающий мир, полезные ископаемые, приручение животных
Освоение энергии -> рождение индустриального общества	Осознание единой природы, материи и энергии, как движение материальных объектов	Накопление (аккумуляторы) Преобразование (трансформаторы) Передача (сети)	Солнце, ветер, вода, огонь, уголь, нефть, газ
Освоение информации -> освоение инфопространства	Осознание единой природы: материи, энергии, информации	Накопление (диски, фото) Передача Обработка (линейные операции) Хранение (книги)	Документы о квалификации (аттестат), авторское право, патенты (машина Зингера), книга рекордов Гиннеса

2. Формы обработки информации и каналы ее получения

Формы:

- **Вычисление:** обработка с применением формул, рисунков, диаграмм, нотных знаков. Обычно по ним вычисляют либо количество чего-то, либо часть от общего количества (математические формулы).
- **Логические преобразования:** обработка основано на принципе «если-то». Применение в криминалистике, алгоритмических языках, на фондовых торгах (Шерлок Холмс)
- **Изменение формы информации, но не содержания:** применяется в системах перевода, переводах в разные СС, системах шифрования информации, а также в системах проверки.
- **Сортировка:** расположение сообщений в заданной порядке (телефонная книга, словарь)
- **Поисковые системы обработки информации:** поиск заданного слова, символа, графика (интернет)

Каналы:

- ✓ Внутренний канал. Всегда связан с органами чувств: зрение, обоняние, слух, вкус, осязание
- ✓ Внешний канал. СМИ, телефон, почта

3. Сложности понятия информации

Информация – система знаков, которая несет смысловую нагрузку.

Информация – сообщение, уменьшающее неопределенность знания об исследуемом объекте.

- Информация должна обладать свойством **новизны** ($2*2=4$)
- Информация должна быть **понятна** получателю (логарифм 8)
- **Объем** получаемой информации:

- 1) Чем ниже вероятность события, тем больше объем (цунами в Перми или в Японии)
- 2) Чем выше неожиданность, тем больше объем (заболеть летом или заболеть зимой)
- 3) Чем выше заинтересованность, тем больше объем (снимок черной дыры)
- 4) Чем выше последствия получающего, тем больше объем (Бабушка попала в больницу)

Восприятие информации человеком:

- 1) Знаковая форма (тексты, звуки, ноты, иероглифы)
- 2) Образная форма (сравнения)

4. Существующие единицы измерения информации. Самое большое и самое маленькое число. Понятие информации – смысловое и в единицах измерения.

Информация – система знаков, которая несет смысловую нагрузку.

Информация – сообщение, уменьшающее неопределенность знания об исследуемом объекте.

Единицы измерения:

Бит – количество информации, которое может поместиться в одном элементе памяти.

Бит – количество информации, уменьшающее неопределенность знания в 2 раза.

Бит

Байт

Килобайт

Мегабайт

Гигабайт

Терабайт

Петабайт

Эксабайт

Зеттабайт

Йоттабайт

Самое большое число: гуголплекс

$10^{(10^{100})}$

5-6. Этапы развития информационных и компьютерных технологий

Этап	Компьютер (КТ)	Решаемые задачи	Информация (ИТ)
1 60-70	<p>Большие ЭВМ (лампы)</p> <p>- огромные</p> <p>- громоздкие устройства вв/в: перфокарты, перфоленты</p> <p>- высокая стоимость</p> <p>- элементарная база - лампы</p> <p>Лампы: нагревались → финансовые расходы на э/э; низкая надежность → дублирование элементов; требовалось охлаждение</p>	<p>1. Инженерные расчеты</p> <p>2. Военные дела</p> <p>3. Кодирование информации</p> <p>4. Задачи сложных математических вычислений</p> <p><i>Режим работы ОС:</i></p> <p><i>Индивидуальный</i></p> <ul style="list-style-type: none"> <i>загрузка всех устройств машины одной задачей</i> <i>следующая задача ожидает завершения предыдущей</i> <i>решение задачи занимает много времени</i> 	<p>ЯНУ (автокод, машинный код, ассемблер) – совокупность 0 и 1</p> <p>ЯНУ владели программисты</p>
2 70-85	<p>Мини- и микро- ЭВМ (полупроводники)</p> <p>- исчезло охлаждение и дублирование элементов</p> <p>Уменьшились габариты →</p>	<p>1. Оптимальное распределение материалов, ресурсов и управления (в производстве)</p> <p>2. Аналитические задачи (выбор из множества решений)</p>	<p>ЯВУ (Фортран, Паскаль)</p> <p>- логические связки (если-то, иначе...)</p> <p>- нет неподготовленных пользователей</p> <p>- время всеобщей компьютерной</p>

	<p>уменьшилась</p> <p>стоимость</p> <p>- увеличилось</p> <p>быстродействие (500 тыс. оп./сек)</p> <p>•магнитные носители(диски)</p> <p>•визуализация через <u>дисплей</u></p> <p>• ввод через <u>клавиатуру</u></p>	<p><i>Режим работы ОС:</i></p> <p><i>Мультипрограммный</i></p> <p>- <i>Короткие задачи идут вперед, на основе приоритета</i></p> <p>- <i>Режим экономичен за счет параллельного использования узлов машины.</i></p>	<p>грамотности (особенно</p> <p>выращивание класса программистов)</p> <p>Транслятор - программа, которая переводит задачу на машинный язык.</p>
3 80-95	<p>Индивидуальные ПК (интегральные схемы)</p> <p>- снижение габаритов компьютера до уровня настольного прибора</p> <p>- уменьшение стоимости в 100 раз</p> <p>- увеличение надежности и снижение числа поломок и отказов</p> <p>- быстродействие 10^8 операций в секунду</p> <p>- увеличение оперативной памяти в 1000 раз за счет появления внешних</p>	<p>Появление игр</p> <p>Решение графических задач</p> <p>Работа с большими БД (здесь же появление первых БД)</p> <p>АРМ специалиста могла заменить специалиста в опасных производствах, задачах с особыми требованиями к внимательности, задачи с аналитикой больших данных.</p> <p><i>Режим работы ОС: Режим разделения времени - терминалы</i></p> <p>- <i>большая мощность ПК</i></p> <p>- <i>многопользовательский</i></p> <p>- <i>короткий такт обслуживания (20 мс)</i></p>	<p>1. Внешние языки (приближенные к языку человеческого общения - трансляторы)</p> <p>а) пакеты прикладных программ – математические – привязка к конкретной области знаний</p> <p>б) инженерные пакеты – расчет схем, чертежей, применение таблиц</p> <p>2. Полное исчезновение оператора</p> <p>3.Появление систем диалога</p>

