**Definition and understanding** - определение и понимание

Science – наука

Основы инженерии

1. Математика – алгебра, логика, топология

2. Электроника – аналоговая электроника (шины, их работа)

3. Алгоритмы – Теории

4. Абстракция – инструментарии идей, компоненты, языки, их связи.

Computational predictability – вычислительная предсказуемость

1. Determinacy - Детерминированность (от лат. determinans — определяющий) — определяемость. **Процесс,** **исход** **которого** **полностью** **определен** **алгоритмом,** **значениями** **входных** **переменных** **и** **начальным** **состоянием** **системы**
2. Indeterminacy – неопределенность

Modeling –

Task solving model - модель решения задач

Result model - модель результата

1.Императивная модель – шаг за шагом

2.Потоки событий (результат, как формула)

Блок-схемы

Paradigms (in CS) -парадигмы

Programming paradigm – парадигма программирования

Paradigm comparison example

1. Imperative - step-by-step (шаг за шагом)
2. Functional

Нет четкой последовательности операций (разный порядок. Параллельно)

Нет переменных

Не может быть циклов.

Есть только функции и их аргументы.

**Парадигмы могут быть:**

**Не структурное программирование** (ассемблер, бейсик)

**Структурное программирование** (блоки кода, функции, видимостьпеременных)

**Процедурное программирование** (разбивает программу наразные блоки)

**Объектно-ориентированное программирование**

**Логическое программирование**

**Декларативное программирование**

**Технологии и техники программирования**

Асинхронные

Параллельные и распределенные

Объектно-ориентированные

Визуальные из визуальных задач формирует голый код, который мы можем уже заполнить

Two main fieds

Системное - Пишут инструменты

Прикладное – Пишут программы

Программу мы можем разделять на несколько программ(процессов), а так же несколько слоев. Можем отделять прикладной код о системного.

Date & Logic

Separation

1. API+data structures
2. Anemic domain modeling
3. Crud principles
4. MVC, MVP, MVVP patterns

Combination

1. Imperative, procedural, structural, generic programming and actor model
2. Functional programming
3. Object-oriented programming
4. Component approach

Development – создание, разработка, совершенствование

1. Продавец
2. Покупатель
3. Пользователи
4. Программисты
5. Тестирование

Development approaches - подходы к развитию

1. Data-centric
2. Process-centric
3. User interface-centric

Hybrid approach - гибридный подход

Data model

Process model

GUI model

Т.е. пользователю делаем интерфейс, но без привязки к программе для демонстрации.

По факту привязка будет осуществлена в конце.

Классификация программных систем:

Окна, базы данных, сетевые сервисы, библиотеки, клиент-серверы и т.п.

**Введение в программирование**

Инструкция или оператор – наименьшая синтаксическая часть языка программирования, исполняемая интерпретатором, средой или компилируемая в машинный код.

Операция – иногда имеют в виду оператор(operator), а иногда инструкцию (statement)

Команда –атомарная задача для интерпретатора или CLI (командной строки)

Выражение(expression) – синтаксическая конструкция языка программирования, предназначенная для выполнения вычислений. Выражение состоит из идентификаторов, значений, операторов и вызова функций.

Оператор -

Директива – инструкция препроцессора

Значение(value) – величина, записанная в определенное место памяти в опредеоенном формате и представляющая данные, которым может манипулировать программа.

Тип(type) – множество значений и операции, которые могут быть произведены над этими значениями. Например, в JavaScript тип Boolean предполагает два значения True & false и логические операции над ними (и, или, не). Тип Null, предполагает одно значение null, а тип number множество рациональных чисел с дополнительными ограничениями на минимальное и максимальное значение, а также ограничения на точность и операции:

+-\* \*\* / % ++ -- > < >= & | ~^ << >>

Литерал(literal)-запись значения в коде программы. Могут иметь различный синтаксис, от очень простого, для записи чисел, до сложных синтаксических конструкций, для записи объектов.

Скаляр (Scalar/primitive/atomic value)-значение примитивного типа данных.

