

Часть 1 Введение. Алгоритм. Исполнитель. Языки программирования.

Курс «Программирование»

ВШИСиСТ, 1 семестр

Щукин Александр

Валентинович

Введение

- О чем курс
 - Алгоритмизация
 - Программирование
 - Процедурное программирование
 - Структуры данных и алгоритмы
- Связь курса с последующими курсами
- Объем
- Формат занятий
- Формат отчетности

Основные цели курса

- Научиться формализовать решение задачи
- Научиться оценивать и сравнивать алгоритмы
- Научиться понимать принципы работы исполнителя алгоритма, транслятора
- Знать основные характеристики ЯП и парадигм программирования
- Владеть основными синтаксическими конструкциями ЯП и структурами хранения данных
- Понимать принципы структурного и процедурного подхода программирования
- Решать типовые алгоритмические задачи и оценивать их оптимальность
- Владеть базовыми инструментами языка программирования высокого уровня

Программное обеспечение

- Компьютер = ПО (software) + АО (hardware)
- Программное обеспечение = Программа ?
- Какое бывает программное обеспечение
 - Информационная система
 - Компонент, библиотека
 - Сервис
 - Утилита
 - ...
- Откуда появляется программное обеспечение?
- Программное обеспечение
 - Продукт?
 - Система?
 - Алгоритм?

Алгоритм и программа

- Алгоритм
- Программа
- Алгоритм → Алгоритмизация
- Программа → Программирование



Алгоритм

- **Алгоритм** — это точный набор инструкций, описывающих последовательность действий исполнителя для достижения результата решения задачи. (Википедия)
- **Алгоритм** – совокупность четко определенных правил, процедур или команд, обеспечивающих решение поставленной задачи за конечное число шагов. (сайт об интернет-бизнесе)
- **Алгоритм** – последовательность выполнения той или иной работы или действия. (словарь терминов психологии и педагогики)
- **Алгоритм** - однозначное пошаговое описание (предписание, инструкция, правило) чисто механически (в отвлечении от содержательного контроля) выполняемого шаг за шагом единообразного и опирающегося на конечное множество правил решения любой конкретной задачи из какого-либо класса задач данного определённого типа. (сайт кафедры филологического образования Московского института Открытого образования)
- **Алгоритм**—точное предписание о выполнении в определенном порядке некоторой системы действий, переводящей исходные данные в искомый результат. Используется для решения задач с помощью ЭВМ. (база данных "Ветеринарное законодательство")

И еще одно определение алгоритма

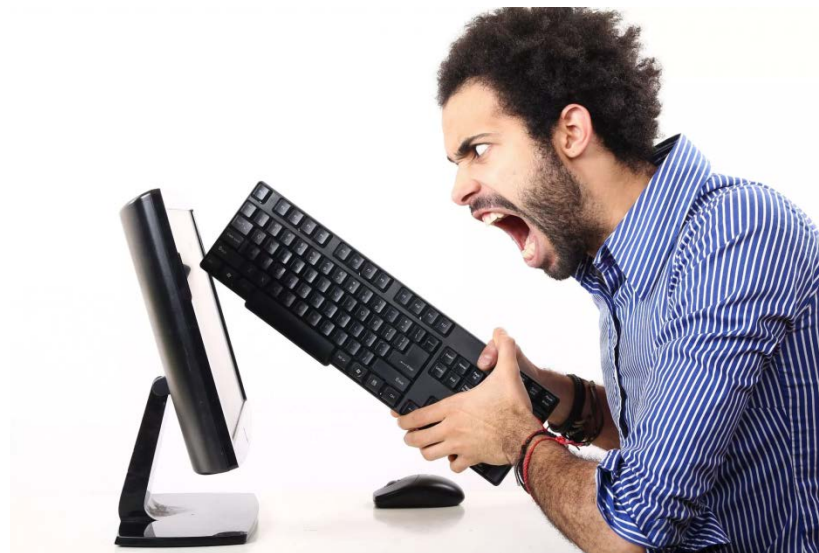
- Алгоритм – пронумерованный список однозначных и понятных для исполнителя указаний (инструкций) направленных на решение задачи за конечное число шагов
- Нам нужны алгоритмы для:
 - Формирования способа решения утилитарных задач программирования (макросы)
 - Формирования способа решения задачи проектирования информационной системы (ИС) и подготовки пакета документов для разработчиков
 - Формирование способа решения любой сложной задачи и ее формализация