	<p>устройств хранения информации (дискеты, CD)</p> <p>Появление <u>мышки</u> – обеспечение контроля над процессами за счет посылы специальных команд управления</p> <p>Появление <u>сенсорного управления</u></p>	<p><i>каждого пользователя</i></p>	
4 с 95-х	<p>Сетевые компьютеры (большие интегральные схемы)</p> <ul style="list-style-type: none"> - уменьшение габаритов до наладонного гаджета - уменьшение стоимости - увеличение операционной памяти в миллион раз за счет использования выносных устройств <p>Появление специальных каналов</p>	<p>Появление датацентров, которые ищут и обрабатывают данные, анализируют связи существующих отношений в сети (авторское право, разведка, экономика).</p> <p>Решение задач когнитивного характера, которые через моделирование вычисляют дальнейшее направление развития человеческой цивилизации.</p> <p><i>Режим работы ОС: Двойной (Индивидуальный (1 ч);</i></p>	<p>1. Искусственные языки (копирование языка человеческого общения)</p> <p>2. Появление программных сред (превращает задачу в проект, где каждый шаг описан в виде имиджа – формы словесного описания, графика, рисунка). Программа создается с помощью сборочной технологии (лего).</p> <p>3. Интеллектуальные программные среды</p> <p>А) системы анализов текстов (выявление</p>

<p>для обеспечения коммуникации в сети: <u>каналы сотовой связи</u>; <u>оптоволоконные</u>; <u>беспроводные</u> <u>каналы связи</u> <u>Появление устройств</u> <u>ввода и вывода</u>: сканеры; контроллеры; ультразвуковые устройства (УЗИ, медицина), трекболы, снятие гальванического потенциала, сканирование движения глазных яблок, считывание QR</p>	<p>Сетевой (2 ч))</p>	<p>резюме) Б) Генераторы технических решений (направление развития технической мысли, подталкивание к решениям проблем современной цивилизации) В) системы моделирования 4. Исчезновение любых посредников между человеком и ЭВМ (программисты и системотехники занимаются разработкой новых операционных систем, интерфейсов)</p>
--	-----------------------	--

7. Специалисты, использующие ЭВМ.

- Неподготовленные пользователи (математики, химики, военные) – делают постановку задачи, не знают возможностей ЭВМ.
- Программисты – знают возможности ЭВМ, переводят задачу на язык машины.
- Системотехники – автоматизируют системы, знают возможности ЭВМ, разбивают задачу на МЕЛКИЕ шаги (алгоритм).
- Оператор – самый неквалифицированный из всех специалистов, выполняет рутинную работу, связанную с вводом и выводом информации, а также с подготовкой информационных носителей.
- Подготовленный пользователь – изучают возможности ЭВМ, умеют разбить задачу на КРУПНЫЕ шаги.

8. Этапы развития языков программирования

Этап	Уровень развития
1 60 – 70	Языки низкого уровня: ассемблеры, автокоды, машинные коды
2 70 – 85	Языки высокого уровня: логические связки (if/else); появление транслятора (каждый ЯВУ имеет свой)
3 85 – 95	Внешние языки (пакеты) – приближены к языку человеческого общения (пакет прикладных программ; инженерные пакеты)
4 с 95-х	Искусственные языки – наиболее приближены к языку человеческого общения

9. Системы диалога

- Активный диалог: человек задает вопрос, а компьютер отвечает, пользуясь сложными алгоритмами поиска ответа на вопрос

- Пассивный диалог: компьютер дает варианты, а человек выбирает ответ

Требования к диалогу:

1. Ясность построения вопроса и ответа
2. Отсутствие синонимов
3. Удобство создания диалога и его использования
4. Время реакции (норма: от 4 сек до 2 мин, т.к. сохраняется эффект диалога + работа с психологическими факторами)

10. Интеллектуальные программные среды

4 этап (ИТ):

1. Анализ текста (выявление резюме)
2. Генераторы технических решений (определяют направление технического развития, типа мозгового штурма)
3. Системы моделирования:
 - 3.1 Ситуационное (если..., то...)
 - 3.2 Имитационное (исследование реальной системы в виде математических уравнений)
 - 3.3 Когнитивное (прогнозы валют, акций)

11. Современные устройства ввода и вывода

- 1) Большая мышь для ног
- 2) Трекбол
- 3) Считывание QR-кодов и штрихкодов
- 4) Лазерные щупы (для измерения недоступных объектов)
- 5) VR
- 6) Снятие гальванического потенциала кожи
- 7) Сканирование движения глазных яблок
- 8) Интерактивная доска
- 9) Беспроводные наушники
- 10) Проектор
- 11) Графический планшет
- 12) Голосовой ввод и вывод (нейросети)
- 13) Тачскрин
- 14) Голография

12. Общие характеристики современных информационных и компьютерных технологий. Правила работы с информацией.

1) Любая постановка задачи должна быть определена в *общепринятых терминах*, которые понятны и заказчику и исполнителю. Заказчик всегда знает, что он хочет, но он не знает, как этого достичь.

2) Любой проект проходит *стадию моделирования*, в которой участвует заказчик и исполнитель, но реальность сложнее: современные модели можно оперативно изменять (структуру, параметры, систему связей между элементами моделей)

3) Результат получается там, где получена хорошая модель. В современных проектах *визуализация модели* обязательна (диаграммы, картинки, символы).