Скаляр копируется при присвоении и передается в функцию по значению.

Ссылка(reference)- указывает на значение ссылочного типа, т.е. сложное (не скалярное) значение.

Например, для JavaScript это подтипы: Object, Function, Array

Синтаксис – правила построения программного кода из символов, но не затрагивающие смысловую (семантическую) нагрузку кода. Синтаксис определяет только формальную структуру кода.

Statements

Операторы присваивания

Операторы сравнения

Арифметические операторы

Бинарные и логические операторы

Строковые операторы

Оператор вызова функции

Унарные и тернарный оператор

Идентификатор – имена переменных, констант, функций, методов, аргументов, классов, как внутренние, так и импортированные из других модулей и глобальные.

Переменная(Variable)-именованная область памяти (идентификатор), имеющая тип данных, адрес и значение. Мы можем менять значение переменной в отличие от константы (а для некоторых языков и тип):

Let cityName = ‘Beijing’

Присвоение – связывания значения и идентификатора (например, переменной). Операция присвоения во многих языках возвращает присваиваемое значение (имеет поведение выражения).

Константа – идентификатор, с которым связано неизменяемое значение и тип:

Константу менять нельзя.

const WATCH\_TIMEOUT =5000;

Структурные типы (composed types) – композитные типы или структуры состоят из нескольких скалярных значений, объединенных в одно таким образом, чтоб над этим объединенным значением можно выполнять набор операций, например: объект, массив, множество, кортеж.

Флаг(Flag) – обычно логическое значение, определяющее состояние чего-либо, например, признак закрытия соединения, признак завершения поиска по структуре данных и т.д. Например: let flagName = falce; Иногда флагами могут называть не логические, перечислимые или другие типы.

Алгоритм(Algorithm)-это формальное описание порядка вычислений для определенного класса задач за конечное время (способность завершаться для любого множества входных данных).

Программа(Programm) – программный код и данные, объединенные в одно целое для вычисления и управления ЭВМ.

Инженерия(Engeneering) – извлечение практической пользы из имеющихся ресурсов при помощи науки, техники, различных методик, организационной структуры, а также приемов и знаний.

Инженерия программного обеспечения – приложение инженерии к индустрии программного обеспечения. Включает архитектуру, исследование , разработку, тестирование. Развертывание и поддержку ПО

Программирование – это искусство и инженерия решения задач при помощи вычислительной техники.

Кодирование(coding) –написание исходного кода программы при помощи определенного синтаксиса(языка), стиля и парадигмы по готовому ТЗ.

Разработка программного обеспечения – это соединение программирования и кодирования на всех этапах жизненного цикла ПО.

Блок кода (block of code) – логически связанная группа инструкций или операторов. Блоки создают область видимости. Блоки могут быть вложенными. Пример: {},(+ a b),begin end, отступы в Python.

Процедура или подпрограмма (procedure or Subroutine)- логически связанная группа инструкций или операторов, имеющая имя. Процедура способствует повторному использованию кода и может быть вызвана из разных частей программы, много раз и с разными аргументами. Процедура не возвращает значений, в отличии от функций, но может модифицировать свои аргументы.

Функция – абстракция преобразования значений. Функция однозначно отображает одно множество значений в другое множество значений. Функция может быть задана блоком операторов или выражением.

Сигнатура функции – это имя(идентификатор), количество аргументов и их типы(а иногда и имена аргументов), тип результата.

Метод (Method) – процедура, связанная с объектным контекстом или программным интерфейсом;

{a: 10, b: 10, sum() { return this.a + this.b;}}

Цикл(loop) – многократное исполнение блока операторов

Условие (Conditional statements) – синтаксическая конструкция, позволяющая выполнить разные действия или возвращающая разные значения(тернарный оператор) в зависимости от логического выражения( возвращающего true или false).

Парадигма программирования – парадигма задает набор идей и понятий, допущений и ограничений, концепций, принципов, постулатов, приемов и техники программирования для решения задач на ЭВМ.

Парадигма предлагает модель решения задач, определенный стиль, шаблоны применяемых для написания программного кода.