Информатика

Программа как последовательность действий компьютера – содержит набор исполняемых команд, которые понимает процессор (исполнительное устройство). Машина фон-неймана.

Условный оператор.

```
if(b<2):
        a=a+1
    else:
        b =b+1
```

Циклы.

```
for i in range(5):
        g=g+i
```

Присвоение.

```
v=1
```

Процедуры. Функции.

```
def func(x):
        return  x**2
```

Команды ввода-вывода данных.

Класс

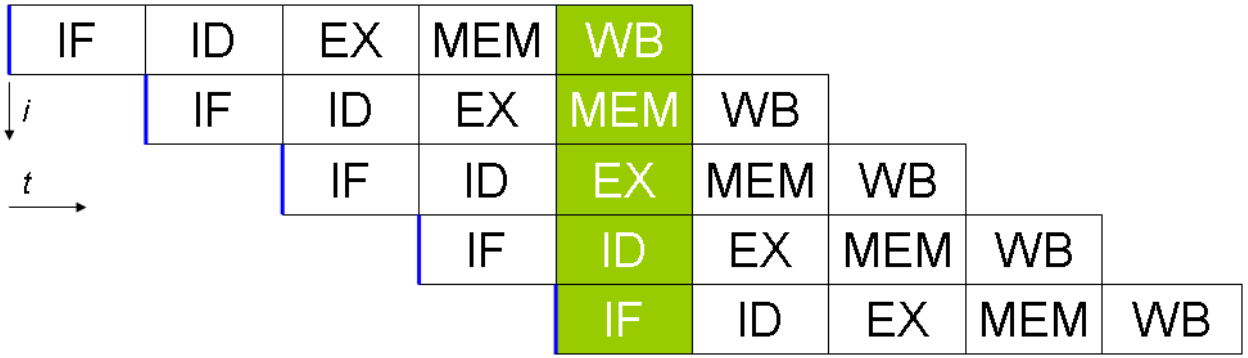
```
class parent():
        def __init__(self,name):
                self.name=name+" parent init"
        def print(self):
                print("Parent ",self.name)

class child(parent):
        def __init__(self,name):
                super().__init__(name)
        def print(self):
                parent.print(self)
                print("Child ",self.name)
        def __hello(self):
                print(“Скрытый метод”)

d = child("Child object")
d.print()
```

Конвейерное исполнение

- способ организации вычислений, используемый в современных процессорах и контроллерах с целью повышения их производительности (увеличения числа инструкций, выполняемых в единицу времени — эксплуатация параллелизма на уровне инструкций), технология, используемая при разработке компьютеров и других цифровых электронных устройств.
- IF (англ. Instruction Fetch) — получение инструкции,
- ID (англ. Instruction Decode) — раскодирование инструкции,
- EX (англ. Execute) — выполнение,
- MEM (англ. Memory access) — доступ к памяти,
- WB (англ. Register write back) — запись в регистр.



Нековейерное время исполнения пропорционально 5*5 = 25
Конвейерное 9

- Конфликты по управлению возникают при конвейерном выполнении условных передач управления и других команд, которые изменяют значение программного счетчика. Существует много способов обработки остановов конвейера, вызванных задержкой передачи управления, но для глубоких конвейеров в основном используются агрессивные средства, такие как предсказания передач управления.
- Предсказание ветвлений позволяет сократить время простоя конвейера за счёт предварительной загрузки и исполнения инструкций, которые должны выполниться после выполнения инструкции условного перехода. Прогнозирование ветвлений играет критическую роль, так как в большинстве случаев (точность предсказания переходов в современных процессорах превышает 90 %) позволяет оптимально использовать вычислительные ресурсы процессора
- Статические методы предсказания ветвлений являются наиболее простыми. Суть этих методов состоит в том, что различные типы переходов: либо выполняются всегда; либо не выполняются никогда.
- Динамические методы, широко используемые в современных процессорах, подразумевают анализ истории ветвлений.

Понятие о машинном языке и языке Ассемблер.

- Язы́к ассе́мблера (англ. assembly language) — машинно-ориентированный язык программирования низкого уровня. Его команды прямо соответствуют отдельным командам машины или их последовательностям, также он может предоставлять дополнительные возможности облегчения программирования, такие как макрокоманды, выражения, средства обеспечения модульности программ.
- Команды языка ассемблера один к одному соответствуют командам процессора. Фактически, они и представляют собой более удобную для человека символьную форму записи — мнемокоды — команд и их аргументов. При этом одной команде языка ассемблера может соответствовать несколько вариантов команд процессора

- Примеры команд для intel 8086
- Команды пересылки данных (mov и др.)
- mov ax,bx – пересылка из 16-и разрядного регистра bx в ax
- mov ax,ds:[si] – пересылка из памяти по адресу ds:[si] в регистр ax
- ds – сегмент данных si – регистр смещения в сегменте.
- Арифметические команды (add, sub, imul и др.)
- Логические и побитовые операции (or, and, xor, shr и др.)
- Команды управления ходом выполнения программы (jmp, loop, ret, call и др.)
- Ret – возврат из прерывания, или процедуры путем извлечения адреса перехода из стека.
- Команды вызова прерываний (иногда относят к командам управления): int
  - mov ah, 9 ; Номер функции DOS - в AH
- mov dx, offset message ; Адрес строки - в DX
- int 21h ; Вызов системной функции DOS
- ret ; Завершение COM-программы
  - message db "Hello World!", 0Dh, 0Ah, '$' ; Строка для выводаю, например:
- Команды ввода-вывода в порты (in, out)
- Для микроконтроллеров и микрокомпьютеров характерны также команды, выполняющие проверку и переход по условию (jne,jqe,jge), например:
- cmp ax,bx
- Jne metka – переход если не равно в какую то область памяти где расположена метка
- Работа со стеком – push, pop, регистры стека ss - сегмент, sp, bp - смещение
- push ax – занесли в стек регистр ax
- pop bx – извлекли из стека в bx

Архитектура процессоров

- Архитекту́ра проце́ссора — количественная составляющая компонентов микроархитектуры вычислительной машины (процессора компьютера) (например, регистр флагов или регистры процессора).
- С точки зрения:
- программиста — совместимость с определённым набором команд (например, процессоры, совместимые с командами Intel x86), их структуры (например, систем адресации или организации регистровой памяти) и способа исполнения (например, счётчик команд).
- аппаратной составляющей вычислительной системы — это некий набор свойств и качеств, присущий целому семейству процессоров (иначе говоря — «внутренняя конструкция», «организация» этих процессоров).
- Имеются различные классификации архитектур процессоров как по организации (например, по количеству и сложности отдельных команд: RISC, CISC, MISC, VLIW; по возможности доступа команд к памяти), так и по назначению (например, специализированные графические, математические или предназначенные для цифровой обработки сигналов).

CISC

- CISC (англ. Complex Instruction Set Computer — «компьютер с полным набором команд») — тип процессорной архитектуры, в первую очередь, с нефиксированной длиной команд, а также с кодированием арифметических действий в одной команде и небольшим числом регистров, многие из которых выполняют строго определенную функцию.
-
- Самый яркий пример CISC архитектуры — это x86 (он же IA-32) и x86_64 (он же AMD64).
-
- В CISC процессорах одна команда может быть заменена ей аналогичной, либо группой команд, выполняющих ту же функцию. Отсюда вытекают плюсы и минусы архитектуры: высокая производительность благодаря тому, что несколько команд могут быть заменены одной аналогичной, но большая цена по сравнению с RISC процессорами из-за более сложной архитектуры, в которой многие команды сложнее раскодировать.

RISC

-
- RISC (англ. Reduced Instruction Set Computer — «компьютер с сокращённым набором команд») — архитектура процессора, в котором быстродействие увеличивается за счёт упрощения инструкций: их декодирование становится более простым, а время выполнения — меньшим. Первые RISC-процессоры не имели даже инструкций умножения и деления и не поддерживали работу с числами с плавающей запятой.
- По сравнению с CISC эта архитектура имеет константную длину команды, а также меньшее количество схожих инструкций, позволяя уменьшить итоговую цену процессора и энергопотребление, что критично для мобильного сегмента. У RISC также большее количество регистров.
- Примеры RISC-архитектур: PowerPC, серия архитектур ARM (ARM7, ARM9, ARM11, Cortex).
- В общем случае RISC быстрее CISC. Даже если системе RISC приходится выполнять 4 или 5 команд вместо одной, которую выполняет CISC, RISC все равно выигрывает в скорости, так как RISC-команды выполняются в 10 раз быстрее.
- Отсюда возникает закономерный вопрос: почему многие всё ещё используют CISC, когда есть RISC? Всё дело в совместимости. x86_64 всё ещё лидер в desktop-сегменте только по историческим причинам. Так как старые программы работают только на x86, то и новые desktop-системы должны быть x86(_64), чтобы все старые программы и игры могли работать на новой машине.
- Для Open Source это по большей части не является проблемой, так как пользователь может найти в интернете версию программы под другую архитектуру. Сделать же версию проприетарной программы под другую архитектуру может только владелец исходного кода программы.

MISC

- MISC (англ. Minimal Instruction Set Computer — «компьютер с минимальным набором команд»).
- Ещё более простая архитектура, используемая в первую очередь для ещё большего уменьшения итоговой цены и энергопотребления процессора. Используется в IoT-сегменте и недорогих компьютерах, например, роутерах.
- Для увеличения производительности во всех вышеперечисленных архитектурах может использоваться “спекулятивное исполнение команд”. Это выполнение команды до того, как станет известно, понадобится эта команда или нет.