4) Проклятие размерности: строится модель реальной системы, и заказчик на новом шаге разработки нагружает модель сложными дополнительными функциями. Задача специалиста: *отсортировать функции*.

5) Немонотонность человеческой логики: решение в острых ситуациях. В алгоритмической модели должны быть заложены заранее предусмотренные действия (*система защиты*).

6) В реальной ситуации бывает, что данная задача уже кем-то решена. Необходимо *найти готовое решение* и проанализировать его. Применить, если подходит.

13. Типы информации. Классификация современных компьютеров.

1) **Структурная** – обладают все материальные объекты, атрибут материи, *свойство* материи.

Пример: материальный объект (книга, стол)

2) **Оперативная** – используется для поглощения и преобразования информации, содержится в процессе, в результате, *состояние* материи

Пример: дорожный знак (столб – структурная информация, нанесенный знак – оперативная); почтовая открытка, не дошедшая до адресата (структурная информация осталась, оперативная потеряна)

Компьютеры:

➤ Суперкомпьютер. Компьютеры с очень высокой мощностью единичного производства, разработанные для решения специальных задач, связанных с обороноспособностью государства, хранением и кодированием гос. Информации. Выпуск не более 2 штук в год, не подключены к всемирной паутине. США, Германия, Россия, Франция, Китай, Индия, Япония

➤ Mainframe: высокая мощность; интенсивные вычислительные задачи, связанные с хранением, поиском и обработкой информации; многопользовательский режим; высокая стоимость; отраслевое назначение

В России владеют Сбербанк, система ГИБДД совместно с МЧС

➤ Workstation: мощный автономный компьютер; использование отдельным пользователем для решения узкого круга его задач (разработка игр, кинематографии); может использоваться кругом пользователей, которые решают задачи одного класса

➤ Персональный компьютер: массовый унифицированный вычислитель; широкий круг пользователей; снабжение функциями ввода и вывода информации; функция коммуникации и манипулирования данными; хранение и обработка информации; функция доступа и работы с Всемирной паутины

- Ноутбук: ПК до 3 кг: обладает всеми функциями персонального стационарного компьютера
- PDI: карманного и наладонного типа; функция распознавания рукописного ввода; мощные коммуникативные возможности; предназначены для быстрых оперативных задач

14. Этапы развития современной цивилизации. 4 этапа. Идеи.

Революции. Этапы разбиты по билетам.

1. Дикое общество

- отсутствие безопасности жизни
- еда не всегда доступна
- голод → болезни → обнищание населения

Природа сильнее человека: низкая сила и скорость у человека, чем у животных. Опасности ведут к снижению безопасности → человеческая популяция не растет.

НОВАЯ ИДЕЯ:

1. Накопление запасов (дом, огонь: пища, запасы, тепло, боязнь зверей)
2. Поддержание огня → первое разделение труда (воспитатели, охотники)
3. Приручение животных
4. Выращивание растений рядом с домом

2. Оседлость

1. **Аграрная революция (дичь и растения рядом с домом)**

- жизнь каждый год становилась легче и безопаснее за счет запасов
- но может быть неурожай → истощение земли, кражи.

НОВАЯ ИДЕЯ:

1. Поливать, полоть, охранять хозяйство → новые дела и обязанности: вычисления, наблюдения, измерения → появление умения думать.
2. Появление права собственности на еду/запасы, землю → первое понимание собственности.
3. Наемный труд → но разный результат труда.

4. Изучение сходств и различий (особенно по орудиям труда. Они должны быть одинаковы хороши). Вручную не получится. Развитие производства орудий труда. Обмен продуктами труда

2. **Промышленная революция**

- промышленное производство породило интерес к астрономии (изучение времен года), времени, размерности (сезонные культуры, посадки)
- появление понятия "промышленное общество".

3. Промышленное общество

- сосредоточение на людях и производстве
- изобретение машинного производства орудий труда
- Промышленные технологии. Общество - это взаимодействие разных людей с разными профессиями
- Появление городов
- Понимание взаимодействия материи и энергии.

НОВАЯ ИДЕЯ:

1. Другие машины делают машины (появление конвейера)

2. **Индустриальная революция**

- появление стандартов (повышение эффективности труда)
- появление понятия КПД - "затратил меньше, получил больше" за счет узкой специализации, массового производства (появление промышленных гигантов, например форт), массовой образованности населения.

Вывод: эффективность высвобождает ресурсы для более сложных разработок, которые не всегда нужны.

Появление мусорной экономики (70е). Цивилизация перешла невидимую линию, когда $V_{\text{производства}} > V_{\text{потребления}}$ (V - скорость)

4. Индустриальное общество

- решение задач управления обществом

- централизация власти
- укрупнение всего: повышение линий конвейера, усложнение деталей машин...

- пирамидальная структура управления (основа – производство)

Люди, которые производят товары - 1 звено.

Наблюдатели за людьми - 2 звено.

Наблюдатели за наблюдателями - 3 звено.

- рост управленцев по квадратичному закону
- основание пирамиды не может прокормить пирамиду (коллапс цивилизации)

- появление конфликтов и их решений.

- Каждый хочет возглавить эту пирамиду, отсюда войны.

- Идет присвоение материй, энергий, информации.

- массовое образование (ВУЗ)

- искусство (понимание мира, как машины. Изображенное очень знакомо, т.к. реализм)

- гуманитарные науки (изучение человека. Т.к. гуманитариев много, это становится неинтересным обществу)

- наука (антибиотики, контрацептивы, УЗИ, МРТ)

Человек остается неопознанным, но не учитывать человека нельзя ни в одной цивилизации, т.к. в руках человека сложное производство как общественное, так и промышленное (города, системы).

Иначе ухудшение безопасности цивилизации, и технологии не должны быть в руках избранных.

