

ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Основы программирования, 1991
лекций 39 часа, практик 78 часов
3 часа в неделю

- ⦿ Современное общество невозможно представить без компьютера. Они настолько широко и глубоко внедрились в нашу жизнь, что очень трудно назвать какую-либо сферу деятельности человека, где бы они не использовались
- ⦿ В связи с этим серьезные требования предъявляются и к аппаратной части современных компьютеров, и к используемому программному обеспечению
- ⦿ В основном именно программное обеспечение, программные продукты, обеспечивают возможность широкого использования компьютеров

ПРОГРАММА - ЭТО...

- ⦿ Программа - это набор последовательных команд (алгоритм) для определенного объекта (исполнителя), который должен их выполнить для достижения той или иной цели.

ПРИМЕРЫ ПРОГРАММЫ

- ⦿ Инструкция для техника «Как собрать компьютер»
- ⦿ Инструкция для решения квадратного уравнения
- ⦿ ...

ОСНОВНЫЕ ПОНЯТИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Возможности компьютера как технической основы системы обработки данных связаны с используемым программным обеспечением (программами).

- ◎ **Программа** (program, routine) - упорядоченная последовательность команд (инструкций) компьютера для решения задачи.
- ◎ **Программное обеспечение** (software) - совокупность программ обработки данных и необходимых для их эксплуатации документов.
- ◎ **Задача** (problem, task) - проблема, подлежащая решению.
Приложение (application) - программная реализация на компьютере решения задачи.

ПРЕДМЕТНАЯ (ПРИКЛАДНАЯ) ОБЛАСТЬ (APPLICATION DOMAIN)

- ⦿ совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей.

К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ ОТНОСИТСЯ ОБЛАСТЬ ДЕЯТЕЛЬНОСТИ ПО ЕГО ПРОЕКТИРОВАНИЮ И РАЗРАБОТКЕ:

- ⦿ Технология проектирования программ (например, структурное, объектно-ориентированное);
- ⦿ Методы тестирования программ;
- ⦿ Методы доказательства правильности программ;
- ⦿ Анализ качества разработки программ;
- ⦿ Документирование программ;
- ⦿ Разработка и использование программных средств, облегчающих процесс проектирования ПО.

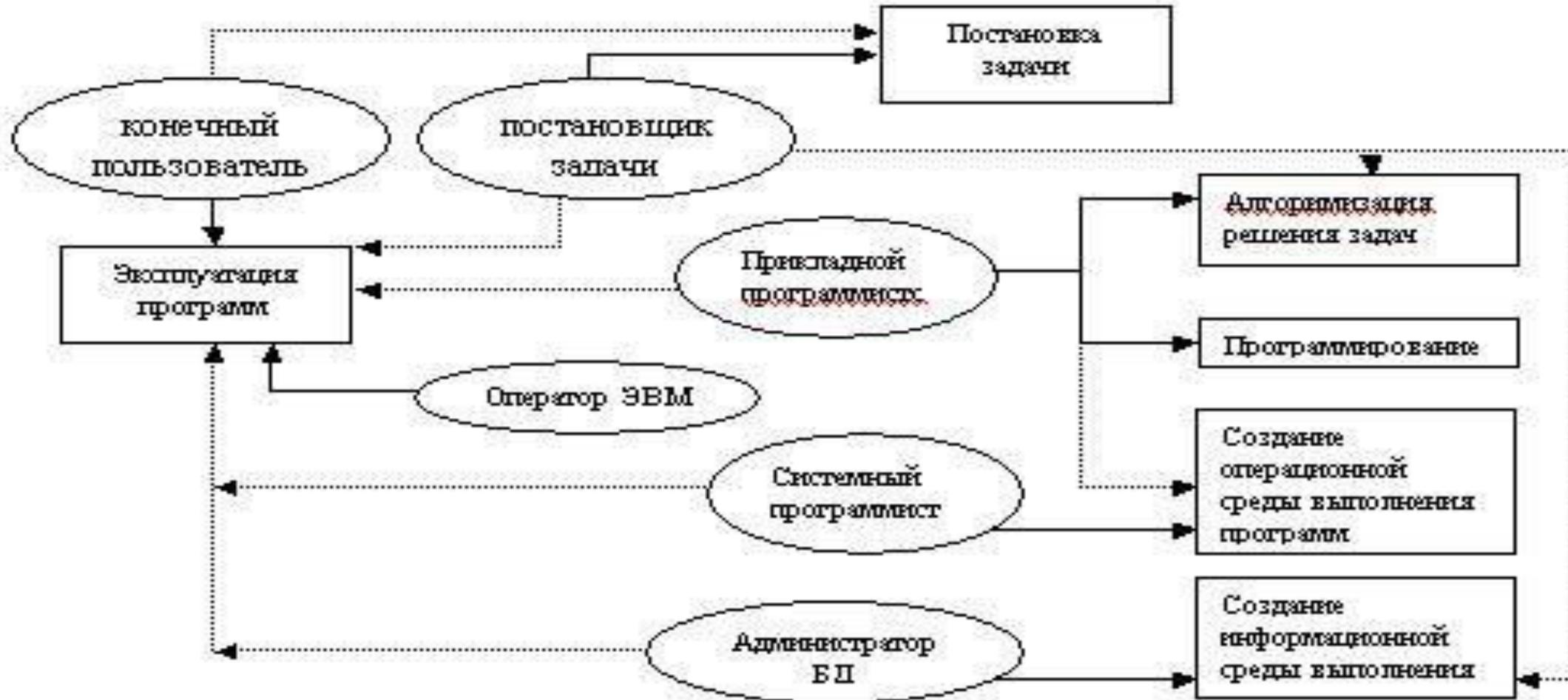
ПРОГРАММИРОВАНИЕ (PROGRAMMING)

