

Программирование на языке Паскаль

- § 54. Алгоритм и его свойства
- § 55. Простейшие программы
- § 56. Вычисления

Программирование на языке Паскаль

§ 54. Алгоритм и его свойства

Что такое алгоритм?

Алгоритм — это точное описание порядка действий, которые должен выполнить исполнитель для решения задачи за конечное время.

Исполнитель – это устройство или одушевленное существо (человек), способное понять и выполнить команды, составляющие алгоритм.

Формальные исполнители: не понимают (и не могут понять) смысл команд.



Мухаммед ал-Хорезми
(ок. 783—ок. 850 гг.)

Свойства алгоритма

Дискретность — алгоритм состоит из отдельных команд, каждая из которых выполняется за конечное время.

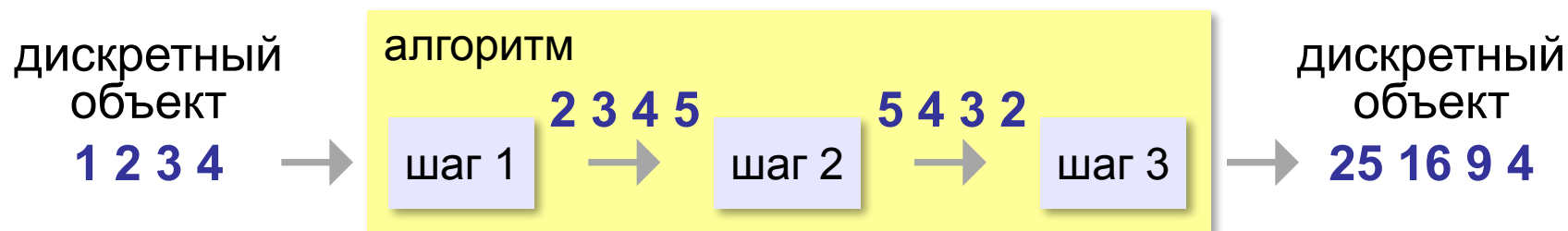
Детерминированность (определённость) — при каждом запуске алгоритма с одними и теми же исходными данными получается один и тот же результат.

Понятность — алгоритм содержит только команды, входящие в **систему команд исполнителя**.

Конечность (результативность) — для корректного набора данных алгоритм должен завершаться через конечное время.

Корректность — для допустимых исходных данных алгоритм должен приводить к правильному результату.

Как работает алгоритм?



- получает на вход дискретный объект
- в результате строит другой дискретный объект (или выдаёт сообщение об ошибке)
- обрабатывает объект по шагам
- на каждом шаге получается новый дискретный объект

Способы записи алгоритмов

- **естественный язык**

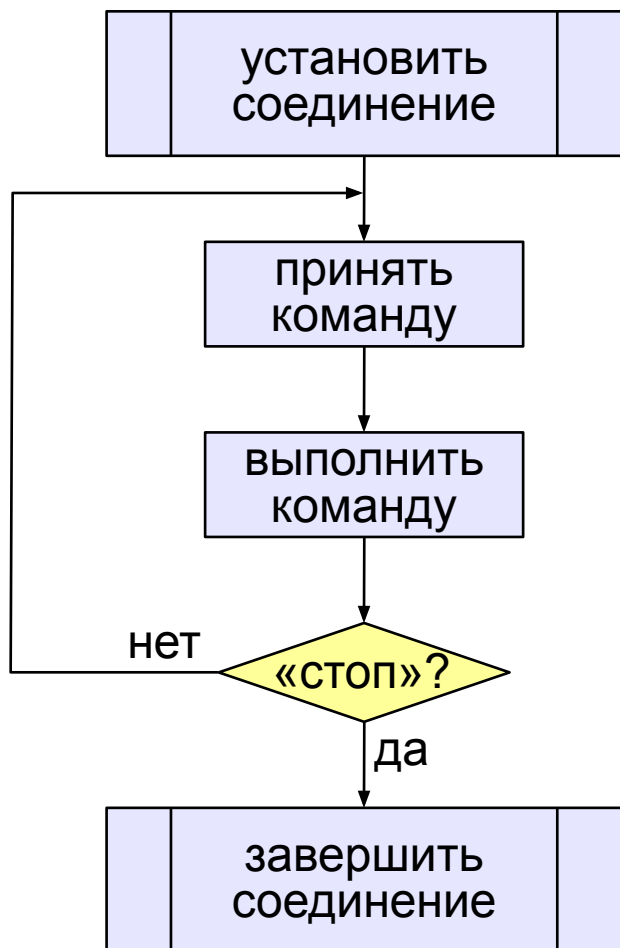
```
установить соединение  
пока не принята команда «стоп»  
    принять команду  
    выполнить команду  
завершить сеанс связи
```