- MISC (Multipurpose Instruction Set Computer). Элементная база состоит из двух частей, которые либо выполнены в отдельных корпусах, либо объединены. Основная часть – RISC CPU, расширяемый подключением второй части – ПЗУ микропрограммного управления. Система приобретает свойства CISC. Основные команды работают на RISC CPU, а команды расширения преобразуются в адрес микропрограммы. RISC CPU выполняет все команды за один такт, а вторая часть эквивалентна CPU со сложным набором команд. Наличие ПЗУ устраняет недостаток RISC, выраженный в том, что при компиляции с языка высокого уровня микрокод генерируется из библиотеки стандартных функций, занимающей много места в ОЗУ. Поскольку микропрограмма уже дешифрована и открыта для программиста, то времени выборки из ОЗУ на дешифрацию не требуется.

VLIW

- VLIW (англ. Very Long Instruction Word — «очень длинная машинная команда») — архитектура процессоров с несколькими вычислительными устройствами. Характеризуется тем, что одна инструкция процессора содержит несколько операций, которые должны выполняться параллельно.
- По сути является архитектурой CISC со своим аналогом спекулятивного исполнения команд, только сама спекуляция выполняется во время компиляции, а не во время работы программы, из-за чего уязвимости Meltdown и Spectre невозможны для этих процессоров. Компиляторы для процессоров этой архитектуры сильно привязаны к конкретным процессорам. Например, в следующем поколении максимальная длина «очень длинной команды» может из условных 256 бит стать 512 бит, и тут приходится выбирать между увеличением производительности путём компиляции под новый процессор и обратной совместимостью со старым процессором. Опять же, Open Source позволяет простой перекомпиляцией получить программу под конкретный процессор.
-
- Примеры архитектуры: Intel Itanium, Эльбрус-3.

Виртуальные архитектуры

- Например, виртуальная JVM-архитектура эмулируется на целевой реальной машине. Поэтому достаточно JVM-машины для запуска на ней любой Java-программы. Другим примером виртуальной архитектуры является .NET CIL (CLR).
-
- Из минусов виртуальных архитектур можно выделить меньшую производительность по сравнению с реальными архитектурами. Этот минус нивелируется с помощью JIT- и AOT-компиляции. Однако большим плюсом будет кроссплатформенность.
-
- Дальнейшим развитием этих архитектур стали гибридные архитектуры. Например современные x86_64 процессоры хотя и CISC-совместимы, но являются процессорами с RISC-ядром. В таких гибридных CISC-процессорах CISC-инструкции преобразовываются в набор внутренних RISC-команд.

Алгоритм и его свойства (Определённость, результативность, массовость, дискретность).

- Определенность - алгоритм выполняет четко определенные действия, и указания алгоритма одинаково интерпретируются исполняющим устройством. (Невозможно, например сложив 2+2 получить один раз 5, а другой раз 4).
- Результативность – алгоритм приводит к конечному результату за конечное число шагов. (Например, зацикливание к результату не приводит).
- Массовость - алгоритм способен решать некоторый класс задач различающихся исходными данными. Например, алгоритм сортировки может сортировать различные массивы, а не только конкретный из чисел 1, 20, 10. Кроме того, алгоритмы сортировки можно применять и для сортировки различных объектов для которых допустима операция ранжирования (например, для строк).
- Дискретность – шаги алгоритма выполняются строго последовательно, друг за другом, каждая следующая команда только после выполнения предыдущей.

Понятие трансляции, компиляции, интерпретации, jit-compilation (компиляция во время исполнения).

- Трансляция – перевод с одного языка на другой, обычно с высокоуровневого языка в машинный.
- Компиляция – перевод с высокоуровневого языка в готовый исполнимый модуль. (например, .COM, a.out, COFF, DOS MZ Executable (exe), Windows PE, Windows NE, ELF (Executable and Linkable format, Linux)). Обычно этап компиляции разбивается на этап непосредственной трансляции в объектные файлы или компиляцию с относительной адресацией данных и объектов в памяти, а затем объектные файлы проходят этап сборки (линковки, link), когда файлы собираются в общий модуль уже с абсолютной адресацией. (c, c++, fortran)
- Интерпретация – выполнение команд последовательно, каждая команда на языке высокого уровня непосредственно транслируется в машинный код и выполняется, затем другая команда. (cpython, php и т.д.)
- Jit (just in time) – компиляция используемая виртуальными машинами, вроде CLR .Net, JVM, LLVM, parrot (perl и др.), руру принцип заключается в том, что сначала код с высокоуровневого языка переводится в байт код – приближенный к машинному универсальному коду, затем исполняется виртуальной машиной путем компиляции во время исполнения в машинный язык (язык) целевой платформы и исполняется (повторная компиляция уже откомпилированных кусков кода не проводится). Примеры: Java – JVM, C# - CLR . Net (виртуальная машина), CIL (байт код) , LLVM (Low level VM)– julia, kotlin и др.

Классы языков программирования высокого уровня: алгоритмические, логические, функциональные, объектно-ориентированные. Императивные, декларативные.

- Императивные языки – используют конструкции позволяющие написать алгоритм в котором указывается как достичь цели. Например, язык C, fortran, python.
- Декларативные языки – используют конструкции указывающие саму цель или конечный результат, который должна разрешить исполняющая машина. К таким языкам часто относят логические и функциональные языки (Prolog, Mercury, Lisp, Clojure, Haskell).

Процедурные языки программирования. Модульные программы.

- Процеду́рное программи́рование — программирование на императивном языке, при котором последовательно выполняемые операторы можно собрать в подпрограммы, то есть более крупные целостные единицы кода, с помощью механизмов самого языка. (Использование процедур и функций).
- int sqr(int x)
- {
- return x*x;
- }
- Модуль – отдельный файл с набором функций решающих общие задачи. Например, модуль math во многих языках программирования для выполнения операций над числами с плавающей запятой – ln, exp, sin, cos и т.д. Обычно представляют собой уже готовые и откомпилированные библиотеки. Реализуется условная инкапсуляция, возможность поддержки и использования модулей другими разработчиками в своих программах. Возможность использования статических и динамически подключаемых библиотек.

Структу́рное программи́рование

- Парадигма программирования, в основе которой лежит представление программы в виде иерархической структуры блоков. Концептуализирована в конце 1960-х — начале 1970-х годов на фундаменте теоремы Бёма — Якопини, математически обосновывающей возможность структурной организации программ, и работы Эдсгера Дейкстры «О вреде оператора goto» (англ. Goto considered harmful).

- В соответствии с парадигмой, любая программа строится без использования оператора goto из трёх базовых управляющих структур: последовательность, ветвление, цикл; кроме того, используются подпрограммы. При этом разработка программы ведётся пошагово, методом «сверху вниз».

- Методология структурного программирования появилась как следствие возрастания сложности решаемых на компьютерах задач, и соответственно, усложнения программного обеспечения. В 1970-е годы объёмы и сложность программ достигли такого уровня, что традиционная (неструктурированная) разработка программ перестала удовлетворять потребностям практики. Программы становились слишком сложными, чтобы их можно было нормально сопровождать. Поэтому потребовалась систематизация процесса разработки и структуры программ.

Объектно-ориентированное программирование. Понятие класса, объекта. Наследование, инкапсуляция, полиморфизм. Виртуальные методы.

- Класс представляет собой абстракцию описывающую различные экземпляры с одними характеристиками и поведением в предметной области – объектами. Объект реализация – класса, отдельный элемент хранимый в памяти содержащий свойства (атрибуты) и методы (процедуры и функции). Отличие от записи – наличие методов.
- Инкапсуляция – объединение в одном программном элементе множества свойств и методов для работы с ними, переменная связанная с объектом ссылается на все эти методы и свойства, но некоторые из них при этом можно скрыть от вызова из внешней программы.
- Объект Cursor.
- Cursor.string = “1212”
- Cursor.gotonextParagraph()

- Наследование – перенос свойств и методов от класса родителя к классу наследнику. Например, класс студент наследует от класса гражданин РФ, фамилию имя отчество, номер паспорта и метод – проживать, получать_прописку. Добавляет метод учиться в высшем учебном заведении (вузе).
- Полиморфизм – возможность использовать одноименные методы и операции для объектов различных классов, типов. То есть действие выполняется по разному в зависимости от объекта к которому применяется.
- Например, операция суммирования полиморфна, если применяется к строкам или числам.
- Либо, например функция (метод) учиться, есть и у школьника и у студента. Но выполняется по разному.

Типы данных. Записи, файлы, динамически структуры данных: очереди, стеки, деревья, деки.

- Запись – составной элемент данных состоящий из полей. Поле – именованный элемент принимающий значение на каком-то типе или домене. Например struct circle { int radius; int x0, int y0 }, struct student {string FIO; int year}. Поле FIO – строка, поле год поступления – целый. Домен уточняет возможное значение, обычно с помощью предиката проверки какого-либо условия.
- Файл – именованная область данных на диске, доступ к элементам, которой реализуется последовательно.
- Очередь – элемент хранения однотипных данных, характеризующаяся способом доступа к отдельным хранимым ее элементам – первый помещаемый элемент в очередь извлекается первым.
- Стек – первый помещенный в стек элемент извлекается последним, последний извлекается первым. Push ax, pop bx.
- Дек – двунаправленный стек.
- Граф – совокупность вершин связанных ребрами (ребру ставится в соответствии вес).
- Дерево – граф без циклов.
- Бинарной дерево – дерево, где из вершин идет всего две ветви.

Рекурсивные алгоритмы, рекурсивные процедуры и функции.