Пирамида рухнула под гнетом слабого управления и нарушения связей.

Решения: упростить структуру, совершенствовать систему.

Человечество обратилось к технологиям. Необходимо усилить ум (сделать его автоматическим). Произошла **4 революция - информационная.**

- через скачок от индустриального общества к информационному.
- появление демократии, вместо централизации власти.

- вместо пирамид - сетевое управление.

15. Характеристики современного этапа развития общества.

Анализ. Капитал, деньги, человек и другое. Формула эффективности развития общества.

Произошла **4 революция - информационная.**

- через скачок от индустриального общества к информационному.
- появление демократии, вместо централизации власти.
- вместо пирамид - сетевое управление.

Каждый сам решает свою жизнь, выстраивает линию поведения, согласовывает цели с обществом.

- Появляются патенты, авторское право.
- перерождение войн (информационные) + психотропные эффекты (подавление воли общества до начала войны)
- люди меняют профессии и приоритеты (уход в аналитику)
- моделирование(!), графика, ЭВМ, разные среды
- развитие визуализации, когнитивных моделей.

Информационное общество - срастание виртуальности и реальности.

Выражается в искусстве

- изучает внутренний мир человека
- выражает мысль автора
- отражает мысли общества

Человек - основной капитал, т.к. обладает разумом. Мог разработать инновацию, чем вел общественное развитие вперед. Большие системы не работают без человека, что делает человека наблюдаемым обществом.

Высока роль денег, т.к. они являются доступом к благам мира, являются удобным обменом.

Основа мира – модели. Мир принадлежит сетям, где данные упорядочены.

Критерий развития технологии – формула эффективности:

$$\frac{\text{результат}}{\text{затраты}} \leftrightarrow \frac{\text{уменьшение рисков}}{\text{увеличение затрат}}$$

Затраты на поддержание существования общества, обустройство окружающей среды, развитие во всех проявлениях.

Только технологии гарантируют результат. Основное мерило цивилизации – технологии.

16. Аддитивные и позиционные системы счисления. Формы представления любого числа.

СС – совокупность элементов, разрешенных в данной СС: знаков, которые называются базис; набора правил, по которым можно образовывать любое другое число в данной СС

- 1) Аддитивные СС (римская) – числа, полученные путем сложения/вычитания базисных чисел.

Достоинство: наглядность

Недостатки: сложность формирования больших чисел; сложность выполнения операций с большими числами

- 2) Позиционные СС (десятичная) – значение знака зависит от позиции в числе.

Достоинства: простое выполнение операций; запись больших чисел небольшим числом знаков

Недостатки: неявная визуализация

Формы представления любого числа:

1. Запись в виде последовательности базисных знаков

$a_n a_{n-1} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-m}$

10101101,1001(2)

2. Запись в виде полинома

$$X = A_n \cdot k^n + A_{n-1} \cdot k^{n-1} + \dots + A_0 \cdot k^0 + \dots + A_{-1} \cdot k^{-1} + A_{-2} \cdot k^{-2} + \dots + A_{-m} \cdot k^{-m}$$
$$A_i \in [0; k-1]$$
$$1545 = 1 \cdot 10^3 + 5 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

17. Распространённые системы счисления.

5 СС:

- анатомия человека (5 пальцев)

12 СС:

- анатомия человека (12 фаланг),
- мореплавание, измерение глубин, 1 фут = 12 дюйм
- денежная система
- измерение объемов жидкостей (дюжина = 12)
- время (вавилон, 12 часов 12 месяцев)

60 СС:

- геометрические измерения (окружность - 360 градусов)
- измерение углов
- измерение времени (1 час = 60 минут и т.д.)

20 СС:

- (древние Кельты = французы)
- система счета ($80 = 4 * 20$)
- денежная система (1 франк = 20 су (ка))

3 СС:

- (экономия памяти)
- мореплавание (азбука морзе - телеграф и интернет каналы, световые сигналы, флажки)

2 СС:

- элементная база компьютера (горит/не горит)

18. Экономичность систем счисления.

Наиболее экономичная та СС, которая позволяет записать с помощью определенного количества заданных знаков наибольшее количество чисел.

$$N = P^{\frac{k}{p}}$$

N - Количество чисел, которые можно составить

P - Основание СС

K - Заданное кол-во знаков

Пример (k = 60):

Для 2 СС:

$$60/2 = 30$$

$$2^{30} = (2^3)^{10}$$

$$2^3 = 8$$

Для 3 СС:

$$60/3 = 20$$

$$3^{20} = (3^2)^{10}$$

$$3^2 = 9$$

Ответ: 3 СС экономичнее

19. Кодирование информации.

Кодирование – переход от исходного представления информации, удобного для человека, к представлению удобному для хранения, передачи и обработки.

Декодирование – обратный процесс

Цели кодирования:

- ☐ Удобство и возможность технической реализации
- ☐ Удобство восприятия
- ☐ Высокая скорость передачи и обработки закодированной информации
- ☐ Экономичность – уменьшение избыточности сообщения
- ☐ Надежность – защита от случайных искажений
- ☐ Сохранность – защита от нежелательного доступа

На разных этапах проекта цели могут противоречить друг другу => информация может перекодироваться

20. Понятие бита. Понятие адреса машинного слова. Форматы данных в компьютере.

Бит – количество информации, которое может помещаться в одном элементе памяти. (Почему появились другие величины? Потому что бит очень мал и не содержит смысловой нагрузки => его стали укрупнять)

ОП компьютера разбита на байты. Порядковый номер байта – адрес. Нумерация адресов байтов идет с нуля.

Байты соединяются в ячейки, которые каждый комп воспринимает как единое целое, которое называется машинное слово => машинное слово – ячейка памяти ЭВМ, которую компьютер воспринимает как единое целое.

Адрес машинного слова в памяти компьютера равен адресу младшего байта, входящего в это слово.