- **псевдокод**

```
установить соединение  
нц  
    принять команду  
    выполнить команду  
кц_при команда = 'stop'  
завершить сеанс связи
```

Способы записи алгоритмов

- блок-схема



- программа

```
setConnection;  
repeat  
    cmd := getCommand;  
    executeCommand(cmd);  
until cmd = "stop";  
closeConnection;
```

Программирование на языке Паскаль

§ 55. Простейшие программы

Простейшая программа

название алгоритма

```
program qq;  
begin { начало программы }  
      { тело программы }  
end.  { конец программы }
```

комментарии в скобках {}
не обрабатываются



Что делает эта программа?

Вывод на экран

```
program qq;  
begin  
▶ write('2+');  
▶ writeln('2=?'); { на новую строку}  
▶ writeln('Ответ: 4');  
end.
```

Протокол:

2+

Ответ: 4

Задания

«В»: Вывести на экран текст «лесенкой»

Вася

пошел

гулять

«С»: Вывести на экран рисунок из букв

```
  Ж
  ЖЖЖ
 ЖЖЖЖЖ
ЖЖЖЖЖЖЖ
  НН  НН
  ZZZZZ
```

Сложение чисел

Задача. Ввести с клавиатуры два числа и найти их сумму.

Протокол:

Введите два целых числа

компьютер

25 30

пользователь

25+30=55

компьютер считает сам!

?

1. Как ввести числа в память?
2. Где хранить введенные числа?
3. Как вычислить?
4. Как вывести результат?

Сумма: псевдокод

```
программ qq;  
begin  
  { ввести два числа }  
  { вычислить их сумму }  
  { вывести сумму на экран }  
end.
```

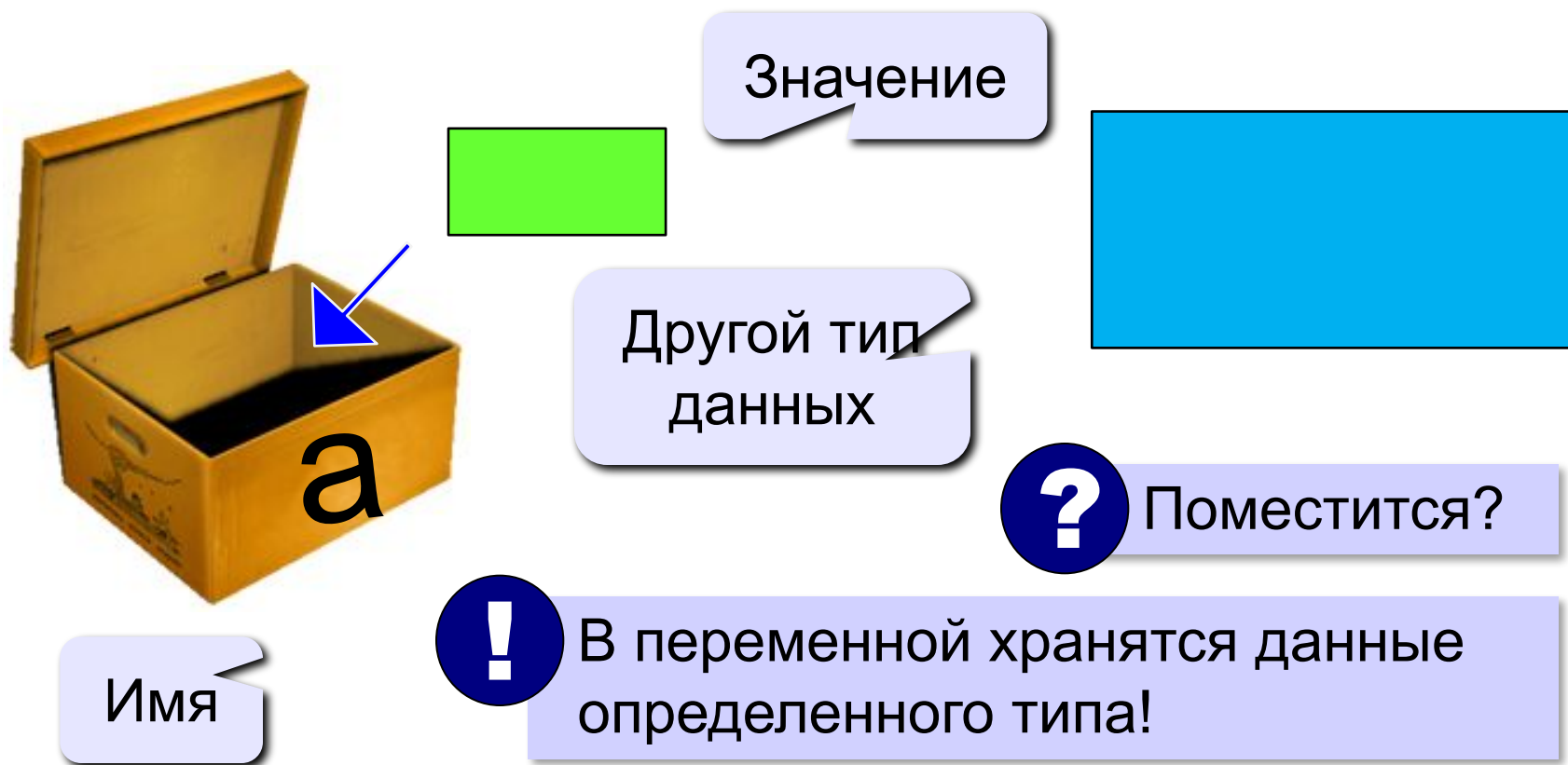
Псевдокод: алгоритм на русском языке с элементами Паскаля.