- ⦿ Теоретическая и практическая деятельность, связанная с созданием программ.
- ⦿ Программирование является собирательным понятием и может рассматриваться и наука, и как искусство, на этом основан научно-практический подход к разработке программ.
- ⦿ Программа - результат интеллектуального труда, для которого характерно творчество, а оно, как известно, не имеет четких границ.

КАТЕГОРИИ СПЕЦИАЛИСТОВ, ЗАНЯТЫХ РАЗРАБОТКОЙ И ЭКСПЛУАТАЦИЕЙ ПРОГРАММ

- ◎ **Системный программист** (system / software programmer, toolsmith) занимается разработкой, эксплуатацией и сопровождением системного программного обеспечения, поддерживающего работоспособность компьютера и создающего среду для выполнения программ, обеспечивающих реализацию функциональных задач.
- ◎ **Прикладной программист** (application programmer) осуществляет разработку и отладку программ для решения функциональных задач.
- ◎ **Программист-аналитик** (programmer-analyst), который анализирует и проектирует комплекс взаимосвязанных программ для реализации функций предметной области.
- ◎ Большинство информационных систем основано на работе с базами данных (БД). Если база данных является интегрированной, обеспечивающей работу с данными многих приложений, возникает проблема организационной поддержки базы данных, которая выполняется **администратором базы данных**

ВЗАИМОДЕЙСТВИЕ СПЕЦИАЛИСТОВ РАЗЛИЧНОГО ВИДА, УЧАСТВУЮЩИХ В РАЗРАБОТКЕ И ЭКСПЛУАТАЦИИ ПРОГРАММ



ЯЗЫКИ ПРОГРАММИРОВАНИЯ ...

- ⦿ Язык программирования (ВИКИПЕДИЯ) – формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно – ЭВМ) под её управлением.

ХАРАКТЕРИСТИКИ ЯЗЫКОВ

- ⦿ Однозначность при выборе фразы языка
- ⦿ Ограниченный набор слов-команд

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

- ⦿ Язык программирования (ВИКИПЕДИЯ) – формальный язык, предназначенный для записи компьютерных программ.
- ⦿ Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно – ЭВМ) под её управлением.