- Рекурсивная процедура - процедура или функция вызывающая сама себя. Промежуточные результаты формируемые функцией обычно сохраняются в стеке. Иногда для функций используют хвостовую рекурсию, дабы не нагружать стек.
- Классический пример факториала:
- Нерекурсивный
- V=1
- For i in range(1,n):
  - V = v*I

  Рекурсивный

  ```
  def  f(n):
    if(n<1): return 1
    return n*f(n-1)
  ```

Жизненный цикл программного обеспечения.

Каскадная модель ЖЦ

- Анализ требований,
- Проектирование,
- Кодирование (программирование),
- Тестирование и отладка,
- Эксплуатация и сопровождение.

Инкрементая модель

Спиральная модель

Проблема верификации и сертификации программ. Тестирование (черного и белого ящика, альфа, бета).

- Тестирование черного ящика – внутреннее устройство не рассматривается, рассматривается что на входе, и что должна получить программа при данном входе. Если что-то не соответствует, значит на каких-то тестах программа провалилась.
- Белый ящик – тестирование проводится с учетом структуры программы, ее реализации, тестируются отдельные части, программа подвергается отладке. Прохождению через этапы которые известны благодаря знанию о структуре программы.

- Альфа-тестирование — имитация реальной работы с системой штатными разработчиками, либо реальная работа с системой потенциальными пользователями/заказчиком. Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но в некоторых случаях может применяться для законченного продукта в качестве внутреннего приёмочного тестирования. Иногда альфа-тестирование выполняется под отладчиком или с использованием окружения, которое помогает быстро выявлять найденные ошибки. Обнаруженные ошибки могут быть переданы тестировщикам для дополнительного исследования в окружении, подобном тому, в котором будет использоваться программа.
- Бета-тестирование — в некоторых случаях выполняется распространение предварительной версии (в случае проприетарного программного обеспечения иногда с ограничениями по функциональности или времени работы) для некоторой большей группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок. Иногда бета-тестирование выполняется для того, чтобы получить обратную связь о продукте от его будущих пользователей.

Стратегии разработки и отладки (проектирование снизу вверх, сверху вниз).

- Проектирование снизу вверх – создаются сначала элементарные функции, из которых синтезируются более крупные блоки программы.
- Сверху вниз – задача декомпозируется на более мелкие, те в свою очередь тоже декомпозируются на более простые функции.

Гибкие методологии разработки (Agile). Итеративная. Экстремальное программирование. Scrum.

- Серия подходов к разработке программного обеспечения, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля.
- Применяется как эффективная практика организации труда небольших групп (которые делают однородную творческую работу) в объединении с управлением ими комбинированным (либеральным и демократическим) методом.
- Большинство гибких методологий нацелены на минимизацию рисков путём сведения разработки к серии коротких циклов, называемых итерациями, которые обычно длятся две-три недели. Каждая итерация сама по себе выглядит как программный проект в миниатюре и включает все задачи, необходимые для выдачи мини-прироста по функциональности: планирование, анализ требований, проектирование, программирование, тестирование и документирование. Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что гибкий программный проект готов к выпуску в конце каждой итерации. По окончании каждой итерации команда выполняет переоценку приоритетов разработки.
- Agile-методы делают упор на непосредственное общение лицом к лицу. Большинство agile-команд расположены в одном офисе, иногда называемом англ. bullpen. Как минимум, она включает и «заказчиков» (англ. product owner — заказчик или его полномочный представитель, определяющий требования к продукту; эту роль может выполнять менеджер проекта, бизнес-аналитик или клиент). Офис может также включать тестировщиков, дизайнеров интерфейса, технических писателей и менеджеров.
- Основной метрикой agile-методов является рабочий продукт. Отдавая предпочтение непосредственному общению, agile-методы уменьшают объём письменной документации по сравнению с другими методами. Это привело к критике этих методов как недисциплинированных.

Экстремальное программирование

- Двенадцать основных приёмов экстремального программирования (по первому изданию книги Extreme programming explained) могут быть объединены в четыре группы:
-
- Короткий цикл обратной связи (Fine-scale feedback)
- Разработка через тестирование (Test-driven development)
- Игра в планирование (Planning game)
- Заказчик всегда рядом (Whole team, Onsite customer)
- Парное программирование (Pair programming)
- Непрерывный, а не пакетный процесс
- Непрерывная интеграция (Continuous integration)
- Рефакторинг (Design improvement, Refactoring)
- Частые небольшие релизы (Small releases)
- Понимание, разделяемое всеми
- Простота проектирования (Simple design)
- Метафора системы
- Коллективное владение кодом (Collective code ownership) или выбранными шаблонами проектирования (Collective patterns ownership)
- Стандарт оформления кода (Coding standard or Coding conventions)
- Социальная защищённость программиста (Programmer welfare):
- 40-часовая рабочая неделя (Sustainable pace, Forty-hour week)
-
- Тестирование – функциональное тестирование и Unit –тестирование модулей (отдельных модулей или кусков кода), предполагается, что тесты создаются даже раньше чем сам код.

SCRUM

- SCRUM используется как в сфере разработки ПО, так и в других производственных бизнес-отраслях.
- Кроме управления проектами по разработке ПО, SCRUM может также использоваться в работе команд поддержки программного обеспечения, или как подход к управлению разработкой и сопровождению программ, и даже в ремонте.
- В методологии Scrum всего три роли:
- Scrum Master
- Product Owner
- Team
- Скрам Мастер (Scrum Master) – самая важная роль в методологии. Скрам Мастер отвечает за успех Scrum в проекте. По сути, Скрам Мастер является интерфейсом между менеджментом и командой. Как правило, эту роль в проекте играет менеджер проекта или тимлид. Важно подчеркнуть, что Скрам Мастер не раздает задачи членам команды. В Agile команда является самоорганизующейся и самоуправлямой. Основные обязанности Скрам Мастера:
- Создает атмосферу доверия
- Участвует в митингах в качестве фасилитатора
- Устраняет препятствия
- Делает проблемы и открытые вопросы видимыми
- Отвечает за соблюдение практик и процесса в команде
- Скрам Мастер ведет Daily Scrum Meeting и отслеживает прогресс команды при помощи Sprint Backlog, отмечая статус всех задач в спринте. ScrumMaster может также помогать Product Owner создавать Backlog для команды.

Product Owner

- Product Owner – это человек, отвечающий за разработку продукта. Как правило, это product manager для продуктовой разработки, менеджер проекта для внутренней разработки и представитель заказчика для заказной разработки. Product Owner – это единая точка принятия окончательных решений для команды в проекте, именно поэтому это всегда один человек, а не группа или комитет. Обязанности Product Owner :
-
- Отвечает за формирование product vision
- Управляет ROI (возврат инвестиций)
- Управляет ожиданиями заказчиков и всех заинтересованных лиц
- Координирует и приоритизирует Product backlog (список требований к продукту)
- Предоставляет понятные и тестируемые требования команде
- Взаимодействует с командой и заказчиком
- Отвечает за приемку кода в конце каждой итерации
-
- Product Owner ставит задачи команде, но он не вправе ставить задачи конкретному члену проектной команды в течении спринта.

- Команда (Team).В методологии Scrum команда является самоорганизующейся и самоуправляемой. Команда берет на себя обязательства по выполнению объема работ на спринт перед Product Owner. Работа команды оценивается как работа единой группы. В Scrum вклад отдельных членов проектной команды не оценивается, так как это разваливает самоорганизацию команды. Обязанности команды таковы:
-
- Отвечает за оценку элементов баклога
- Принимает решение по дизайну и имплементации
- Разрабатывает софт и предоставляет его заказчику
- Отслеживает собственный прогресс (вместе со Скрам Мастером)
- Отвечает за результат перед Product Owner
-
- Размер команды ограничивается размером группы людей, способных эффективно взаимодействовать лицом к лицу. Типичные размер команды – 7 плюс минус 2.
-
- Команда в Scrum кроссфункциональна. В нее входят люди с различными навыками – разработчики, аналитики, тестировщики. Нет заранее определенных и поделенных ролей в команде, ограничивающих область действий членов команды. Команда состоит из инженеров, которые вносят свой вклад в общий успех проекта в соответствии со своими способностями и проектной необходимостью.
-
- Команда самоорганизуется для выполнения конкретных задач в проекте, что позволяет ей гибко реагировать на любые возможные задачи. Для облегчения коммуникаций команда должна находиться в одном месте (colocated). Предпочтительно размещать команду не в кубиках, а в одной общей комнате для того, чтобы уменьшить препятствия для свободного общения. Команде необходимо предоставить все необходимое для комфортной работы, обеспечить досками и флипчартами, предоставить все необходимые инструменты и среду для работы.

- Product Backlog
- Product Backlog – это приоритезированный список имеющихся на данный момент бизнес требований и технических требований к системе. Product Backlog включает в себя use cases, defects, enhancements, technologies, stories, features, issues, и т.д.. Product backlog также включает задачи, важные для команды, например «провести тренинг», «добить всем памяти»
- Product Backlog
- Product Backlog постоянно пересматривается и дополняется – в него включаются новые требования, удаляются ненужные, пересматриваются приоритеты. За Product Backlog отвечает Product Owner. Он также работает совместно с командой для того, чтобы получить приближенную оценку на выполнение элементов Product Backlog для того, чтобы более точно расставлять приоритеты в соответствии с необходимым временем на выполнение.
- Sprint Backlog
- Sprint Backlog содержит функциональность, выбранную Product Owner из Product Backlog. Все функции разбиты по задачам, каждая из которых оценивается командой. Каждый день команда оценивает объем работы, который нужно проделать для завершения задач.
-
- Пример Sprint Backlog
- Сумма оценок оставшейся работы может быть построена как график зависимости от времени. Такой график называется Sprint Burndown chart. Он демонстрирует прогресс команды по ходу спринта.
- Sprint Burndown chart
- Спринт (Sprint)
- В Scrum итерация называется Sprint. Ее длительность составляет 1 месяц (30 дней). Результатом Sprint является готовый продукт (build), который можно передавать (deliver) заказчику (по крайней мере, система должна быть готова к показу заказчику). Короткие спринты обеспечивают быстрый feedback проектной команде от заказчика. Заказчик получает возможность гибко управлять scope системы, оценивая результат спринта и предлагая улучшения к созданной функциональности.
- Такие улучшения попадают в Product Backlog, приоритезируются наравне с прочими требованиями и могут быть запланированы на следующий (или на один из следующих) спринтов. Каждый спринт представляет собой маленький «водопад». В течение спринта делаются все работы по сбору требований, дизайну, кодированию и тестированию продукта. Scope спринта должен быть фиксированным. Это позволяет команде давать обязательства на тот объем работ, который должен быть сделан в спринте. Это означает, что Sprint Backlog не может быть изменен никем, кроме команды.