Компьютер не может исполнить псевдокод!

Переменные

Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



Имена переменных

МОЖНО использовать

- латинские буквы (A-Z)

заглавные и строчные буквы **НЕ различаются**

- цифры

имя не может начинаться с цифры

- знак подчеркивания _

НЕЛЬЗЯ использовать

- ~~русские буквы~~
- ~~пробелы~~
- ~~скобки, знаки +, =, !, ? и др.~~

Какие имена правильные?

AXby R&B 4Wheel Вася "PesBarbos"
TU154 [QuQu] _ABBA A+B

Объявление переменных

Типы переменных:

- **integer** { целая }
- **real** { вещественная }
- и другие...

выделение
места в памяти

Объявление переменных:

variable – переменная

тип – целые

```
var a, b, c: integer;
```

СПИСОК ИМЕН
переменных

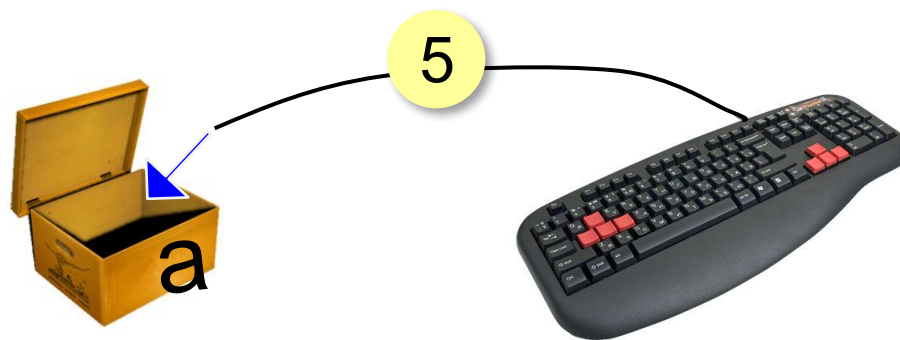
Тип переменной

- область допустимых значений
- допустимые операции
- объём памяти
- формат хранения данных
- для предотвращения случайных ошибок

Ввод значения в переменную

оператор
ввода

```
read ( a );
```



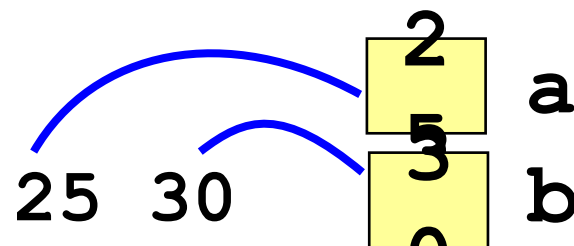
1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную **a**.