[HTTPS://WWW.TIOBE.COM/TIOBE-INDEX/](https://www.tio.be.com/tio.be-index/)

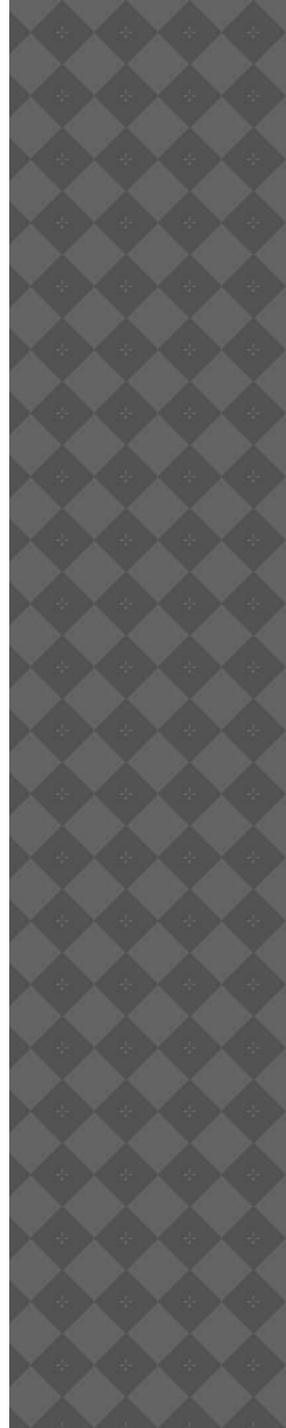
1. Java
2. C
3. Python
4. C++
5. C#
6. Visual Basic .NET
7. JavaScript
8. PHP
9. SQL
10. Objective-C

[HTTPS://GITHUB.COM/](https://github.com/)

1. JavaScript
2. Java
3. Python
4. PHP
5. C++
6. C#
7. TypeScript
8. Shell
9. C
10. Ruby

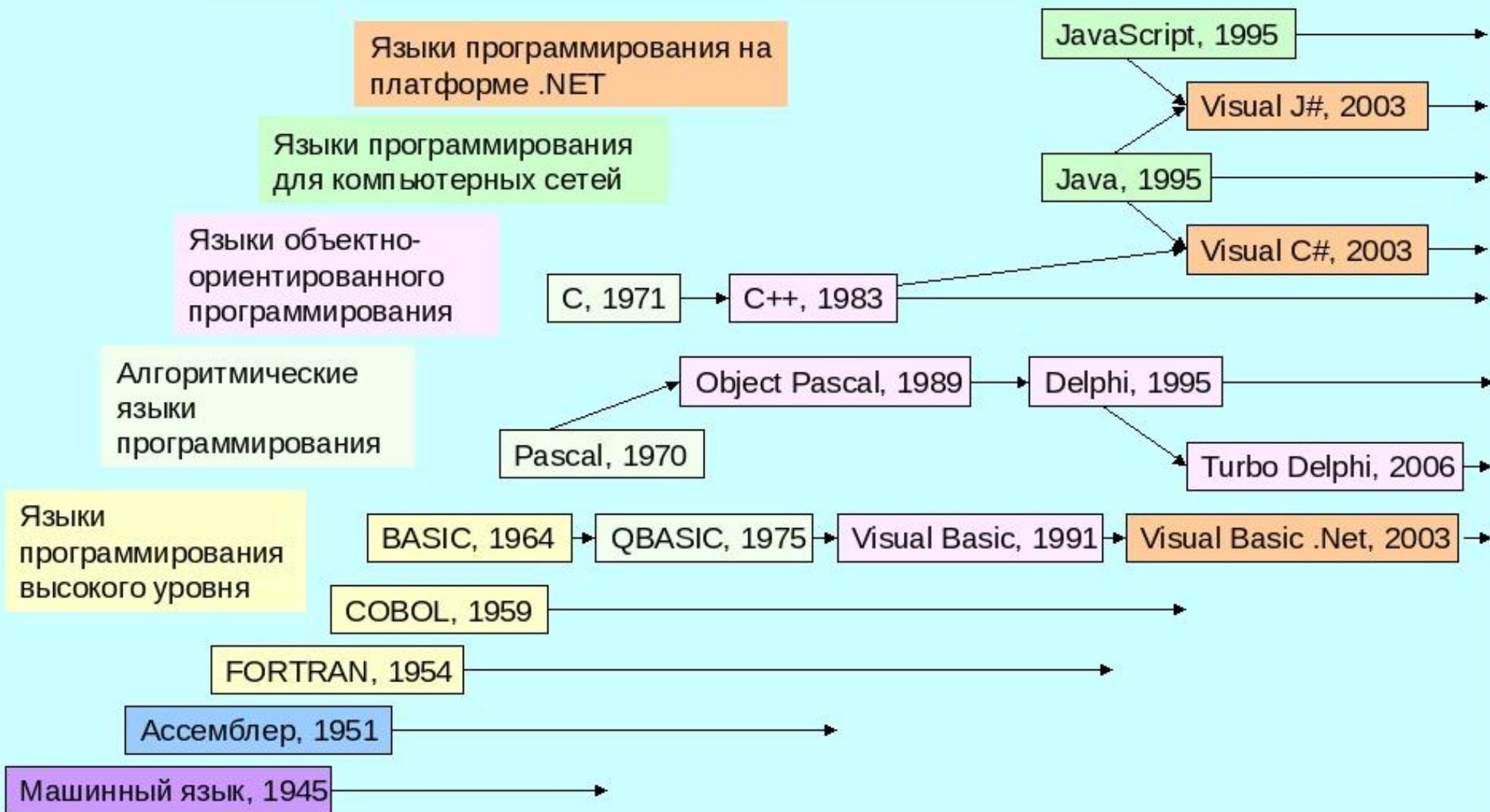
Большое
влияние на
рейтинг GitHub
оказывает
продуктивность
разработчиков