Форматы данных в компьютере:

1) *Логические коды.*

- Размещаются в отдельных байтах

(от 0 до 8: всего $2^8 = 256$ символов: 128 латинский, цифры, знаки препинания, 128 национальные алфавиты, научные символы и др.);

- в машинных словах

(обычно от 0 до N-1 разрядов, N-величина машинного слова. Для 16 бит – 15; символьных величинах (1 сим = 8 бит = 1 байт)

2) *Числа с фиксированной точкой* – для целых чисел.

Диапазон зависит от машинного слова (ячеек памяти) компьютера. Если машинное слово имеет k разрядов (байт), то всего k^2 целых чисел.

3) *Числа с плавающей точкой*

21. Числа с фиксированной точкой. Представление в машинном слове компьютера.

Формат служит для представления целых чисел. Диапазон целых чисел зависит от размера ячеек – машинных слов памяти компьютера, которые используются для хранения целых чисел.

Если для хранения целого числа используется k -разрядное машинное слово, то это значит, что слово содержит k бит памяти. Таким образом, в k -разрядном машинном слове может храниться 2^k различных целых чисел.

Положительное число:

- Число переводится в 2 СС
- Полученный результат дополняется слева незначащими нулями

до размера машинного слова

- Полученное число упаковывается в 16 СС

Отрицательное число:

1. Переводим модуль числа в 2 СС
2. Полученный результат дополняется слева нулями до размера

машинного слова

3. Меняем 0 на 1, и наоборот
4. Прибавляем к числу 1
5. Полученное число упаковывается в 16 СС

22. Вещественные числа. Представление в машинном слове компьютера. Понятие машинного и математического порядков.

Вещественные числа – числа, которые можно записать в виде дроби.

Представление в машинном слове:

- Перевести модуль данного числа в 2 СС, учитывая, что число должно занимать 24 бита, т.к. под мантиссу выделено 3 байта=24 бита
- Нормализуем двоичное число
- Найти машинный порядок по формуле $M(p_2) = p_2 + 10000000$
- Учитывая знак числа в старшем бите 1ого байта, вписать полученный машинный порядок в 7ой бит 1ого байта, а мантиссу в 2ой, 3ий, 4ый байты.
- Перевод в 16 СС.

Математический порядок числа и машинный порядок смещены относительно друг друга, для того чтобы не хранить знак порядка. Смещение выбрано так что машинный порядок числа всегда имеет положительное значение.

23. Измерение информации. Формула Хартли. Связь с вероятностью. Формула Шеннона.

Один бит несет уменьшение неопределенности нашего знания об исследуемом объекте в два раза.

Формула Хартли для определения количества информации.

$$2^i = N$$

$$i = \log_2(N)$$

i - количество информации в сообщении в битах

N - Количество возможных исходов события

Формула Хартли работает для равновероятных событий.

Вероятность – степень возможности наступления некоторого события. Если события равновероятны, то $P = 1/N \Rightarrow$ чем меньше вероятность наступления данного события, тем больше информации оно несет.

Для достоверного события вероятность его свершения равна 1.

Формула Шеннона для не равновероятных событий.

Если события не равновероятны, то количество информации, получаемое от одного исхода, будет равно: $i_k = \log_2\left(\frac{1}{P_i}\right)$, где P_i вероятность конкретного исхода, преобразованная формула Хартли.

Если исходов от 1 до N , то общее количество исходов равно N , а общее количество получаемой информации равно их сумме.

Задача:

В коробке 50 шаров: 40 красных, 10 зеленых

$$P_{\text{красн}} = 40/50 = 0,8$$

$$P_{\text{зел}} = 10/50 = 0,2$$

Шеннон:

$$K: \log_2(1/0.8) = 0.32 \text{ бита}$$

$$3: \log_2(1/0.2) = 2.32 \text{ бита}$$

24. Алфавитный подход к измерению информации. Кибернетика.

Кибернетика – наука об общих связях между объектами процессов управления. Изучает взаимодействие как живых, так и неживых объектов.

Основатель – Норберт Виннер (1940)

Любой объект кибернетики представляется как «черный ящик», где главными характеристиками являются входная и выходная информации, которая передается как последовательность сигналов.

Информационный объект между кибернетическими системами происходит в символьной форме, т.е. кибернетический подход имеет дело со знаковой информацией.

Пример:

Рассчитать кол-во информации, которая содержится в некоторой книге. Книга 150 стр., 40 строк на странице, 80 символов в строке. Предположим, что мощность алфавита = 256 символов. Каждый символ занимает 1 байт.

Одна страница - 3200 символов.

Всего: $3200 * 150 = 480\,000$

$480\,000 * 8 = 3\,840\,000$ бит

25. Алгоритмы Маркова.

Алгоритм преобразований слов по некоторым правилам.

Слово – произвольная последовательность символов, входящих в алфавит.

Алфавит – непустое конечное множество символов. Также может включать в себя цифры, знаки препинания.

Входное (выходное) слово:

abababbbb

Правила:

Исходный текст

```
*a->a*  
*b->b*  
*=>c  
->*
```

Текст после загрузки в «память»

```
1. *a→a*  
2. *b→b*  
3. *→c  
4. →*
```

Для предотвращения закливания:

1. Нетерминальные « \rightarrow »
2. Терминальные « $\mid\rightarrow$ »

Примечания:

1. Если левая часть некоторого правила входит несколько раз, то на одном шаге заменяется только самое левое вхождение.

2. После каждого применения к слову правила последующее преобразование идет с начала списка правил.

3. Если левая часть правила пуста, то к началу слова добавляется правая часть правила.

Если есть такое правило, то для завершения преобразований обязательно должно присутствовать хотя бы одно терминальное правило. Такое преобразование считается применимым к любому слову, поэтому оно должно всегда быть последним в списке правил, т.к. заблокируются все последующие действия.

4. Правая часть правила тоже может быть пустой. Результатом применения такого правила является удаление из слова последовательности символов, совпадающих с левой частью правила.