Ввод значений переменной

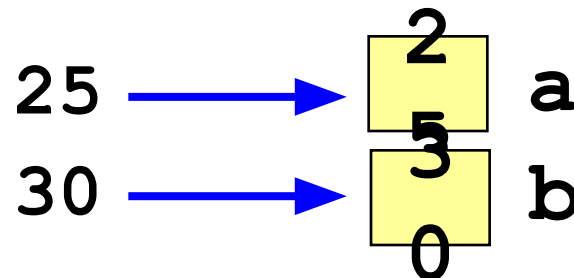
```
read ( a , b ) ;
```

Ввод значений двух переменных (через пробел или *Enter*).

через пробел:



через *Enter*:



Изменение значений переменной

```
var a, b: integer;
```

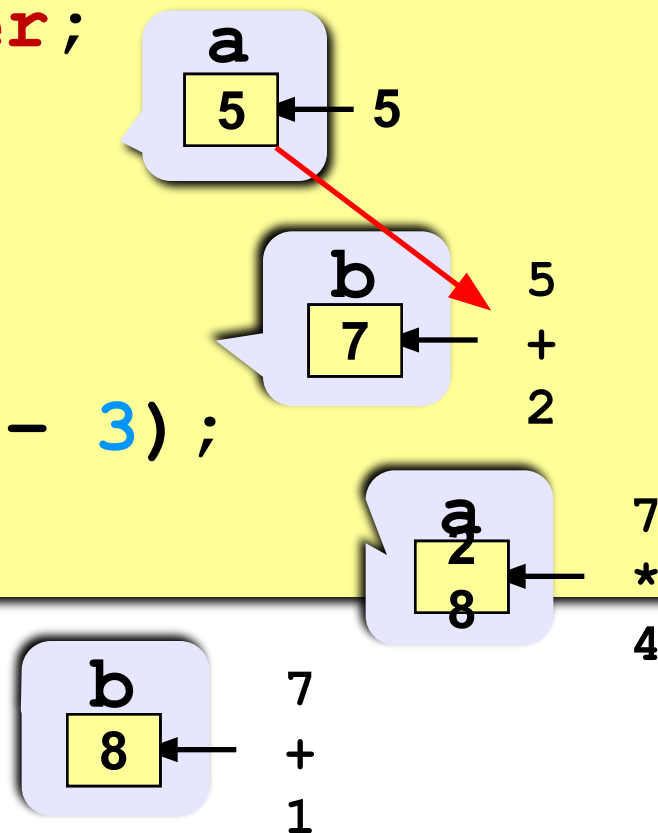
```
...
```

```
a := 5;
```

```
b := a + 2;
```

```
a := (a + 2) * (b - 3);
```

```
b := b + 1;
```



Вывод данных

```
write( a ); { вывод значения  
             переменной a }
```

```
writeln( a ); { вывод значения  
              переменной a и переход  
              на новую строку }
```

```
writeln( 'Привет!' ); { вывод текста }
```

```
writeln( 'Ответ: ', c );
```

{вывод текста и значения переменной c}

```
writeln ( a, '+', b, '=', c );
```

Сложение чисел: простое решение

```
program Sum;  
var a, b, c: integer;  
begin  
  read ( a, b );  
  c := a + b;  
  writeln ( c );  
end.
```



Что плохо?

Сложение чисел: полное решение

```
program Sum;  
var a, b, c: integer;  
begin  
  writeln('Введите два целых числа');  
  read ( a, b );  
  c := a + b;  
  writeln ( a, '+', b, '=', c );  
end.
```

Протокол:

компьютер

Введите два целых числа

25 30

пользователь

25+30=55

Снова про оператор вывода

Вычисление выражений:

```
writeln ( a, '+', b, '=', a+b );
```

Форматный вывод:


```
a := 123;  
write( a:5 );
```

__123
5 знаков

Программирование на языке Паскаль

§ 56. Вычисления

Типы данных

- `byte` { целые 0..255 }
- `shortint` { целые -128..127 }
- `word` { целые 0..65535 }
- `longint` { целые -2147483648..2147483647 }
- `single` {  Сколько байт в памяти? 4 байта }
- `real` { вещественная, 6 байта }
- `double` { вещественная, 8 байтов }
- `extended` { вещественная, 10 байтов }
- `boolean` { логическая, 1 байт }
- `char` { символ, 1 байт }
- `string` { символьная строка }