ЭТАПЫ РАЗВИТИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ



ЭТАПЫ РАЗВИТИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Алгоритм, записанный на «понятном» компьютеру языке программирования, называется **программой**.



1. МАШИННЫЙ ЯЗЫК

- **Машинный язык** – это единственный способ взаимодействия с электронно-вычислительными машинами.
 - Каждую команду выполняет ЭВМ
 - Данные записываются в цифровом виде (в 16-й или 2-й системах счисления)
 - Сложен для понимания
 - У каждой ЭВМ свой ЯЗЫК

Программа «Hello, world!» [\[править | править код \]](#)

Программа «Hello, world!» для процессора архитектуры x86 (ОС MS DOS, вывод при помощи BIOS прерывания int 10h) выглядит следующим образом (в шестнадцатеричном представлении):

```
BB 11 01 B9 0D 00 B4 0E 8A 07 43 CD 10 E2 F9 CD 20 48 65 6C 6C 6F 2C 20 57 6F 72  
6C 64 21
```

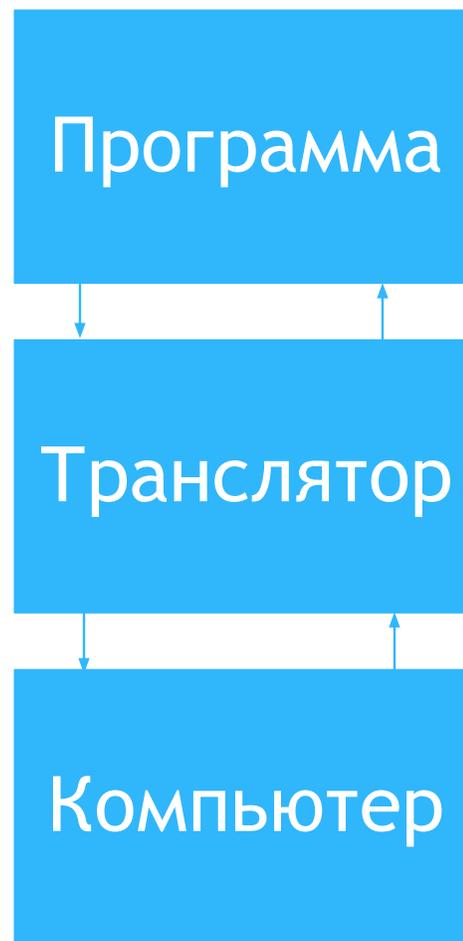
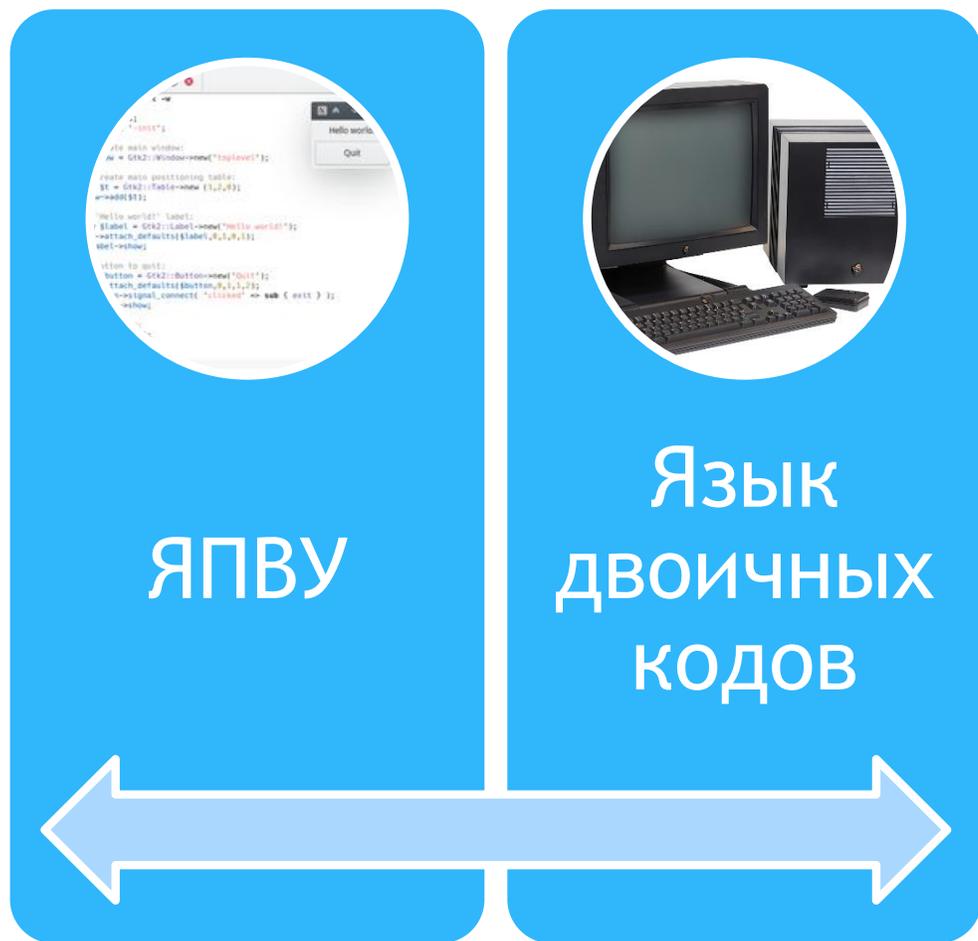
2. АССЕМБЛЕРЫ

- Ассемблер (от англ. assembler — сборщик) — транслятор исходного текста программы, написанной на языке ассемблера, в программу на машинном языке

```
.586 ;Разрешение трансляции всех
;команд (386-486-Pentium)
code segment use16 ;Начало сегмента команд
; 16-разрядное приложение
    assume CS:code, DS:data
main proc ;Начало главной процедуры
    mov AX,data ;Инициализация
    mov DS,AX ;сегментного регистра DS
    . . . ;Текст главной процедуры
    mov AX,4C00h ;Вызов функции DOS
    int 2 In ;завершения программы
main endp ;Конец главной процедуры
code ends ;Конец сегмента команд
data segment use16 ;Начало сегмента данных
    . . . ;Определения данных
data ends ;Конец сегмента данных
stk segment stack ;Начало сегмента стека
    db 256 dup(0) ;Стек
stk ends ;Конец сегмента стека
end main ;Конец программы и точка входа
```

- Специфичны для конкретной архитектуры машины, для ОС и варианта синтаксиса языка
- Вместо чисел используются буквенный команды