- Жизненный цикл спринта
- Планирование спринта
-
- В начале каждого спринта проводится планирование спринта. В планировании спринта участвуют заказчики, пользователи, менеджмент, Product Owner, Скрам Мастер и команда. Планирование спринта состоит из двух последовательных митингов.
-
- Планирование спринта, митинг первый. Участники: команда, Product Onwer, Sxrum Master, пользователи, менеджемент Цель: Определить цель спринта (Sprint Goal) и Sprint Backlog –функциональность, которая будет разработана в течение следующего спринта для достижения цели спринта. Артефакт: Sprint Backlog.
-
- Планирование спринта, митинг второй. Участники: Скрам Мастер, команда Цель: определить, как именно будет разрабатываться определенная функциональность для того, чтобы достичь цели спринта. Для каждого элемента Sprint Backlog определяется список задач и оценивается их продолжительность. Артефакт: в Sprint Backlog появляются задачи Если в ходе спринта выясняется, что команда не может успеть сделать запланированное на спринт, то Скрам Мастер, Product Owner и команда встречаются и выясняют, как можно сократить scope работ и при этом достичь цели спринта.
-
- Остановка спринта (Sprint Abnormal Termination)

- Понятие интеллектуальной собственности.
- Согласно статье 1225 Гражданского кодекса интеллектуальная собственность – это охраняемые законом результаты интеллектуальной деятельности и средства индивидуализации. Основные признаки (характеристики) интеллектуальной собственности:
- а) Интеллектуальная собственность нематериальна. В этом ее главное и важнейшее отличие от собственности на вещи (собственность в классическом смысле). Если у Вас есть вещь, Вы можете пользоваться ей сами или передать в пользование другому лицу. Однако невозможно в один момент времени использовать одну вещь вдвоем независимо друг от друга. Если Вы обладаете интеллектуальной собственностью, Вы можете использовать ее сами и одновременно предоставить права на нее другому лицу. Причем этих лиц могут быть миллионы, и все они могут независимо друг от друга использовать один объект интеллектуальной собственности.
- б) Интеллектуальная собственность абсолютна. Это означает, что одному лицу – правообладателю – противостоят все остальные лица, которые без разрешения правообладателя не вправе использовать объект интеллектуальной собственности. Причем отсутствие запрета использовать объект не считается разрешением.
- в) Нематериальные объекты интеллектуальной собственности воплощаются в материальных объектах. Приобретая диск с

Свободное, несвободное, закрытое, открытое ПО, лицензия GNU GPL, LGPL, BSD, MIT, CDDL.

- В соответствии со ст.1261 Гражданского кодекса РФ, программа для ЭВМ – это представленная в объективной форме совокупность данных и команд, предназначенных для функционирования компьютерных устройств в целях получения определенного результата. Программы для ЭВМ являются объектами авторского права. При этом авторские права распространяются на следующие части программы для ЭВМ:
- исходный текст и объектный код;
- подготовительные материалы, полученные в процессе создания программы для ЭВМ;
- аудиовизуальные отображения, порождаемые программой, в частности, интерфейс, дизайн компьютерных игр и т.п.
- Для использования вышеуказанных элементов программы для ЭВМ каким-либо способом требуется разрешение автора (правообладателя), полученное в форме лицензионного договора. Так, например, согласие автора необходимо для того, чтобы печатать изображения персонажей компьютерных игр на футболках, сувенирах и иной продукции.
- Основным охраняемым элементом программы для ЭВМ является исходный текст. Он существует в языковой форме, т.е. доступен для

- Подобно сюжету художественного произведения, не охраняется авторским правом алгоритм программы для ЭВМ. Алгоритм относится к области содержания, а не формы рассматриваемого объекта авторского права, поэтому является неохраноспособным. С другой стороны, некоторые алгоритмы программ для ЭВМ могут защищаться патентом на изобретение.

- Исключительное право на использование программы для ЭВМ подразумевает право автора использовать программу любыми способами и в любых формах и, соответственно, запрещать такое использование третьим лицам. Наиболее распространенные способы использования программы:
-
- воспроизведение, т.е. создание экземпляров программы в любой материальной форме. Воспроизведением считается и копирование программы для ЭВМ в память компьютера.
- распространение экземпляров программы путем отчуждения в какой-либо форме, в т.ч. путем продажи;
- импорт экземпляров в целях распространения. В связи с этим не допускается без согласия правообладателя ввозить экземпляры в страну в целях продажи, даже если экземпляры были приобретены за границей правомерно. Это не препятствует импорту экземпляра ПО для использования в личных целях;

- Программы для ЭВМ характеризуются рядом особенностей.
- Во-первых, не допускается создание экземпляра программы для ЭВМ в личных целях. Например, купив книгу, Вы можете сделать для себя и членов семьи её копию для использования в личных целях. В отношении программы для ЭВМ допускается только изготовление копии в архивных целях. Какие-либо действия в отношении копии правомерны, если только утерян первоначальный экземпляр. Более того, при прекращении срока действия лицензионного договора архивная копия должна быть уничтожена.
- Во-вторых, в целях обеспечения технической совместимости с компьютером разрешается вносить изменения в программы для ЭВМ и исправлять явные ошибки.
- В-третьих, при соблюдении следующих условий допускается декомпилировать программу для ЭВМ:
- цель декомпиляции – обеспечить взаимодействие между разными программами;
- декомпилируются только те части программы, которые необходимо для достижения вышеуказанной цели;
- исходный текст не был доступен из других источников;
- исходный текст не используются в каких-либо иных целях, в т.ч. для создания аналогичной или схожей программы.
- В-четвертых, условия лицензионного договора на программы для ЭВМ могут быть изложены на экземпляре ПО. Начало использования программы признается согласием с условиями коробочной лицензии

- Первоначальным субъектом авторского права всегда является «физическое лицо, творческим трудом которого создано» произведение науки, литературы или искусства, а также другая интеллектуальная собственность — автор. Ему принадлежит весь комплекс авторских прав — личные неимущественные права и исключительное право (имущественное право) на использование произведения в любой форме и любым не противоречащим закону способом. Лицо, указанное в качестве автора на оригинале или экземпляре произведения, считается его автором, если не доказано иное (презумпция авторства).
-
- Субъектами авторского права также являются лица, обладающие исключительным правом на произведение, которое перешло к ним от автора по различным основаниям (в силу закона или в силу договора). Такие субъекты называются правообладателями

- В большинстве стран мира авторские права действуют в течение жизни автора и либо 50, либо 70 лет после его смерти. В большинстве стран авторские права перестают действовать в конце календарного года, который находится под вопросом.
-

- Международные соглашения, такие как Бернская конвенция,

Единая система программной документации ЕСПД. Понятие – программа (компонент, комплекс).

- Единая система программной документации (ЕСПД) — комплекс государственных стандартов Российской Федерации, устанавливающих взаимосвязанные правила разработки, оформления и обращения программ и программной документации.
- Программу (по ГОСТ 19781-83) допускается идентифицировать и применять самостоятельно и (или) в составе других программ, подразделяется на виды
- Компонент
- Программа, рассматриваемая как единое целое, выполняющая законченную функцию и применяемая самостоятельно или в составе комплекса
- Комплекс
- Программа, состоящая из двух или более компонентов и (или) комплексов, выполняющих взаимосвязанные функции, и применяемая самостоятельно или в составе

Оценка качества программных средств. Критерии качества программ по ISO 9126-1.

- это степень, в которой программное обеспечение обладает требуемой комбинацией свойств.
- Функциональность — Набор атрибутов характеризующий, соответствие функциональных возможностей ПО набору требуемой пользователем функциональности. Детализируется следующими подхарактеристиками (субхарактеристиками):
- Пригодностью для применения
- Корректностью (правильностью, точностью)
- Способностью к взаимодействию (в частности сетевому)
- Защищенностью
- Надёжность — Набор атрибутов, относящихся к способности ПО сохранять свой уровень качества функционирования в установленных условиях за определенный период времени. Детализируется следующими подхарактеристиками (субхарактеристиками):
- Уровнем завершенности (отсутствия ошибок)
- Устойчивостью к дефектам
- Восстанавливаемостью
- Доступностью
- Готовностью
- Практичность (применимость) — Набор атрибутов, относящихся к объему работ, требуемых для исполнения и индивидуальной оценки такого исполнения определенным или предполагаемым кругом пользователей.

