

# **Алгоритмы и его свойства**

Введение в программирование

# Алгоритм

---

Появление алгоритмов связывают с зарождением математики. Более 1000 лет назад (в 825 году) ученый из города Хорезма Абдулла (или Абу Джафар) Мухаммед бен Муса аль-Хорезми создал книгу по математике, в которой описал способы выполнения арифметических действий над многозначными числами. Само слово алгоритм возникло в Европе после перевода на латынь книги этого математика.

**Алгоритм** – это строго детерминированная последовательность действий, описывающая процесс преобразования объекта из начального состояния в конечное, записанная с помощью понятных исполнителю команд.

# Алгоритм приготовления чая

---

- Подготовить исходные величины – чай, воду, чайник, ложку.
- Налить в чайник воду.
- Поставить чайник на огонь.
- Довести до кипения и снять с огня.
- Всыпать в чайник 2 ложки чая.
- Поставить чайник на огонь.
- Довести воду до кипения (но не кипятить), снять с огня.
- Процесс прекратить.

# Задание

---

- Составить алгоритм оценки знаний по результатам теста из 10 вопросов, согласно следующему правилу:

*если правильные ответы на 10 вопросов, то выставляется оценка 5, при 9 и 8 правильных ответах – оценка 4, при правильных ответах не более чем на 5 вопросов - оценка 2, в остальных случаях – оценка 3.*

# 1) Формулировка задачи в математическом виде:

---

$$\text{Оценка} = \begin{cases} 5, & \text{если } \text{ot} = 10 \\ 4, & \text{если } 8 \leq \text{ot} \leq 9 \\ 3, & \text{если } 6 \leq \text{ot} \leq 7 \\ 2, & \text{если } \text{ot} \leq 5 \end{cases}$$

где оценка - оценка, ot - правильные ответы

## 2) Алгоритм "Оценка"

---

- Ввод значения  $ot$ .
- Если  $ot = 10$ , то  $осенка = 5$ ,
- иначе (проверить условие)
- Если  $ot \geq 8$ , то  $осенка = 4$ ,
- иначе (проверить условие)
- Если  $ot \geq 6$ , то  $осенка = 3$ ,
- иначе
- $осенка = 2$ ,
- Вывести значение  $осенка$ .

# Исполнители алгоритмов

---

Объект, выполняющий инструкции, последовательности команд, предписания алгоритма **исполнителем**.

- **Исполнитель** - объект, который выполняет алгоритм.

Исполнителем может быть человек, машина, робот, компьютер...

- **Формальный исполнитель** – исполнитель выполняющий команды определенного алгоритма без анализа действий и ситуации.
- **Компьютер** – формальный исполнитель алгоритмов.

# Основные характеристики исполнителя:

---

- *Система команд* – совокупность команд понятных исполнителю;
- *Программная среда* – разбита на 3 области: поле программы, поле рисунка, строка меню;
- *Отказы.*

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



# ПРИМЕР

В систему команд исполнителя Удвоитель входят следующие команды: умножь на 2, прибавь 1.

---

а) Составить для исполнителя Удвоитель алгоритм получения числа 100 из числа 0.

Решение:

- прибавь 1
- умножь на 2
- прибавь 1
- умножь на 2
- умножь на 2
- умножь на 2
- прибавь 1
- умножь на 2
- умножь на 2

# Задание

Какое число получит Удвоитель из числа 0, исполнив следующий алгоритм:

- прибавь 1
- прибавь 1
- умножь на 2
- прибавь 1
- умножь на 2
- умножь на 2
- прибавь 1
- умножь на 2
- умножь на 2

Ответ: \_\_\_\_\_

# Свойства алгоритмов:

---

1. **Дискретность** (разбиение алгоритма на последовательность отдельных законченных действий).
2. **Выполнимость и понятность команд** (алгоритм должен содержать команды, входящие в систему команд исполнителя и записанные на понятном для исполнителя языке).
3. **Детерминированность (однозначность)** (строгое и недвусмысленное определение любого действия в каждом случае).
4. **Массовость** (возможность использования алгоритма с разными исходными данными).
5. **Результативность** (возможность получения из исходных данных результата за конечное число шагов).
6. **Конечность.** (завершение работы алгоритма в целом за конечное число шагов).

<b>Свойства</b>	<b>Пример выполнения свойства</b>	<b>Пример невыполнения свойства</b>
<i><b>Дискретность</b></i>	Казнить нельзя, помиловать!	Казнить, нельзя помиловать
<i><b>Однозначность</b></i>	На дубе ларец, а в ларце утка, а в утке яйцо, в яйце игла, в игле смерть Кощея.	Поди туда, не знаю куда, принеси то, не знаю что.
<i><b>Массовость</b></i>	Каждой дочери отец привёз по дорогому подарку	Принц мог жениться только на настоящей принцессе
<i><b>Понятность</b></i>	Инструкция по-русски	Инструкция на японском языке
<i><b>Конечность</b></i>	Мама сварила кашу в горшочке	Каша уже заполнила все улицы, а горшочек всё варил кашу.
<i><b>Результативность</b></i>	Мышка хвостиком махнула, яйцо и разбилось	Баба била-била не разбила.