ТРАНСЛЯТОРЫ



ТРАНСЛЯЦИЯ ПРОГРАММ

- **Транслятор** — специальная программа, преобразующая программный код с того или иного языка программирования в машинный код
- Ассемблеры и сегодня находят применение, т.к. системные программы (обслуживающие работу аппаратного обеспечения), написанные на ассемблере могут работать быстрее, чем аналогичные программы, написанные на других языках программирования

3. ЯЗЫКИ ВЫСОКОГО УРОВНЯ

- ⦿ **Высокоуровневый (ВИКИПЕДИЯ) — язык программирования, разработанный для быстроты и удобства использования программистом.**
- ⦿ **Основная черта высокоуровневых языков — это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.**

3. ЯЗЫКИ ВЫСОКОГО УРОВНЯ

- ⦿ Программные коды, написанные на языках высокого уровня, обладают логичной структурой. Это облегчает разработку программы и ее отладку.
- ⦿ В отличие от ассемблеров, которые все еще остаются привязанными к своим типам машин, языки высокого уровня обладают **переносимостью**. Т.е., написав один раз программу, программист может выполнить ее на любой машине.

ТРАНСЛЯЦИЯ

Для перевода кода с одного языка программирования (например, высокого уровня) на другой (например, машинный язык) требуется специальная программа — транслятор.

- ⦿ Механизм перевода весьма сложен, выделяют два основных способа трансляции — это компиляция программы или ее интерпретация.
 - При компиляции исходный программный код сразу целиком переводится в машинный. Создается исполняемый файл, который уже никак не связан с исходным кодом. Выполнение исполняемого файла обеспечивается операционной системой самостоятельно.
 - При интерпретации выполнение кода происходит построчно. Интерпретатор, выполняя программу, напрямую взаимодействует с операционной системой.
- ⦿ Выполнение откомпилированной программы происходит быстрее, т.к. она представляет собой готовый машинный код. Однако на современных компьютерах снижение скорости выполнения при интерпретации обычно не заметна.

4. ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ЯЗЫКИ

- Отличие от языков высокого уровня заключается в возможности отстранения от алгоритма выполнения программы.
- С помощью таких языков разработчик как бы оперирует виртуальными объектами.
- На сегодняшний день, реализация больших и сложных проектов осуществляется в основном с помощью ООП

ООП

- ⦿ **Объектно-ориентированное программирование (ООП)** – методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования
- ⦿ Понятие объектно-ориентированного программирования – программный модуль, объединяющий в единое целое данные и программы, манипулирующие данными. Объект характеризуется свойствами, которые являются параметрами объекта и методами, которые позволяют воздействовать на объект и его свойства.