- Эффективность — Набор атрибутов, относящихся к соотношению между уровнем качества функционирования ПО и объемом используемых ресурсов при установленных условиях.
- Детализируется следующими подхарактеристиками (субхарактеристиками):
- Временной эффективностью
- Используемостью ресурсов
- Сопровождаемость — Набор атрибутов, относящихся к объему работ, требуемых для проведения конкретных изменений (модификаций). Детализируется следующими подхарактеристиками (субхарактеристиками):
- Удобством для анализа;
- Изменяемостью
- Стабильностью
- Тестируемостью
- Мобильность — Набор атрибутов, относящихся к способности ПО быть перенесенным из одного окружения в другое. Детализируется следующими подхарактеристиками (субхарактеристиками):
- Адаптируемостью
- Простотой установки (инсталляции)
- Сосуществованием (соответствием)

Классификация программного обеспечения. Системное ПО.

- Систе́мное програ́ммное обеспе́чение — комплекс программ, которые обеспечивают управление компонентами компьютерной системы, такими как процессор, оперативная память, устройства ввода-вывода, сетевое оборудование, выступая как «межслойный интерфейс», с одной стороны которого аппаратура, а с другой — приложения пользователя. В отличие от прикладного программного обеспечения, системное не решает конкретные практические задачи, а лишь обеспечивает работу других программ, предоставляя им сервисные функции, абстрагирующие детали аппаратной и

- К системному ПО относятся:
- операционные системы (эта программа загружается в ОЗУ при включении компьютера);
- программы – оболочки (обеспечивают более удобный и наглядный способ общения с компьютером, чем с помощью командной строки DOS, например, Norton Commander);
- операционные оболочки – интерфейсные системы, которые используются для создания графических интерфейсов, мультипрограммирования и.т.;
- Драйверы (программы, предназначенные для управления портами периферийных устройств, обычно загружаются в оперативную память при запуске компьютера);
- утилиты (вспомогательные или служебные программы, которые представляют пользователю ряд дополнительных услуг).
- К утилитам относятся:
- диспетчеры файлов или файловые менеджеры;
- средства динамического сжатия данных (позволяют увеличить количество информации на диске за счет ее динамического сжатия);
- средства просмотра и воспроизведения;
- средства диагностики; средства контроля позволяют проверить конфигурацию компьютера и проверить работоспособность устройств компьютера, прежде всего жестких дисков;
- средства коммуникаций (коммуникационные программы)

Виды системного ПО: операционные системы (ОС). Операционные системы персональных компьютеров и их классификация (многозадачные, однозадачные, реального времени, сетевые).

- Многозадачные операционные системы (Windows, Linux) в отличие от однозадачных (MS-DOS) позволяют одновременно выполнять несколько процессов или имитировать их параллельное выполнение с помощью переключение контекста исполнения, обычно по таймеру с выделением ресурса процессора на какое-то время.
- Сетевые операционные системы имеют возможность обеспечения работы со стеками протоколов различных локальных или/и глобальных сетей. Организовывать интранет, обеспечивают выполнение сетевых сервисов и предоставление их пользователям. См. Сетевые сервисы. Авторизация. Служба каталогов. (UNIX, Novell Netware)
- Реального времени позволяют выполнить операцию за заданное время или операция считается неуспешной (жесткого времени) и позволяют выполнить в среднем

Операционная система MS Windows, дистрибутивы Linux.

- Семейство коммерческих операционных систем (ОС) корпорации Microsoft, ориентированных на применение графического интерфейса при управлении. Изначально Windows была всего лишь графической надстройкой-программой для операционной системы 80-х и 90-х годов MS-DOS. Проприетарные, с закрытым программным кодом.
- Ли́нукс — семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе как правило создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов — в форме, готовой для установки и удобной для сопровождения и обновлений,

Файловая структура операционных систем. Операции с файлами. Файловые системы. FAT (привести пример как устроена таблица). NTFS. EXT2. EXT3.

- Файловая система FAT (File Allocation Table) была разработана Биллом Гейтсом и Марком Макдональдом в 1977 году.
- Сейчас существуют три типа файловой системы FAT:
- FAT12 – поддерживает очень небольшие объемы дисков, поэтому сейчас она применяется только на дискетах.
- FAT16 – используется на винчестерах и поддерживает диски объемом до 2 Гб, поэтому сейчас данная файловая система практически не используется.
- FAT32 – теоретически поддерживаются диски объемом до 2 Тб.
- – FAT32
- Основным преимуществом этой файловой системы является ее простота и совместимость со старыми операционными системами. Для этой файловой системы существует большое количество подробной документации. Существует ограничение на максимальный размер файла – 4 Гб. Сбои в системе часто приводят к повреждением одного или нескольких файлов. Однако, при серьезных повреждениях, восстановить информацию гораздо проще, чем в случае с NTFS.
- – NTFS
- Основными преимуществами файловой системы NTFS являются ее защищенность от несанкционированного доступа. В этой файловой системе отсутствуют ограничения на размер файлов и каталогов. Так же ее особенность является журналирование – запись всех операций перед их

- Second Extended File System (дословно: «вторая расширенная файловая система»), сокращённо ext2 (иногда ext2fs) — файловая система ядра Linux. Была разработана Реми Кардом (англ.) взамен существующей тогда ext. По скорости и производительности работы она может служить эталоном в тестах производительности файловых систем. Так, в тестах на скорость последовательного чтения и записи, проведённых The Dell TechCenter, файловая система ext2 обгоняет ext3 и уступает лишь более современной ext4 в тесте на чтение.
-
- Главный недостаток ext2 (и одна из причин демонстрации столь высокой производительности) заключается в том, что она не является журналируемой файловой системой. Он был устранён в файловой системе ext3 — следующей версии Extended File System, полностью совместимой с ext2. Но для ssd это скорее плюс, поскольку продлевает жизнь накопителя. Это основная причина, почему EXT2 до сих пор поддерживается в Anaconda и Ubiquity.
-
- Файловая система ext2 по-прежнему используется на флеш-картах и твердотельных накопителях (SSD), так как отсутствие журналирования является преимуществом при работе с накопителями, имеющими ограничение на количество циклов

Загрузочный сектор	Таблица FAT (1-я копия)	Таблица FAT (2-я копия)	Корневой каталог	Область данных

- В таблице файлов FAT хранятся указатели номера ячейки в таблице FAT соответствующей номеру кластера (позиция ячейки соответствует номеру кластера). Каждая ячейка хранит указатель на следующий кластер в содержании файла и соответственно на следующую ячейку где

Прикладное ПО

- Прикладные программы могут использоваться автономно или в составе программных комплексов или пакетов.
-
- Прикладное ПО – программы, непосредственно обеспечивающие выполнение необходимых работ на ПК: редактирование текстовых документов, создание рисунков или картинок, создание электронных таблиц и т.д.
-
- Пакеты прикладных программ – это система программ, которые по сфере применения делятся на проблемно – ориентированные, пакеты общего назначения и интегрированные пакеты. Современные интегрированные пакеты содержат до пяти функциональных компонентов: тестовый и табличный процессор, СУБД, графический редактор, телекоммуникационные средства.
-
- К прикладному ПО, например, относятся:
-
- Комплект офисных приложений MS OFFICE.
- Бухгалтерские системы.
- Финансовые аналитические системы.
- Интегрированные пакеты делопроизводства.
- CAD – системы (системы автоматизированного проектирования).

СУБД. Системы управления базами данных. Примеры. Назначение. (Записи, поля, основные базовые операции, транзакция)

- СУБД предназначено для управления и предоставления привилегированного доступа к базе данных, выполнения транзакций (обеспечения их атомарности, независимости), восстановления данных в результате сбоев (мягких – отключения питания (системный журнал), жестких – потеря жесткого диска (резервное копирование)), сетевого доступа. База данных – формализованное описание предметной области в виде совокупности взаимосвязанных записей. Запись набор полей определенных на домене.
- Транзакция – несколько операций воспринимаемых как неделимое целое, выполняются либо все операции, либо происходит откат всех выполненных операций, если ошибка, для этого используется системный журнал. Например – перевод со счета на счет, считать данные с одного счета, вычесть со счета, записать, считать с другого счета, добавить, записать (если выполнить наполовину, потеряем деньги).
- Базовые операции – выборка – select * from table where id = 1
- Удаление – Delete from table where id =2
- Обновление – Update table set оценка = 5 where FIO = «Иванов»

Прикладное ПО. Текстовые и табличные процессоры (MS Word, MS Excel, LibreOffice Writer, Calc). Назначение, основные характеристики.

- Текстовые процессоры предназначены для форматирования (задание формы представления), редактирования (исправление текстового содержания), сохранения, печати, добавления рисунков и других объектов в текстовых документ, выполнение сложных операций над текстом с помощью макросов.
- Табличные процессоры нужны для вычислений, где данные представлены в табличном виде, с адресацией ячейки в

Графические редакторы. Системы деловой (инженерной) графики.

- Photoshop, GIMP, Paint, 3dMax.

Системы автоматизированного проектирования (САПР, CAE, CAD).

- САПР (CAD,CAE,CAM)– система автоматизированного проектирования.
- CAD - средства САПР, предназначенные для автоматизации двумерного и/или трехмерного геометрического проектирования, создания конструкторской и/или технологической документации, и САПР общего назначения.
- CAE более общее понятие чем CAD и CAM, включающее любое использование компьютерных технологий в инженерной деятельности.
- Computer-aided manufacturing (CAM) — система автоматизированной разработки программ обработки деталей для станков с ЧПУ или ГАПС (Гибких автоматизированных производственных систем)).
- Основная цель создания САПР — повышение эффективности труда инженеров, за счет автоматизации работ на стадиях проектирования и подготовки производства. Так, благодаря САПР, удается добиться:
-
- - сокращения трудоёмкости проектирования и планирования;
- - сокращения сроков проектирования;
- - сокращения себестоимости проектирования и изготовления, уменьшение затрат на эксплуатацию;
- - повышения качества и технико-экономического уровня результатов проектирования;
- - сокращения затрат на натурное моделирование и испытания.
-
- В качестве входной информации САПР использует технические знания специалистов, которые вводят проектные требования, уточняют результаты, проверяют полученную конструкцию, изменяют ее и т.д.