# Виды алгоритмов

---

1. **Линейный алгоритм** (описание действий, которые выполняются однократно в заданном порядке);
2. **Циклический алгоритм** (описание действий, которые должны повторяться указанное число раз или пока не выполнено задание);
3. **Разветвляющийся алгоритм** (алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий)
4. **Вспомогательный алгоритм** (алгоритм, который можно использовать в других алгоритмах, указав только его имя).

# Представления алгоритма

---

- Словесный, письменный на естественном языке;
- Графический (**блок-схема**, которая составляется из стандартных графических объектов);
- С помощью алгоритмического языка.

# ПРИМЕР


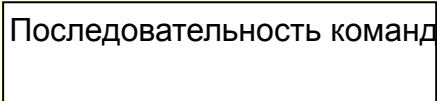
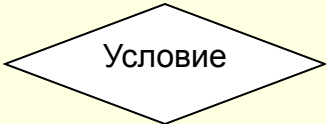
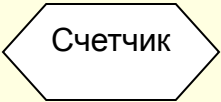
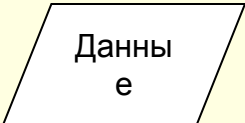
Составить алгоритм поиска площади круга  
радиусом  $R$

---

## *а) Словесно-пошаговый.*

- 1. Ввод значения  $r$
- 2. Вычислить  $s = \pi \cdot r^2$
- 3. Записать в ответ значение  $s$ .
- 4. Конец.

## *б) Графический (в виде блок-схемы)*

Вид графического объекта	Назначение
	<i>Начало и конец алгоритма</i>
	<i>Функциональный блок - выполняемое действие</i>
	<i>Блок условия - условие выполнения действий</i>
	<i>Счетчик кол-во повторов</i>
	<i>Блок ввода-вывода данных</i>
	<i>Последовательность выполнения действий</i>



**начало**



**Ввод r**



**$S := 3.14 * r^2$**



**Вывод s**



**конец**

## *в) Алгоритмический язык*

алг ЗАДАЧА (вещ  $r, s$ )

- арг  $r$
- рез  $s$
- нач
- ВВОД  $r$
- $s := 3.14 * r^2$
- ВЫВОД  $s$
- КОН

# Команда присваивания

---

- При записи вычислительных алгоритмов удобно использовать специальный **знак присваивания** : =
- $Y := X$ . (Читается: "Y присвоить X").
- ПРИМЕР:
- $Y := K$ ;  $Y := 37$ ;  $X := X + 1$  ("возьми то, что хранится в X, добавь 1 и результат опять положи в X")
- $V := \text{"Ура! Скоро каникулы!"}$ , V – является символьной или литерной переменной, ее значением является текст.

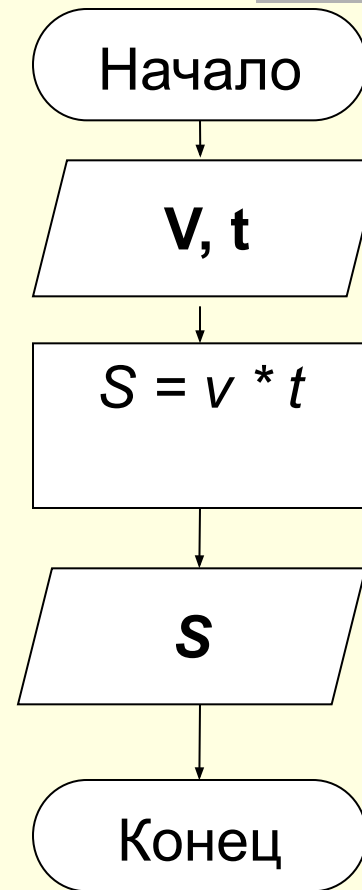
# Пример записи алгоритма в виде блок-схемы

**Задача:** определить расстояние, пройденное человеком, если известно время движения, а так же известно, что движение было равномерным.

**Условие задачи:** заданы скорость движения и время. Нужно найти путь.

**Математическая модель:**

$$S = v * t$$



# Практические задания

- Определить значение переменной  $A$  в результате исполнения последовательности команд присваивания:  $A := 7$   $B := A$   $A := 3 * A - B$
- Составить алгоритм действий Удвоителя для получения числа 8 из числа 0, используя не более четырех шагов.
- Записать в виде команды присваивания:
  - а) Увеличить значение переменной  $C$  на 2;
  - б) Уменьшить значение переменной  $M$  в три раза;
  - в) Занести в память ЭВМ свою фамилию, имя, год рождения, используя три команды присваивания.

# Вопросы:

---

- Что такое алгоритм? Приведите примеры алгоритмов.
- Какие свойства алгоритмов вы знаете?
- Какие виды алгоритмов вы знаете?
- Какие способы записи алгоритмов вы знаете?
- Что такое исполнитель алгоритмов?
- Что такое программа?

# Домашнее задание

---

## § 4.1.1;

### Задачи:

1. Представить схематически алгоритм решения задачи по нахождению периметра прямоугольника.
2. Составить на алгоритмическом языке программу нахождения минимального из трех значений переменных  $x$ ,  $y$ ,  $z$ .
3. Составить алгоритм действий Удвоителя для получения числа 8 из числа 0, используя не более четырех шагов.