



Основы алгоритмизации и программирования.

Введение

Педагогическое образование, 3 семестр

Mayer Svetlana Fyodorovna

Ресурсы и расписание

- **Internet ресурсы**

- Среда Moodle: edu.mmcs.sfedu.ru

- Moodle курс:

- <http://edu.mmcs.sfedu.ru/course/view.php?id=575>

- Сайт для занятий: <https://labs-org.ru>

- Официальный сайт языка PascalABC.NET

- <http://pascalabc.net>

Programming as a human activity

- Программирование - это часть человеческой деятельности
- В повседневной жизни мы совершаем действия, принимаем решения, используя некоторые алгоритмы
- Многие задачи сегодня требуют автоматизации в виде компьютерного программирования
- Количество таких задач растет
- Каждый год программирование охватывает новые области человеческой жизни. Среди них сегодня: машинное обучение, распознавание изображений, программирование роботов и т.д.
- Нам нужно научиться писать компьютерные программы, используя какой-то язык программирования и изучая некоторые стандартные компьютерные алгоритмы

Algorithms

- Основное понятие в области ИТ - **Алгоритм**
- Алгоритм – это последовательность инструкций (шагов) для решения какой-либо задачи.

Пример. Как зарегистрироваться в системе moodle?

- * Шаг 1. Запустите браузер и наберите адрес **edu.mmcs.sfedu.ru**
- * Шаг 2. Щелкните по кнопке «Зарегистрироваться» (Register)
- * Шаг 3. Введите Ваш электронный адрес (**@sfedu**), имя и пароль
- * Шаг 4. Откройте электронную почту и подтвердите регистрацию
- В нашей жизни нас окружает множество алгоритмов
- Алгоритм в информатике предназначен для решения задач с помощью компьютера
- Наша задача научиться создавать алгоритмы и писать программы


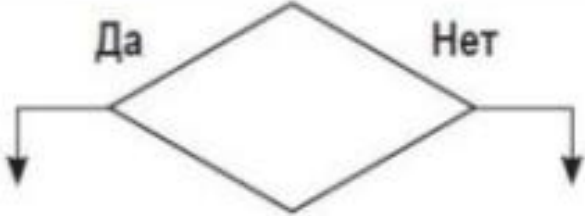
Основными свойствами алгоритма являются

- 1. Детерминированность** (определенность). Предполагает получение однозначного результата вычислительного процесса при заданных исходных данных. Благодаря этому свойству процесс выполнения алгоритма носит механический характер.
- 2. 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.
- 3. Массовость.** Это свойство предполагает, что алгоритм должен быть пригоден для решения всех задач данного типа.
- 4. Дискретность.** Означает расчлененность определяемого алгоритмом вычислительного процесса на отдельные этапы, возможность выполнения которых исполнителем (компьютером) не вызывает сомнений.

Блок-схема

- **Блок-схемой** называется графическое изображение логической структуры алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций.
- При всем многообразии алгоритмов решения задач в них можно выделить три основных вида вычислительных процессов:
 - **линейный;**
 - **ветвящийся;**
 - **циклический.**
- **Линейным** называется такой вычислительный процесс, при котором все этапы решения задачи выполняются в естественном порядке следования записи этих этапов.
- **Ветвящимся** называется такой вычислительный процесс, в котором выбор направления обработки информации зависит от исходных или промежуточных данных (от результатов проверки выполнения какого-либо логического условия).
- **Циклом** называется многократно повторяемый участок вычислений. Вычислительный процесс, содержащий один или несколько циклов, называется циклическим.

Блок-схема

Внешний вид блока	Назначение блока
	<p>Функциональный блок (<i>операторный блок, процесс</i>).</p> <p>Служит для указания действия (шага) алгоритма. В прямоугольник входит одна направленная линия, из него выходит одна направленная линия. Внутри прямоугольника записывается команда, которая должна быть выполнена. Можно записывать несколько команд в одном блоке (для наглядности).</p>
	<p>Альтернативный блок (<i>условный блок, условие</i>).</p> <p>Служит для указания выбора одного из двух возможных действий. Внутри ромба размещается условие выбора (например, вопрос или сравнение). Условием может быть выражение, для которого возможно только одно из двух значений — либо «истина», либо «ложь».</p> <p>В ромб входит одна направленная линия. Из ромба выходят две направленные линии, каждая из которых подписана «Да» или «Нет». Если условие, записанное внутри ромба, будет верным (значение «истина», «да»), то управление будет передано по стрелке «Да».</p>

Блок-схема

	<p>Блок начала / конца алгоритма (пуск/останов). Используется в начале и конце блок-схемы алгоритма</p>
	<p>Блок ввода/вывода. Служит для организации ввода исходных данных и вывода результирующих данных.</p>
	<p>Блок цикла. Предназначен для организации циклического процесса с параметром. Количество повторений (<i>итераций</i>) цикла и шаг изменения параметра должны быть известны. Внутри блока указываются (через запятую) начальное значение параметра цикла, конечное значение и шаг его изменения.</p>
	<p>Блок predefined process (subprogram). Используется для указания обращения к отдельным модулям, вспомогательным алгоритмам, библиотечным подпрограммам.</p>
	<p>Блок печати Указывает вывод результатов на печать.</p>

Основные определения

- **Алгоритм** – набор команд для решения задачи
- **Программа** – алгоритм, написанный на каком-либо языке программирования
- Шаги алгоритма называются **командами**
- Наиболее простые шаги можно назвать **выражением** (или оператором)

Некоторые языки программирования:

Java	JavaScript	Scratch
Kotlin	Go	Perl
C	Swift	Assembler
C++	Delphi Pascal	PascalABC.NET
Python	Basic	
C#	Haskell	

Мы будем использовать Pascal и PascalABC.NET.