PDM, PLM

- PDM-система (англ. Product Data Management — система управления данными об изделии) — организационно-техническая система, обеспечивающая управление всей информацией об изделии. При этом в качестве изделий могут рассматриваться различные сложные технические объекты (корабли и автомобили, самолёты и ракеты, компьютерные сети и др.).
- Product Lifecycle Management (PLM) — жизненный цикл продукта, изделия. Продразумевается совокупность процессов, выполняемые от момента выявления потребностей общества в определенном продукте до утилизации изделия после его использования. Понятие применимо для любого

корпоративные информационные системы (ERP, MRP, CRM).

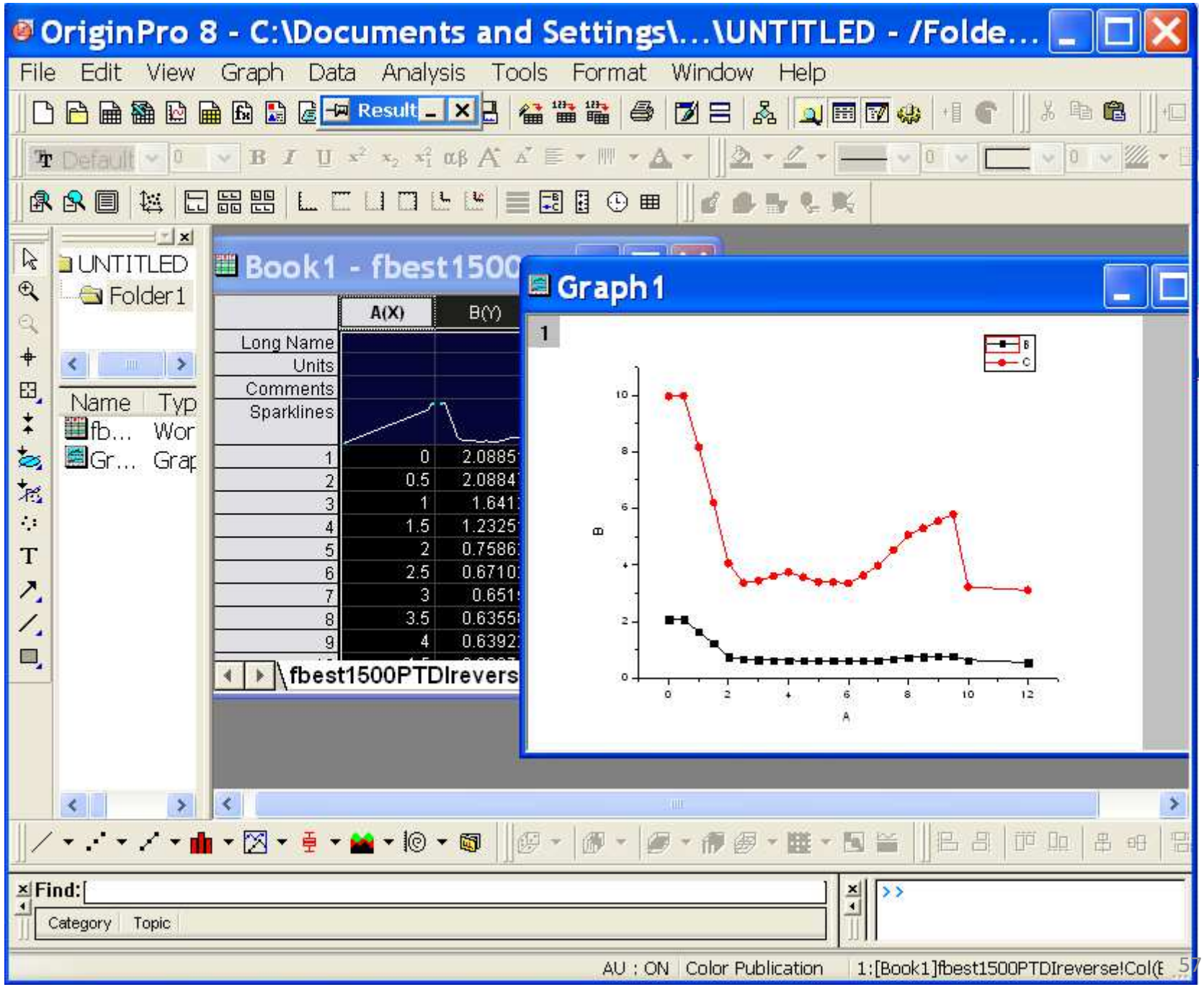
- ERP (англ. Enterprise Resource Planning, планирование ресурсов предприятия) — организационная стратегия интеграции производства и операций, управления трудовыми ресурсами, финансового менеджмента и управления активами, ориентированная на непрерывную балансировку и оптимизацию ресурсов предприятия посредством специализированного интегрированного пакета прикладного программного обеспечения, обеспечивающего общую модель данных и процессов для всех сфер деятельности. ERP-система — конкретный программный пакет, реализующий стратегию ERP.
- MRP (англ. Material Requirements Planning — планирование потребности в материалах) — система планирования потребностей в материалах, одна из наиболее популярных в мире логистических концепций, на основе которой разработано и функционирует большое число микрологистических систем.
- Система управления взаимоотношениями с клиентами (CRM, CRM-система, сокращение от англ. Customer Relationship Management) — прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности для

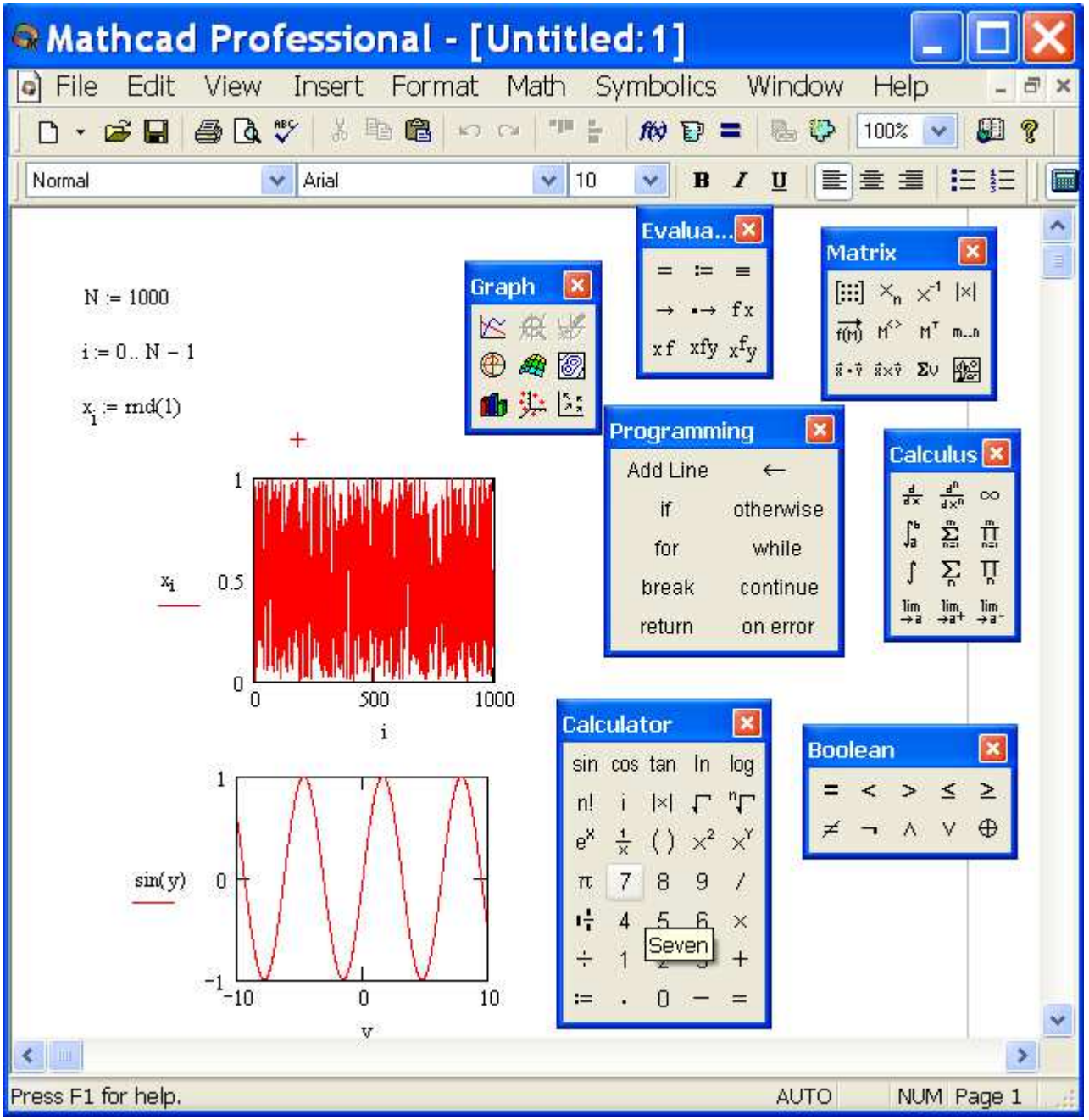
Системы управления версиями, управления проектами (SVN, git).

- Система управления версиями (от англ. Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.
-
- Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы. Однако они могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов. В частности, системы управления версиями применяются в САПР, обычно в составе систем управления данными об изделии (PDM). Управление версиями используется в инструментах конфигурационного управления (Software Configuration Management Tools).
- SVN, GIT - примеры.