Арифметические выражения

3 1 2 4 5 6

`a := (c + b*5*3 - 1) / 2 * d;`

Приоритет (*старшинство*):

1) скобки

2) умножение и деление

3) сложение и вычитание

$$a = \frac{c + b \cdot 5 \cdot 3 - 1}{2} \cdot d$$

Деление, **div**, **mod**

Результат деления «/» – вещественное число:

```
var a: single;  
a := 2 / 3; → 0.6666...
```

div – деление нацело (остаток отбрасывается)

mod – остаток от деления

```
var a, b, d: integer;  
...  
d := 85;  
b := d div 10;  
a := d mod 10;
```

div и mod для отрицательных чисел

```
write(-7 div 2, ', ');  
write(-7 mod 2);
```

$$\begin{array}{l} -3 \\ -1 \end{array} \quad -7 = (-3) * 2 + (-1)$$



В математике не так!

$$-7 = (-4) * 2 + 1$$

остаток ≥ 0

Вещественные числа



Целая и дробная части числа разделяются точкой!

```
var x: double;  
...  
x := 123.456;
```

Форматный вывод:

```
a := 1;  
write( a/3 );  
write( a/3:7:3 );
```

$$3,333333 \cdot 10^{-1} = 0,3333333$$

```
3.333333E-001  
  0.333
```

всего знаков

в дробной части

Стандартные функции

- abs** (x) — модуль
- sqrt** (x) — квадратный корень
- sin** (x) — синус угла, заданного **в радианах**
- cos** (x) — косинус угла, заданного **в радианах**
- exp** (x) — экспонента e^x
- ln** (x) — натуральный логарифм
- trunc** (x) — отсечение дробной части
- round** (x) — округление до ближайшего целого

Случайные числа

Случайно...

- встретить друга на улице
- разбить тарелку
- найти 10 рублей
- выиграть в лотерею

Случайный выбор:

- жеребьевка на соревнованиях
- выигравшие номера в лотерее

Как получить случайность?



Случайные числа на компьютере

Электронный генератор



- нужно специальное устройство
- нельзя воспроизвести результаты

Псевдослучайные числа – обладают свойствами случайных чисел, но каждое следующее число вычисляется по заданной формуле.

Метод середины квадрата (Дж. фон Нейман)

зерно

564321

в квадрате

- малый период
(последовательность повторяется через 10^6 чисел)

318458191041

209938992481

Линейный конгруэнтный генератор

$X := (a * X + b) \bmod c$ | интервал от 0 до $c-1$

$X := (X + 3) \bmod 10$ | интервал от 0 до 9

$X := 0 \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 2 \rightarrow 5 \rightarrow 8$

$8 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 0$

зерно

зацикливание



Важен правильный выбор параметров a , b и c !

Компилятор GCC:

$a = 1103515245$

$b = 12345$

$c = 2^{31}$

Генератор случайных чисел

Вещественные числа в интервале [0,1):

```
var X, Y: double;  
...  
X := random; { интервал от 0 до 1 (<1) }  
Y := random; { это уже другое число! }
```

англ. *random* – случайный

Целые числа в интервале [0,10):

```
var K, L: integer;  
...  
K := random( 10 ) { интервал от 0 до 9 (<10) }  
L := random( 10 ) { это уже другое число! }
```

Другой интервал

Вещественные числа:

```
var X, a, b: double;
...
X := random*10; { расширение интервала: [0,10) }
X := random*10 + 5;
                { расширение и сдвиг: [5,15) }
X := random*(b-a) + a;
                { расширение и сдвиг: [a,b) }
```

Целые числа:

```
var K, a, b: integer;
...
K := random(10) + 5;    { [5,14] }
X := random(b-a+1) + a; { [a,b] }
```

Задачи

«А»: Ввести с клавиатуры три целых числа, найти их сумму, произведение и среднее арифметическое.

Пример:

Введите три целых числа:

5 7 8

$$5+7+8=20$$

$$5*7*8=280$$

$$(5+7+8)/3=6.667$$

«В»: Ввести с клавиатуры координаты двух точек (А и В) на плоскости (вещественные числа). Вычислить длину отрезка АВ.

Пример:

Введите координаты точки А:

5.5 3.5

Введите координаты точки В:

1.5 2

$$\text{Длина отрезка АВ} = 4.272$$

«С»: Получить случайное трехзначное число и вывести через запятую его отдельные цифры.

Пример:

Получено число 123.

Его цифры 1, 2, 3.