

# Informatics basics

Starovoytov Alexandr, Grechko Georgy

*<2021-09-04 Sat 13:50>*

## Contents

|  |          |
|--|----------|
| <b>1 лекция 1</b>                          | <b>2</b> |
| 1.1 Основные понятия информатики . . . . . | 2        |
| 1.1.1 Данные . . . . .                     | 2        |
| 1.1.2 Алгоритм . . . . .                   | 2        |
| 1.1.3 Свойства алгоритма: . . . . .        | 2        |
| 1.1.4 Компьютерная программа . . . . .     | 2        |
| 1.1.5 Язык программирования . . . . .      | 2        |
| 1.1.6 Компьютер . . . . .                  | 3        |
| 1.1.7 Подпрограмма . . . . .               | 3        |
| 1.1.8 Сопрограмма . . . . .                | 3        |
| 1.2 Парадигмы программирования . . . . .   | 3        |
| 1.2.1 Основные группы парадигм . . . . .   | 3        |
| 1.2.2 примеры . . . . .                    | 4        |
| 1.3 язык Scheme . . . . .                  | 4        |
| 1.3.1 информация о языке . . . . .         | 4        |

- Лекция в pdf формате
- Преподаватель
  - Коновалов Александр Владимирович
  - 89175404352
  - akonovalov@bmstu.ru
  - a.v.konovalov87@mail.ru

# **1 лекция 1**

## **1.1 Основные понятия информатики**

### **1.1.1 Данные**

это представление фактов, понятий, инструкций в форме, приемлемой для обмена, интерпретации или обработки человеком или с помощью автоматических средств

### **1.1.2 Алгоритм**

это конечная совокупность точно заданных правил решения произвольного класса задач или набор инструкций, описывающий порядок действий исполнителя для решения некоторой задачи

### **1.1.3 Свойства алгоритма:**

1. Дискретность - делится на отдельные элементарные части, отвечающие за определенные действия.
2. Детерминированность - на одних и тех же входных данных - один и тот же результат.
3. Понятность - элементы алгоритма должны быть понятны исполнителю.
4. Завершаемость - если не завершается - то это вычислительный процесс.
5. Массовость - алгоритм пригоден для решения всех задач данного типа.
6. Результативность - указывает на наличие таких исходных данных, для которых реализуемый по заданному алгоритму вычислительный процесс должен через конечное число шагов остановиться и выдать искомый результат.

### **1.1.4 Компьютерная программа**

это алгоритм, записанный на некотором языке программирования

### **1.1.5 Язык программирования**

формальный язык, предназначенный для записи компьютерных программ

### **1.1.6 Компьютер**

программно управляемое устройство для обработки информации

### **1.1.7 Подпрограмма**

некоторый именованный блок кода.

Вызывающая программа приостанавливается, управление передается подпрограмме, по завершению управления передается обратно

### **1.1.8 Сопрограмма**

в отличие от подпрограммы работает поочередно с вызывающей программой.

При повторном вызове возобновит выполнение с момента, где остановилась

## **1.2 Парадигмы программирования**

совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию). Это способ концептуализации, определяющий организацию вычислений и структурирование работы, выполняемой компьютером.

### **1.2.1 Основные группы парадигм**

1. Императивные способ записи программ, в котором указывается последовательность действий  
основной признак - оператор присваивания (меняющий значение переменной)
  - (a) Структурное программирование программа является композицией блоков с одним входом и одним выходом  
(есть операторы ветвления и цикла, но нет goto)
  - (b) Процедурное программирование совокупность подпрограмм, где одни подпрограммы вызывают другие
  - (c) Объектно-ориентированное программа рассматривается как набор некоторых взаимодействующих объектов.  
Объект сочетает в себе данные и методы их обработки, методы вызываются в ответ на сообщение:  
посыпаем сообщение объекту -> вызывает метод -> возвращает ответ  
объекты объединяются в классы

2. Декларативные способ записи программ, в котором описывается взаимосвязь между данными, описывается цель, но не последовательность шагов ее вычисления
  - (a) Функциональное алгоритм записывается как набор взаимосвязанных функций, функции рассматриваются с математической точки зрения, описывает взаимосвязь между данными и результатом
 
$$x = f(y) + g(z);$$
  - (b) Логическое (Prolog + отчасти SQL) алгоритм описывает взаимосвязь между понятиями.  
Выполнение программы сводится к выполнению запросов
3. Метапрограммирование метаирония. Программа рассматривается как данные/объект для другой программы
  - (a) Программу пишут программы Например: макросы, генераторы кода, шаблоны C++ (фигня нечитаемая зачем он постоянно в руках часы крутит попит симпл димпл кручे нет попит симпл димпл попит симпл димпл маленький красивый попит большой и милый)
  - (b) Программа взаимодействует с вычислительной средой Рефлексия или интроспекция - самоанализ.  
Например, посмотреть поля и значения в объекте класса

### 1.2.2 примеры

```
ИП: sort(array);
ФП: sorted_list = sort(list)
      sorted_list - константа, т.к. нет присваивания
ЛП: sort(unsorted, sorted)
```

## 1.3 язык Scheme

### 1.3.1 информация о языке

Lisp 1950-е годы Джон МакКарти  
 LISt Processing  
 Scheme 1970-е годы Абельсон и Сассман  
 изучаем R5RS  
 современная редакция R7RS

- другие языки семейства LISP:

- Common Lisp
- Clojure
- Racket