Научное ПО. MathCad. Octave. Matlab. Origin. QtPlot. SciLab.

- Позволяют рисовать графики по данным в файлах, проводить вычисления, осуществлять решение систем линейных уравнений, проводить матричные вычисления, решать системы нелинейных уравнений, проводить дифференцирование, интегрирование, как численное, так и аналитическое, строить сплайны, проводить анализ сигналов, рассчитывать статистические характеристики


OriginPro 8 - C:\Documents and Settings\...\UNTITLED - /Folde...
File Edit View Graph Data Analysis Tools Format Window Help
Result
UNTITLED
Folder1
Name Typ
fb... Wor
Gr... Grap
Book1 - fbest1500
A(X) B(Y)
Long Name
Units
Comments
Sparklines
1 0 2.0885
2 0.5 2.0884
3 1 1.641
4 1.5 1.2325
5 2 0.7586
6 2.5 0.6710
7 3 0.651
8 3.5 0.6355
9 4 0.6392
fbest1500PTDIrevers
Graph1
1
B
C
B
A
Find:
Category Topic
AU : ON Color Publication
1:[Book1]fbest1500PTDIreverse!Col(


Mathcad Professional - [Untitled:1]
File Edit View Insert Format Math Symbolics Window Help
Normal
Arial
10
100%
N := 1000
i := 0.. N – 1
x_i := rnd(1)
Graph
Evalua...
Matrix
Programming
Add Line
if
for
break
return
otherwise
while
continue
on error
Calculus
Calculator
sin cos tan ln log
Seven
Boolean
x_i
sin(y)
i
y
Press F1 for help.
AUTO
NUM Page 1

Системы поддержки принятия решений. Примеры.

- Систе́ма подде́ржки приня́тия реше́ний (СППР) (англ. Decision Support System, DSS) — компьютерная автоматизированная система, целью которой является помощь людям, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности. СППР возникли в результате слияния управленческих информационных систем и систем управления базами данных.
-
- Для анализа и выработок предложений в СППР используются разные методы. Это могут быть: информационный поиск, интеллектуальный анализ данных, поиск знаний в базах данных, рассуждение на основе прецедентов, имитационное моделирование, эволюционные вычисления и генетические алгоритмы, нейронные сети, ситуационный анализ, когнитивное моделирование и др. Некоторые из этих методов были

Системы Искусственного интеллекта (Экспертные системы, Нейросетевые технологии).

- Экспертная система – система основанная на знаниях и призванная заменить человека эксперта в какой-то предметной области. Например,MyCIN для диагноза болезней и т.д. Обычно использует продукционную модель знаний – правила если – то, которые позволяют выводить одно знание из другого путем обработки правил машиной вывода. Может работать с ненадежными знаниями для которых указывается коэффициент уверенности или

Утилиты. Назначение утилит и их классификация по функциональному признаку.

- Программы обслуживания дисков, программы архивирования данных, программы обслуживания сети.

Вредоносные программы (определение).

- Вредоносная программа - любое программное обеспечение, предназначенное для получения несанкционированного доступа к вычислительным ресурсам самой ЭВМ или к информации, хранимой на ЭВМ, с целью несанкционированного использования ресурсов ЭВМ или причинения вреда (нанесения ущерба) владельцу информации, и/или владельцу ЭВМ, и/или владельцу сети ЭВМ, путём копирования, искажения, удаления или подмены информации. Многие антивирусы считают крэки (кряки), кейгены и прочие программы для взлома приложений вредоносными программами, или потенциально опасными. С точки зрения УК нужны три критерия:
- Уничтожение информации или нарушение работы. Таким образом, взломщик защиты от копирования — не вредоносная программа. Однако иногда во взломщики добавляют «троянских коней». тут нарушается другой закон связанный с модификацией исходного кода – авторские права.
- Несанкционированная работа. Определяется обычно от противного: для санкционированной работы программа должна предупредить пользователя о своей опасности и не исполнять опасные функции в неожиданные моменты. Программа форматирования диска, входящая в комплект любой ОС, уничтожает данные, но не является вредоносной, так как её

Виды вирусов (полиморфные, стеллс, троянские, черви). Бот-неты.

- Вирус – программа копирующая сама себя в исполнимые файлы, загрузочные области диска, другие виды файлов, например, выполняющие макросы, выполняющая какие-либо действия заложенные в нее.
- Полиморфный вирус – видоизменяет свой код постоянно, шифрующий свой код, чтобы трудно было найти по сигнатурам.
- Стеллс – скрывает себя от сканнера, путем перехвата прерывания обращения к диску или каким-либо способом копирующим свой код в разные места исполнимого файла, чтобы труднее было обнаружить.
- Троянские вирусы – вирус заражающий компьютер пользователя и дающий возможность внешнего управления злоумышленником.
- Черви – вирусы распространяющиеся путем проникновения через сетевые уязвимости по глобальным или локальным сетям.
- Бот-нет – эшелонированный набор управляемых компьютеров с троянскими вирусами, позволяющих выполнять DDOS атаки на

Антивирусные программы.

- Сканнер, Монитор.
- Сканнер ищет вирусы по сигнатуром – слепки кода вирусов.
- Монитор осуществляет поиск вирусов резидентно, проверяет запускаемые исполнимые файлы, память.
- Примеры….

Хранение и обработка видео, изображений и звуковой информации. Методы сжатия данных. Сжатие видео, изображений и звуковой информации

- Изображения хранятся в файлах в виде наборов пикселей. Несжатый файл может занимать большое место. Для сжатия применяются методы сжатия с потерями и затем без потерь. Сжатие с потерями основано на преобразовании исходного представления изображения по пространственным координатам в частотную область координат (при этом обычно изображение разбивается на части, например на блоки 8 на 8 пикселей) и подвергается переводу из RGB в HSB, переход в частотное представление реализуется применяя дискретно-косинусное преобразование или Wavelet. Это позволяет убрать далее части сигнала (изображение это тоже сигнал) с высокой частотой, которые слабо влияют на обратное преобразование и восстановление исходного образа. Так же при таком способе появляется множество маленьких значений цветовой интенсивности и небольшое количество больших значений, что позволяет далее применить метод сжатия с без потерь – например Хаффмана (Шеннона-Фано). Так же типичной схемой является использование вычитания (конечных разностей), очевидно что соседние одинаковые пиксели вычитаясь друг из друга дадут ноль, что приведет ко множеству маленьких значения, которые можно сжать Хаффманом. Вычитание хороший способ сжать видео, ведь на повторяющихся кадрах много повторяющихся картинок, потому например сохранив блоки опорного кадра и найдя на следующем кадре наиболее похожий блок, можно осуществить вычитание, и сохранить двумерный сдвиг блока относительно опорного. Вычтенная разностная картинка легко

Векторная и растровая графика.

- Растр – изображение состоящее из пикселей, имеет разрешение по вертикали и горизонтали, сколько то байт отводится под пиксель, для полноцветных обычно отводится три байта соответственно красного, синего, зеленого (иногда добавляется прозрачность, 4-й байт). Если один байт на пиксель, то это обычно файл с палитрой, где значению байта в соответствие ставится набор из трех комбинаций цветов, потом хранятся просто номера палитры. Для рисования можно применять различные способы заполнения, кисти и т.д.
- Векторная графика – используются готовые

Гипертекстовые документы, HTML, XML.

- HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.
- <html>
- <head>
- <meta charset="utf-8" />
- <title>HTML Document</title>
- </head>
- <body>
- <p>
- <b>
- Этот текст будет полужирным, а этот — ещё и курсивным

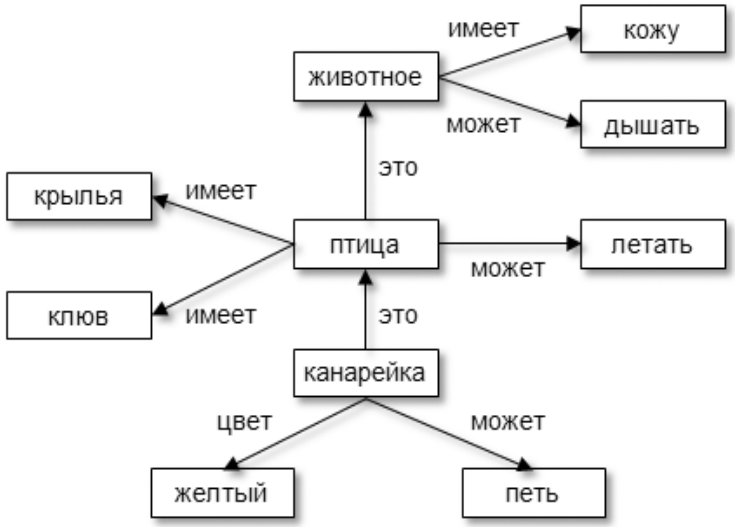
XML (eXtensible Markup Language) — расширяемый язык разметки. Рекомендован Консорциумом Всемирной паутины (W3C). Позволяет описать данные.

<?xml version="1.1" encoding="UTF-8" ?>
<tag name = “name”>
Привет
</tag>

Можно обработать с помощью XSLT и получить любой файл с любым содержанием.

Представление знаний на сетях, семантическая паутина и Web онтологии.

- Семантические сети позволяют представить знания о предметной области в виде понятий и связей между ними графически, понятия и объекты описываются в овалах, а связи описываются стрелками.
- Web онтология или семантическая паутина ставит своей целью создать машинно-обрабатываемое представление сайтов описывающих различные предметные области, например с помощью языка OWL. Что позволит отвечать на запросы



Значение моделирования при решении задач в профессиональной области.