НЕКОТОРЫЕ ОПРЕДЕЛЕНИЯ

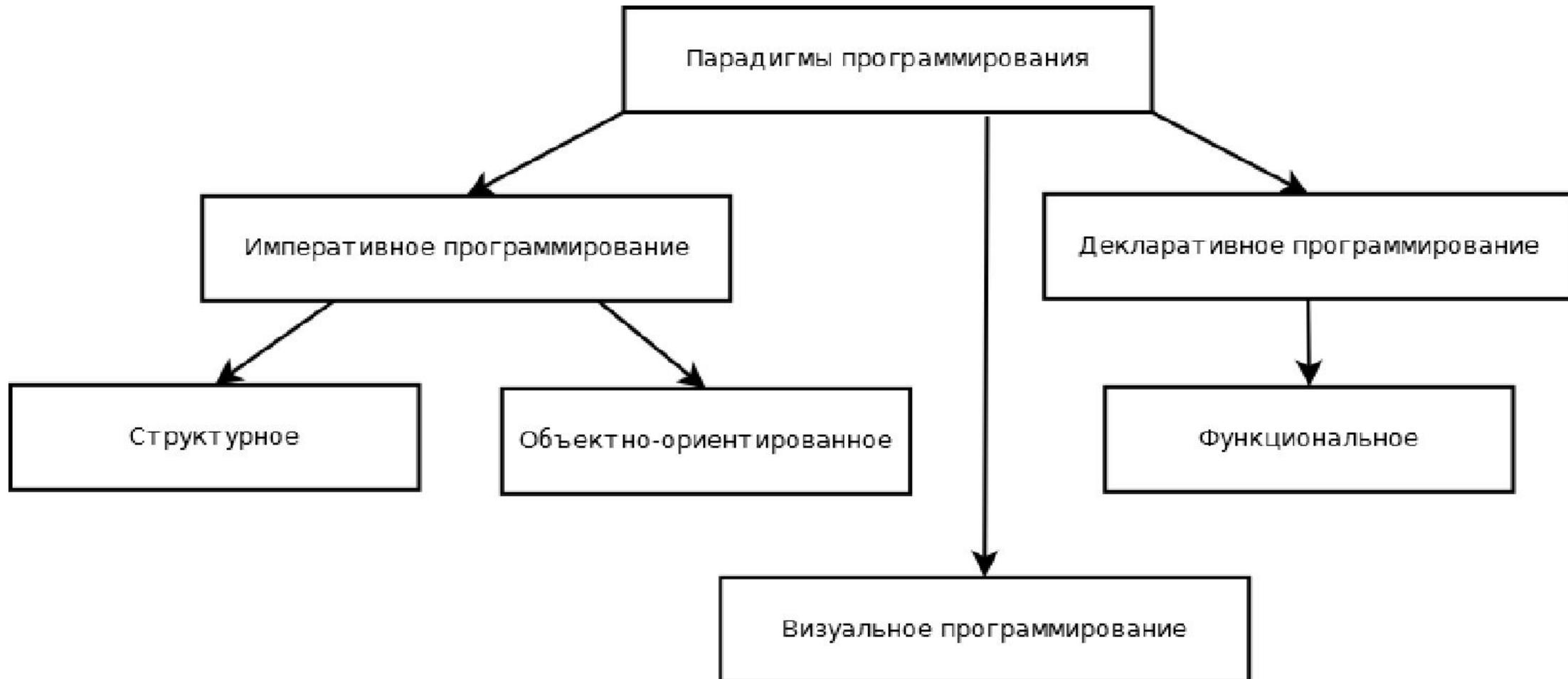
- ⦿ **Метод** - действие в виде процедуры, которое выполняет объект (иногда говорят - выполняется над объектом).
- ⦿ **Идентификатор** – символическое имя объекта, переменной или подпрограммы, которые однозначно идентифицируют их в программе.
- ⦿ **Выражение** – конструкция на языке программирования, предназначенная для выполнения вычислений.
- ⦿ **Операнд** – константа, переменная, функция, выражение или другой объект языка программирования, над которым производятся операции.
- ⦿ **Операция** - конструкция в языках программирования, аналогичная по записи математическим операциям, то есть специальный способ записи некоторых действий
- ⦿ **Присваивание** - операция записи значения в переменную.

ПРИМЕРЫ ЯЗЫКОВ ВЫСОКОГО УРОВНЯ И ООП

- ⦿ C, C++, C#, Delphi, Fortran, Java, JavaScript, Лисп, Паскаль, PHP.
- ⦿ Языкам высокого уровня свойственно
 - умение работать с комплексными структурами данных
 - в большинстве из них интегрирована поддержка строковых типов, объектов, операций файлового ввода-вывода и т. п.

ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

- ◎ **Императивное программирование** - это парадигма программирования, которая описывает процесс вычисления в виде инструкции, изменяющих состояние данных.
 - структурное и объектно-ориентированное
- ◎ **Декларативное программирование** это парадигма программирования, вместо пошагового алгоритма решения задачи задаётся спецификация решения задачи, т.е. описывается, что собой представляет проблема и что требуется получить в качестве результата.
 - функциональное программирование
- ◎ **Визуальное программирование** способ создания программы для ЭВМ путём манипулирования графическими объектами вместо написания её текста
- ◎ **Мультипарадигмальное программирование** - допускают использование различных парадигм



Визуальное программирование способ создания программы для ЭВМ путём манипулирования графическими объектами вместо написания её текста

ДЗ

- ⦿ Эволюция языков высокого уровня и их краткая характеристика (можно с примерами, можно в виде презентации или текста)