Почему?

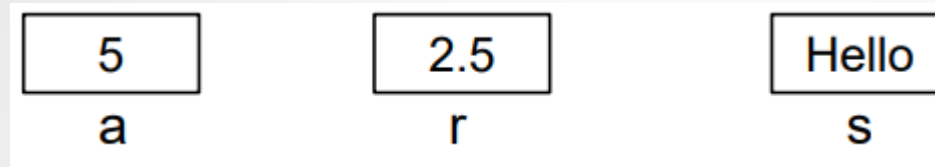
- **Pascal :**
- Пока наиболее популярен в школах России
- Дает возможность использовать более продвинутый вариант языка - PascalABC.NET
- **PascalABC.NET :**
 - Разработан в нашем университете
 - Ориентирован на современное программирование
 - Допускает ясное и компактное написание кода
 - Включает традиционный Pascal, Delphi Pascal, расширение .NET (например, C#)



К программированию, господа!

Переменные

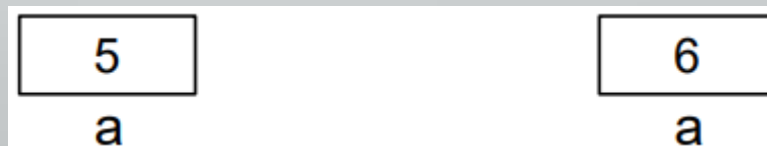
- **Переменная** – ячейка памяти, у которой есть имя, тип и значение



- **Тип переменной** определяется набор значений, которые может принимать переменная
- **Основные типы:**

```
integer    5 -3  2019
real       2.5  3.33
string     'Hello' 'Student' '2019'
```

- Каждая переменная одновременно может хранить единственное значение:



Присваивание значений переменной

```
a := 5;
```

Присваивание

В памяти:

```
5  
a
```

Сохраненное значение

Ячейка памяти хранит одно значение! Предыдущее стирается!

```
a := 6;
```

```
6  
a
```

Переменные в выражениях

- Как вычисляются выражения с переменными?

```
x := 3;  
y := 4;  
a := x + y;
```

- **Шаг 1.** Если выражение содержит переменные, то переменные заменяются их значениями:
- **Шаг 2.** Затем вычисляется выражение
- **Шаг 3.** Результат присваивается переменной

```
a := 3 + 4;
```

Первая программа PascalABC.NET

```
var x,y: integer; }  
var a: integer;
```

```
begin  
  x := 3;  
  y := 4;  
  a := x + y;  
  Write(a);  
end.
```

Раздел
описаний

Раздел
операторов

Write – оператор,
выводит сообщения или
результаты в окно
вывода (консоль)

7

Окно
вывода(консоль)

Как извлекается программа?

Щелкните:  на панели инструментов

- Запустится специальная программа – **компилятор** pascalabc.net
- Компилятор проверит программу, и, если в ней нет ошибок, сгенерирует **машинный код**
- Затем машинный код обрабатывается центральным процессором

```
var x,y: integer;  
var a: integer;  
  
begin  
  x := 3;  
  y := 4;  
  a := x + y;  
  Write(a);  
end.
```

7



компилятор

```
5:      x := 3;  
mov dword [rip-0x11d212], 0x3  
6:      y := 4;  
mov dword [rip-0x11d218], 0x4  
7:      a := x + y;  
mov ecx, [rip-0x11d222]  
add ecx, [rip-0x11d224]  
mov [rip-0x11d226], ecx  
8:      Write(a);  
mov rcx, 0x7fff94d9c158  
call 0x5f5e0aa0  
mov [rbp-0x10], rax  
mov rcx, [rbp-0x10]  
mov eax, [rip-0x11d243]  
mov [rcx+0x8], eax  
mov rcx, [rbp-0x10]  
call 0xffffffffffffe758
```

Машинные коды



- https://labs-org.ru/pascal-1_2/

Улучшение программы(1)

```
var x,y: integer;
```

```
var a: integer;
```

```
begin
```

```
  x := 3;
```

```
  y := 4;
```

```
  a := x + y;
```

```
  Write(x,y,a);
```

```
end.
```

347

Вывод некрасивый: нет
пробелов между элементами

Улучшение программы(2)

```
var x,y: integer;
```

```
var a: integer;
```

```
begin
```

```
  x := 3;
```

```
  y := 4;
```

```
  a := x + y;
```

```
  print(x,y,a);
```

```
end.
```

```
3 4 7
```

Используйте **Print** вместо **Write**, чтобы выводить элементы через пробел

Улучшение программы(3)

```
var x,y: integer;  
var a: integer;  
  
begin  
  x := 3;  
  y := 4;  
  a := x + y;  
  print('${x} + {y} = {a}');  
end.
```

Все выражения и переменные внутри фигурных скобок заменяются на их значения

3 + 4 = 7

Улучшение программы(4)

```
begin ↓  
  var x,y: integer;  
  var a: integer;  
  x := 3;  
  y := 4;  
  a := x + y;  
  print('${x} + {y} = {a}');  
end.
```

Можно перенести объявление переменных в раздел операторов

3 + 4 = 7



Q & A