Допустимые подстановки и порядок их применения:

1. А - Анатолий

2. ГБ - Гигабайт
3. кг - килограмм
4. т.р. - тысяч рублей
5. тысяч рублей → тысяч рублей (HALT).

Строка: "А купил диск на 500 ГБ весом 3 кг за 15 т.р."

А купил диск на 500 ГБ весом 3 кг за 15 т.р.

Анатолий купил диск на 500 ГБ весом 3 кг за 15 т.р.

Анатолий купил диск на 500 гигабайт весом 3 кг за 15 т.р.

Анатолий купил диск на 500 гигабайт весом 3 килограмм за 15 т.р.

Анатолий купил диск на 500 гигабайт весом 3 килограмм за 15 тысяч рублей.

Анатолий купил диск на 500 гигабайт весом 3 килограмм за 15 тысяч рублей. HALT

26. Машина Тьюринга

Машина Тьюринга – алгоритм (абстрактная машина), где каждый шаг реализует заданную последовательность действий.

1. Определяет вычислимость функций
2. Решает проблему: можно ли создать алгоритм, который определяет, будет ли работать другой алгоритм при новом наборе входных данных

Структура:

- 1) Бесконечная в обе стороны лента
- 2) Голова машины
- 3) Набор состояний (устройство управления)

Голова считывает символ, записанный в ячейке под ним. Считываемый символ и текущее состояние головы однозначно определяет переход в новое состояние.

27. Структура программы на C++

1. Основная часть программы – функция `main ()`.
2. Если используются другие функции, то они прописываются до `main`.
3. Перед `main` указывается тип значения, которое должна вернуть эта функция.
4. Тело `main` размещается в отдельно расположенных скобках.
5. Обязательные компоненты: директивы (`#include`) – подключают необходимые библиотеки заголовочного файла; определение пространства имен через `using namespace` нужны для уменьшения конфликтов совпадения имен функций, переменных, классов

```
#include <iostream> //подключение необходимых библиотек и файлов (директивы процессора)
using namespace std; //указание на используемые пространства имен
//можно делать указание на пространства имен через двоеточие std::cout
//объявление и/или инициализация глобальных переменных (доступны всему, что описано в файле)
int b;
int c = 0;
//определение функций, классов
float a () {
    //какой-то алгоритм
    return 0.0; //возврат успешного завершения работы
}
//определение главной или единственной функции
int main () {
    //какой-то алгоритм
    return 0;
}
```

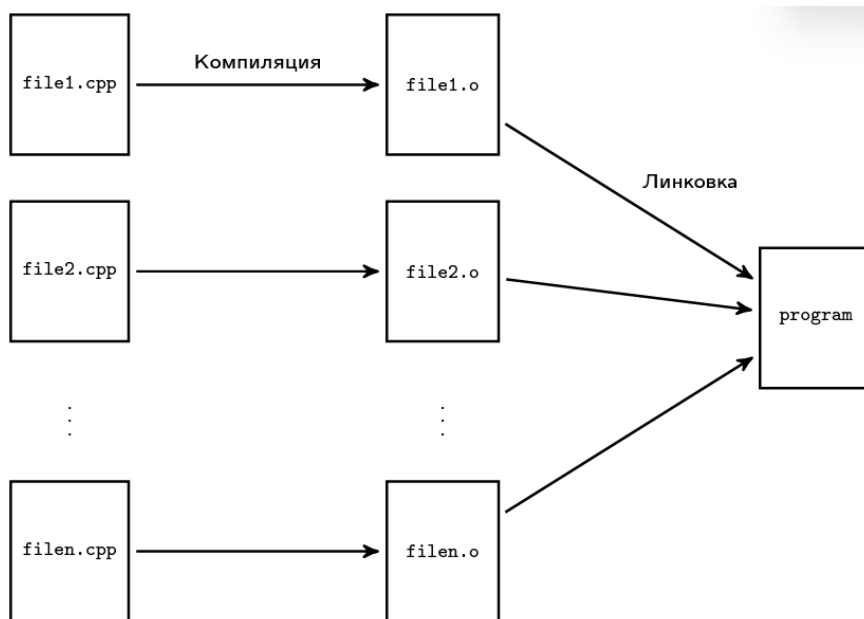
28. Этапы обработки программ.

1. Препроцессорное преобразование текста (работа с кодом как с текстом)

Лексема: `include`, `ifndef`, `ifdef...` - то, что обрабатывается препроцессором перед компиляцией.

2. Компиляция (исходный код → объектный файл)

3. Компоновка (линковка) (объектный файл → исполняемый файл)



29. Типы данных в C++. Спецификаторы типов данных

Любое действие в программе происходит с какими-либо данными, которые в программировании называются операндами.

Каждый из операндов отличается от другого своими свойствами, т.е. имеет свой тип данных.

Данные разных типов:

- 1 по-разному хранятся в памяти компьютера;
- 2 по-разному обрабатываются;
- 3 отличаются разным набором операций и функций, разрешенных для каждого типа;
- 4 отличаются множеством значений, разрешенных для величин каждого типа.

Переменная - это именованное место в памяти, которое может хранить значение определенного типа.

1. Хранит только одно значение
2. Перед использованием любой переменной в программе необходимо зарезервировать для нее область памяти компьютера. Это действие называется описанием переменной.

Основные типы данных:

Целочисленные:

1. int – целый;
2. char – символьный;
3. wchar_t – расширенный символьный;
4. bool – логический;

Типы с плавающей точкой:

5. float – вещественный
6. double – вещественный с двойной точностью;

Составные типы данных:

1. Массивы
2. Функции
3. Структуры
4. Ссылки
5. Классы
6. Объединения
7. Указатели
8. Перечисления

Спецификаторы типа:

1. short – короткий (хочу 16 бит);
2. long – длинный (хочу 32 бита, long long = хочу 64 бита);
3. signed – знаковый: $\{-2^{n-1}; 2^{n-1}-1\}$;
4. unsigned – беззнаковый: $\{0; 2^n-1\}$.