- Классификации моделей. Классификация по области использования (учебные, опытные, научно – технические, игровые, имитационные).
- Классификация с учетом фактора времени (Статическая и динамическая модели). Классификация по способу представления (Материальные и информационные модели). Знаковые и вербальные информационные модели.

Методы решения инженерной задачи с помощью ЭВМ. Аналитические, графические, комбинированные и численные методы.

- Постановка проблемы
- Выбор или построение математической модели
- Постановка вычислительной задачи
- Предварительный анализ свойств вычислительной задачи (на корректность, устойчивость)
- Выбор и построение численного метода
- Алгоритмизация и программирование
- Отладка программы
- Расчеты с помощью программы
- Обработка и интерпретация результатов

Погрешности. Способы уменьшения погрешностей.

- Погрешности и нечеткости встречаются постоянно. При измерениях, при подсчетах. Нечеткости могут быть обусловлены пропуском в измерениях, потерями данных, их искажениями при передаче. Лексические формы могут иметь собственные нечеткости обусловленные расплывчатостью понятия – например, легкий, мягкий, твердый, быстрый. Быстрый автомобиль, быстрый человек. Быстрая скорость это как – 30 км/ч, 300 км/ч. И т.д.
- Для уменьшения погрешности обычно повторяют измерение множество раз. При этом погрешность падает пропорционально корню из количества измерений. Прибегают к правильному применению арифметических операций – изучают на численных методах.
- Для избавления от проблемы потерь и искажений при передаче – передают данные повторно, включают

Аутентификация. Авторизация.

- Авторизация
- Наделение правами и привилегиями пользователя желающего начать сеанс взаимодействия с программой, сетевым сервисом. Для получения данных привилегий пользователь обычно проходит аутентификацию.
- Аутентификация
- Проверка пользователя перед тем как его авторизовать с помощью какого-либо метода подтверждения его личности и его прав.

Криптографические методы защиты данных. Шифрование, электронная подпись.

- Шифрование основано на применении к тексту или/и данным текстового ключа нескольких раундов перестановок, подстановок, но обращаемых. Очевидно, что сам алгоритм шифрования может быть известен, тогда как ключ является секретным.
- Симметричные методы шифрования – используют один ключ для шифрования и расшифровки. Ассиметричные методы шифровани – используют два ключа, один закрытый, другой открытый. При этом применение закрытого ключа к тексту зашифрованному открытым ключом позволяет расшифровать исходное сообщение, то есть с точки зрения функционального применения З(О(Т)) = Т. Типичная схема защиты канала связи заключается в создании открытого и закрытого, ключа, отправки открытого ключа своему «визави», который шифрует текст открытым ключом и отправляет вам шифрованный текст, вы расшифровываете его закрытым ключом.
- Также это можно использовать для создания цифровой подписи. Мы хотим дать возможность проверить что наш текст не изменялся. Шифруем текст или хэш функцию от текста, с помощью закрытого ключа. Публикуем открытый ключ. Отправляем шифрованный текст – цифровая подпись, и открытый текст. Тот кто

Коммутация каналов, коммутация пакетов, коммутация сообщений.

- Коммутация сообщений – отправляемое сообщение целиком передается другому узлу, и хранится там какое-то время пока не будет забрано целевым узлом. (Пример FidoNet, электронная почта).
- Коммутация каналов – между двумя узлами в коммутационной среде выстраивается путь – канал, занимающий постоянно ресурс сети (временной, частотный) в течение сеанса связи, ресурс освобождается после того как заканчивается сеанс. Неэффективно используется ресурс сети, так как например канал может и не использоваться, примет телефонный разговор – канал занят, а собеседники молчат, сложно выделить ресурс для нового абонента, но есть гарантия постоянной пропускной способности.
- Коммутация пакетов – исходное сообщение разбивается на несколько пакетов, каждый из которых независимо

Локальные сети: принципы построения, архитектура, основные компоненты, их назначение и функции.

- Связывают несколько узлов (10-100) на небольшой территории обычно одной средой передачи данных. Обычно однородное программное и аппаратное обеспечение, простота в использовании и настройке.
- Проводные технологии.
- Сети с топологией кольцо. (Token Ring, FDDI) – по кольцу (узлам объединенным последовательно) двигается маркер в одном направлении, который захватывается узлом желающим отправить данные, вместо маркера отправляется кадр с данными, кадр доходит до узла назначения, и тот посылает далее кадр подтверждения который принимается узлом отправителем, после чего он отпускает маркер и маркер снова курсирует по кругу.
- Сети с топологией шина (Ethernet) – общая среда для передачи, узел желающий отправить кадр с данными проверяет свободна ли среда от сигнала, если да, то кадр отправляется. Может быть так, что два узла одновременно обнаружили что среда свободна и начали передавать данные , что приводит к искажению сигнала, что обнаруживается всеми узлами прослушивающими среду, в том числе и отправляющими, в этом случае передача прерывается и возобновляется снова через случайное время. Разрыв шины

Глобальные сети: принципы построения, архитектура, основные компоненты, их назначение и функции.

- Глобальные сети - объединение множества разнородных сетей. Занимают континенты, включают миллионы узлов и пользователей, включают разнородное программное и аппаратное обеспечение. Сложнее в управлении локальных.
- Интернет.
- FidoNet.
- Спутниковые глобальные сети (GlobalStar, Iridium, VSAT)

Сетевые сервисы.

- Файловый сервис – предназначен для передачи файлов по сети. (FTP)
- Сервис печати – предназначен для передачи на печать документов по сети на принтеры, факсы, удаленного управления устройством выполняющего задания. Для снижения расходов на закупку оборудования и удобного доступа к устройству в офисах и отделах.
- Сервис приложений (SAAS, PAAS, IAAS). Предоставление вычислительных мощностей удаленно пользователю. SAAS – предоставление готового приложения. PAAS – предоставление средств разработки (обычно для создания web-сервиса, пр. Microsoft Azure, Heroku, Amazon web services, Google App Engine). IAAS – инфраструктура как сервис, вычислительная инфраструктура для организации сети, с установкой операционных систем и т.д.
- Сервис сообщений – электронная почта (SMTP, POP3), мгновенные сообщения (SMS).
- Сервис СУБД – предоставления удаленного доступа к базе данных. Файл-серверная архитектура, база данных хранится на отдельном сервере, а СУБД разворачивается на конечных узлах пользователей (неудачное решение из-за невозможности организовать синхронизацию и распараллеливание привилегированного доступа). Сервер с СУБД – СУБД разворачивается на отдельном

Понятие и модели протоколов обмена информацией, семиуровневая модель.

- Модель взаимодействия открытых систем или OSI (open system interaction). Разбивает взаимодействие на несколько уровней для обеспечения независимости прикладных сервисов от физической сети, независимое развитие уровней, технологий, устройств, программного обеспечения. Например, на физическом уровне – антенны, адаптеры. На сетевом – маршрутизаторы, на прикладном – протоколы torrent, почтовые.
- Прикладной (SMTP, POP3, HTTP) – сетевые сервисы
- Представительский (SSL, base64, ASCII, MIME) - кодирование, декодирование, шифрование данных.
- Сеансовый – отвечает за сеанс связи, установление соединения, закрытие, синхронизацию взаимодействия. TCP, SCTP.
- Транспортный уровень – отвечает за упорядочивание доставки сегментов, байт при взаимодействии точка-точка (двух приложений), надежность доставки. (TCP, UDP, SCTP). Не рассматривает сетевую структуру, воспринимает связь как прямую.
- Сетевой уровень – позволяет маршрутизировать пакеты, рассматривает сетевую структуру в виде графа с маршрутизаторами отвечающую за оптимальную доставку данных. Связывает разнородные сети между

Основные принятые в мире протоколы. (IP, TCP, UDP, SMTP, HTTP, POP3). Электронная почта.

- IP – протокол сетевого уровня задающий формат IP пакета, включающего адреса назначения и отправления (контрольную сумму, время жизни и т.д.), маршрутизирующегося в сетях Интернет устройствами маршрутизаторами (в общем случае сеть представляет собой граф) между конечными узлами.
- TCP (transfer control protocol) - протокол сеансового и транспортного уровня обеспечивающего в сетях Интернет установление соединения путем тройной отправки сегментов между двумя приложениями и надежную передачу потока данных за счет повторной отправки не подтвержденных за заданное время таймера сегментов в которых нумеруется каждый байт потока.
- UDP (user datagram protocol) – предполагает ненадежную отправку дейтаграмм на порт (приложение) назначения. Дейтаграмма может и не дойти при отправке.
- SMTP (send message transfer protocol) – протокол передачи почтовых сообщений в сетях интернет. vasia@mail.ru

Среды передачи данных. Модемы. Спутниковые и оптоволоконные каналы связи.

- Беспроводные каналы передачи. Для передачи используется модуляция электромагнитной волны, частотная, фазовая или/и амплитудная (характеристики синусоидально - косинусоидального сигнала), для модуляции используется модем (модулятор-демодулятор).
- Проводные каналы. Оптоволокно – прозрачный материал покрытый слоем со значением коэффициента преломления ниже внутреннего и за счет полного внутреннего отражения оптический или инфракрасный лазерный луч распространяется по проводу покрытому кевларом для защиты.
- Витая пара – пары перекрученных медных изолированных проводников с целью подавления помех при дифференциальной схеме приема (вычитании двух сигналов на приемнике, при этом шум нивелируется, в предположении что он аддитивный P11 = P1+e, P22 = P2+e, P11-P22 = ?).
- Коаксиальный кабель – изолированный медный провод покрытый медной оплеткой. Информация передается в жиле и в оплетке, схема так же дифференциальная, оплетка – экран, подавляет помехи.