Основные свойства алгоритмов

- Дискретность (прерывность, раздельность)
 - Алгоритм представлен как пронумерованный список автономных, самодостаточных команд. Дискретность в частности позволяет выполнять пошаговое исполнение или адресоваться к нужной команде
- Определенность
 - Каждая инструкция в алгоритме однозначно интерпретируется исполнителем, без помощи автора
- Результативность (конечность)
 - Результат алгоритма (в том числе и отрицательный) может быть достигнут за конечное число шагов
- Массовость
 - Алгоритм может быть применен не к конкретной ситуации с определенными данными, а к множеству однотипных ситуаций, описывающих одну задачу

Алгоритм позволяет

- Справиться со сложными задачами
- Получить устойчивый и управляемый код
- Эффективно работать в команде
- Эффективно выполнять модификацию кода



Исполнитель

- Кем выполняются программы?
- Что мы будем понимать под термином Исполнитель?
- Универсальный исполнитель
 - Внутренняя память
 - Система внешних команд
 - Система внутренних команд (способность управлять другими исполнителями)

ЯЗЫКИ

- Что мы будем понимать под термином «ЯЗЫК»
- Формальные языки
 - Примеры...
- Язык программирования – формальная знаковая система, предназначенная для записи компьютерных программ

Язык программирования

- Алфавит (лексема)
 - Синтаксис
 - Семантика
 - Прагматика
-
- Говорят о лексическом анализе, синтаксическом, семантическом.

Лексема

- Слово, как абстрактная единица естественного языка (лингвистика)
- Последовательность допустимых символов языка программирования, имеющая смысл для транслятора (информатика)

Виды анализа текста программы

- Лексический анализ: процесс анализа и преобразования потока входных символов в токены, «атомы» текста программы на метауровне.
- Синтаксический анализ (парсинг): в лингвистике и информатике — процесс сопоставления линейной последовательности лексем (слов, токенов) естественного или формального языка с его формальной грамматикой. Результатом обычно является дерево, описывающее синтаксическую структуру входной последовательности.

Способы определения языков

- Перечислить все множество допустимых, корректных последовательностей лексем
- Задать правила для генерации корректных последовательностей лексем
- Задать правила для проверки на корректность последовательностей лексем

Языки программирования

- Классификация по уровню абстракции
 - Машинные коды
 - Язык программирования низкого уровня (ассемблеры)
 - Язык программирования высокого уровня
 - CASE системы

Пример программы на машинном языке

```
        x = 10
BB 0A 00 00 00
        y = x / 4
89 5D C8
DB 45 C8
DC 35 E8 FA 76 03
D9 FC
DD 5D CC
FF 75 D0
FF 75 CC
E8 74 22 A4 75
8B F8
        Console.WriteLine(y)
B9 14 81 D2 02
E8 4F 25 07 FD
8B F0
```

Пример программы на ассемблере

```
        x = 10
mov     ebx, 0Ah
        y = x / 4
mov     dword ptr [ebp-38h], ebx
fild    dword ptr [ebp-38h]
fdiv    qword ptr ds:[0376FAE8h]
frndint
fstp    qword ptr [ebp-34h]
push    dword ptr [ebp-30h]
push    dword ptr [ebp-34h]
call    75A422B9
mov     edi, eax
        Console.WriteLine(y)
mov     ecx, 2D28114h
call    FD0725A0
mov     esi, eax
```

Пример программы языке высокого уровня

```
X = 10  
Y = X / 4  
Console.WriteLine(Y)
```

Язык программирования

- Зачем так много языков программирования?
- Язык программирования определяет:
 - Возможности программиста
 - Время и усилия по разработке программного обеспечения
 - Способ организации данных и методы их обработки
 - Стил программирования, способ организации кода

ЯВУ vs Машинные языки

- Алфавит шире
- Набор операций не зависит от аппаратной реализации
- Формат предложений гибкий и удобный
- общепринятые математические обозначения
- Идентификаторы для данных, задаваемые программистом
- Широкий набор типов данных

Классификация ЯП по парадигме

- Неструктурные (BASIC...)
- Структурные (Fortran...)
- Процедурные (C, Pascal...)
- Объектно-ориентированные (C++, VB, C#...)
- Интерпретируемые (JavaScript, VBA...)
- Функциональные (ЛИСП, Haskell, OCaml...)
- Логические (Пролог...)
- ...

Тезис Черча-Тьюринга

- Знаменитый тезис Чёрча утверждает, что все виды алгоритмов эквивалентны друг другу, т.е. классы задач, решаемых разными типами алгоритмов, всегда совпадают. Тезис этот недоказуем (можно лишь доказать совпадение для двух конкретных типов алгоритмов, например, машин Тьюринга и нормальных алгоритмов), но никто в его верности не сомневается. Так что все языки программирования эквивалентны друг другу и различаются лишь тем, насколько они удобны для решения конкретных классов задач.
- Или: всякая вычислимая в интуитивном смысле функция вычислима с помощью некоторой машины Тьюринга