Локальная область действия: определяется внутри блока; область действия от описания данной переменной до конца блока

Глобальная область действия: определена вне любого блока; область действия – файл, в котором эта переменная определена, от точки описания до конца

Классы памяти:

Класс памяти определяет время жизни и область видимости переменной.

Если класс не указан, его определяет компилятор из контекста объявления (резервирования памяти под переменную).

C++ располагает четырьмя спецификаторами класса памяти:

auto

register

static

extern

Auto (автоматическая переменная):

Время жизни: с точки описания до конца блока.

Область видимости: блок.

Не используется для глобальных переменных. Для локальных переменных auto применяется по умолчанию, поэтому задавать его смысла не имеет.

Временное выделение памяти (при выходе из блока память, где находится описание переменной - освобождается)

Register (аналогично auto):

Память выделяется в регистрах самого процессора, а не в оперативной памяти. Если такой возможности у компилятора нет, переменные обрабатываются как auto.

Время жизни: с точки описания до конца блока.

Область видимости: блок

Временное выделение памяти в машинном регистре, убыстряет работу программы

Static:

Статическая переменная.

Инициализация выполняется только один раз. Как константа)

Если задать static вне блока: то она будет глобальной (время жизни: вся программа).

Если задать static в блоке: то она будет локальной (время жизни: с точки описания до конца блока).

Область видимости: и глобальная (видны только в файле, в котором описаны) и локальная (в блоке)

Выделяется постоянная память, значение сохраняется при выходе из блока

Extern:

Время жизни: глобальное. То есть чтобы переменную увидеть в другом файле, можно использовать модификатор — `extern`

Область видимости: глобальная. Вся программа и все программы.

Переменная определяется в любом месте программы, может даже в другом файле.

Используется для создания переменных, доступных во всех файлах программы.

Если переменная при описании инициализируется, то `extern` – игнорируется

30. Switch. Структура. Пример задачи.

Предназначен для разветвления процесса вычислений на несколько направлений.

```
switch (выражение)
{
case значение_1:
    код1;
    break;
case значение_2:
    код2;
    break;
case значение_n:
    кодn;
    break;
default:
    код;
}
```

1. Переключаемое выражение может быть только целочисленным
2. Константные выражения в каждой веточке проверяются на совпадение с переключаемым выражением
3. Если константное выражение совпало с переключаемым, то управление передается этому case
4. Если совпадений нет, то выполняется переход к default, который может быть только один
5. Если default не предусмотрен и выражения не совпали, то в switch не выполняется ни один оператор и алгоритм идет дальше
6. Управление осуществляется с помощью break
7. В case можно объявлять переменную, но нельзя ее инициализировать

31. Виды циклических процессов.

Цикл – группа действий, повторяющихся неоднократно.

Тело цикла – повторяющееся действие.

Шаг цикла – однократное выполнение тела.

Параметр цикла – управление циклическим процессом.

Виды циклов:

Арифметический: известно, сколько раз необходимо выполнить тело цикла; переменная цикла - счетчик;

Итерационный: количество раз работы цикла неизвестно; одна причина окончания цикла – достижение результата

Поисковый: количество раз работы цикла неизвестно; есть более одной причины окончания цикла

32. Методы решения нелинейных уравнений.

1. Метод половинного деления (дихотомия) – метод бисекции

Известен интервал

$$f(a) * f(b) < 0$$

1.1 Интервал делится пополам: $c = (a + b)/2$

если условие не выполняется, значит корня там нет. Часть интервала отбрасывается.

1.2 Итерации продолжают, пока расстояние между a и b не будет eps.

```
float a,b,c;
float fA, fB, fC;
double eps = 0.000001;

a = 1;
b = 2;
while ((b-a) > eps) {
    c = (a+b)/2;
    fA = (pow(a, 3) - a - 1);
    fB = (pow(b, 3) - b - 1);
    fC = (pow(c, 3) - c - 1);
    if (fA * fC < 0) {
        b = c;
    }
    else if (fC * fB < 0) {
        a = c;
    }
    else {
        cout << "Что-то не так" << endl;
    }
}
cout << "Корни: " << a << "; " << b << endl;
return 0;
}
```

Необходимо решить уравнение $x^3 - 0,2x^2 + 0,5x + 1,5 = 0$ с точностью до $\epsilon < 0,001$ на отрезке $[-1,0]$.

1. Обозначим начальную и конечную точки отрезка символами a и b соответственно.

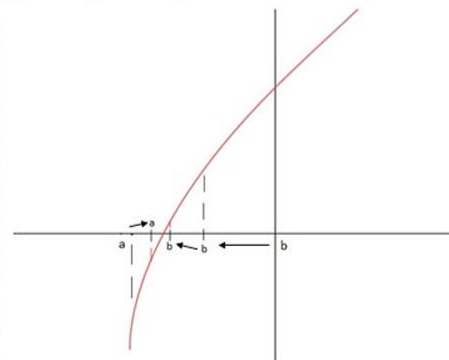
В общем виде уравнение имеет вид: $F(x) = x^3 - 0,2x^2 + 0,5x + 1,5$

3. Разделим отрезок на 2 части: $(a-b)/2 = (-1+0)/2 = -0,5$.

4. Если произведение $F(a)*F(x)>0$, то начала отрезка a переносится в x ($a=x$), иначе, конец отрезка b переносится в точку x ($b=x$).

Полученный отрезок делим опять пополам и т.д. Весь произведенный расчет отражен ниже в таблице. Корнем является любая граница после окончания итераций.

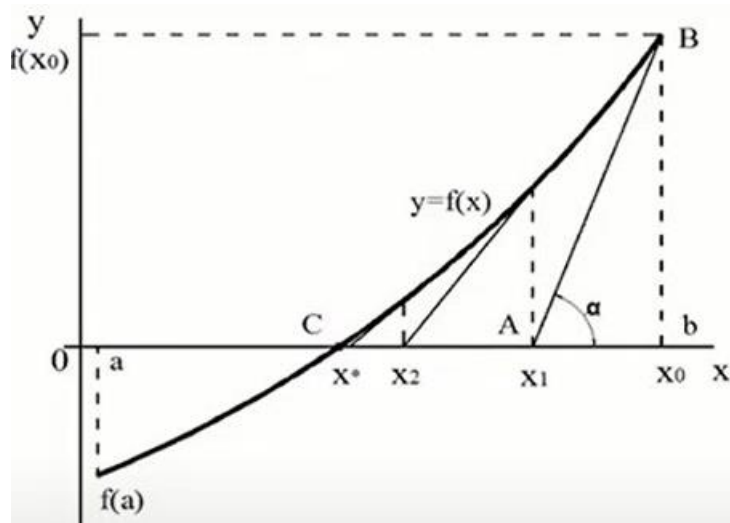
a	b	x	F(a)	F(x)	F(a)*F(x)
-1.0000	0.0000	-0.5000	-0.2000	1.0750	-0.2150
-1.0000	-0.5000	-0.7500	-0.2000	0.5906	-0.1181250
-1.0000	-0.7500	-0.87500	-0.2000	0.2395	-0.0478906
-1.0000	-0.8750	-0.93750	-0.2000	-0.315	-0.0062988
-1.0000	-0.9375	-0.96875	-0.2000	-0.0812	0.0162439
-0.9688	-0.9375	-0.95313	-0.0812	-0.0241	0.0019587
0.9531	0.93750	-0.94531	-0.0241	0.0039	-0.0000934
-0.9531	-0.945313	-0.94922	-0.0241	-0.0101	0.0002429
-0.9492	-0.945313	-0.94727	-0.0101	-0.0031	0.0000311
-0.9473	-0.945313	-0.94629	-0.0031	0.0004	-0.0000012
-0.9473	-0.946289	-0.94678	-0.0031	-0.0013	0.0000042



2. Метод Ньютона (метод касательной)

Построение касательных к графику.

Сначала на одном из концов интервала, в точке пересечения с осью X строится новая касательная. Продолжается до тех пор, пока значение не будет приближено к ϵ_{rs} .



1. Найти 1 и 2 производные

2.

Дано: $f(x) = x^3 - x - 1$

$f'(x) = 3x^2 - 1$;

$f''(x) = 6x$;

$[a;b] = [1;2]$

проверяем, берем сначала а или б за начальное значение x

$f(b) * f''(b) > 0$

$(8-2-1)*(6*2) > 0 \rightarrow x_0 = b$ (иначе a)

```
#include <iostream>
#include <cmath>
using namespace std;
int main () {
    float x; xpred; // xpred - икс предыдущий
    x = 2; // x0 = 2
    xpred = 0; //инициализация переменной x(n-1)
    double eps = 0.000001;

    while (abs(x-xpred) > eps) {
        xpred = x;
        //сюда пишем формулу, где x(i-1) - f(x(i-1)) / f'(x(i-1))
        x = xpred - (pow(xpred, 3) - xpred - 1) / (3*pow(xpred,2)-1) //xпред - f(x)-f'(x);
    }

    cout << "Корень: " << x << endl;
    return 0;
}
```

1. Геометрический смысл первой производной – тангенс угла наклона касательной к Ох, то есть $f'(x_0) = \operatorname{tg} \alpha = k$
2. Запишем уравнение прямой с угловым коэффициентом $y = k \cdot x + b$.
3. Найдем уравнение касательной в точке x_0 : $f(x_0) = f'(x_0) \cdot x_0 + b$
4. Выразим b: $b = f(x_0) - f'(x_0) \cdot x_0$
5. Теперь запишем уравнение касательной в новом виде, подставив выражение b, полученное в 4 пункте:
 $y = f'(x_0) \cdot x + f(x_0) - f'(x_0) \cdot x_0$
6. Преобразуем уравнение, вынеся общий множитель $f'(x_0)$: $y = f'(x_0) \cdot (x - x_0) + f(x_0)$
7. Так как нам нужно найти точку пересечения уравнения из 6 пункта с Ох, то приравняем его к нулю:
 $f'(x_0) \cdot (x - x_0) + f(x_0) = 0$
8. Выражаем x – пересечение уравнения касательной с Ох:

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

○ Метод итераций

Дано: $f(x) = x^3 - x - 1$

$f'(x) = 3x^2 - 1$;

$f''(x) = 6x$;

$[a; b] = [1; 2]$

$\phi(x) = x + \lambda(x^3 - x - 1)$

$-1/r < \lambda < 0$ (если $f'(x) > 0$)

$0 < \lambda < 1/r$ (если $f'(x) < 0$)

$r = \max(|f'(a)|, |f'(b)|)$

$r = \max(|3 \cdot 1^2 - 1|, |3 \cdot 2^2 - 1|) = \max(2, 11) \rightarrow r = 11$

$f'(x) > 0 \rightarrow -1/r < \lambda < 0$

Берем любое удобное число в диапазоне

$\lambda = -1/20 = -0.05$

```
#include <iostream>
#include <cmath>
using namespace std;
int main () {
    float x; xpred; // xpred - икс предыдущий
    l = -0.05
    x = 2; //любой икс из [1;2]
    xpred = 0; //заведомо меньше чем x
    double eps = 0.000001;

    while (abs(x-xpred) > eps) {
        xpred = x;
        x = 1 * (pow(xpred, 3) - xpred - 1) + xpred;
    }
    cout << "Корень:" << x;
    return 0;
}
```


33. Операторы циклических процессов.

1. С предусловием (while) – ромб

while (условие) {

тело цикла

}

2. С постусловием (do while) – ромб

do (счетчик; условие завершения цикла; правило изменения счетчика шагов)

while (условие)

тело цикла

3. С автоматическим изменением параметра (for) – шестиугольник

for (счетчик; условие завершения цикла; правило изменения счетчика шагов) {

тело цикла

}