

Программирование на языке Паскаль

§ 54. Алгоритм и его свойства

§ 55. Простейшие программы

§ 56. Вычисления

§ 57. Ветвления

§ 58. Циклические алгоритмы

§ 59. Процедуры

§ 60. Функции

§ 61. Рекурсия

Программирование на языке Паскаль

**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 (одну и ту же команду всегда выполняют одинаково).



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

Свойства алгоритма

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

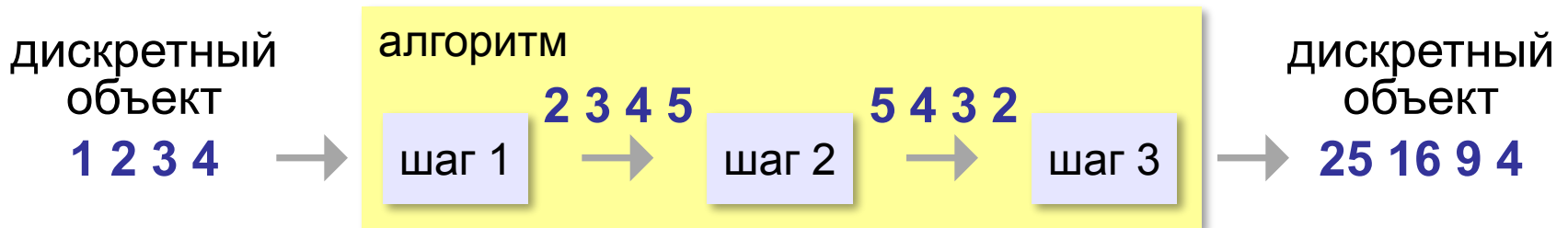
Детерминированность (определённость) — при каждом запуске алгоритма с одними и теми же исходными данными получается один и тот же результат.

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**

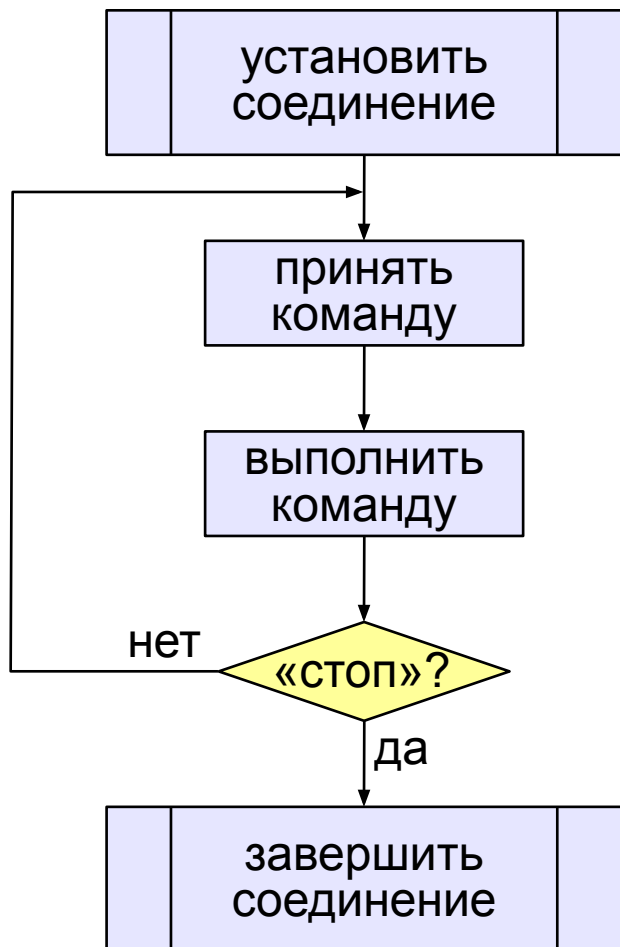
```
установить соединение  
пока не принята команда «стоп»  
    принять команду  
    выполнить команду  
завершить сеанс связи
```

- **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

Тема 1. Введение

Программа

Программа – это

- алгоритм, записанный на каком-либо языке программирования
- набор команд для компьютера

Команда – это описание действий, которые должен выполнить компьютер.

- откуда взять исходные данные?
- что нужно с ними сделать?

Оператор – это команда языка программирования высокого уровня.

1970 – язык Паскаль (Н. Вирт)

Простейшая программа

название алгоритма

```
Program { название программы } ;  
begin { начало программы }  
      { тело программы }  
end.   { конец программы }
```

комментарии в скобках {}
не обрабатываются



Что делает эта программа?

Вывод на экран

```
program z1;  
begin  
▶ write('2+');  
▶ writeln('2=?'); { на новую строку}  
▶ writeln('Ответ: 4');  
end.
```

Протокол:

2+

Ответ: 4

Вывод на экран

```
write({ список вывода }); { без перехода }
writeln({ список вывода }); { на новую строку }
```

Список вывода:

1.

Значение
арифметиче
ского
выражения.

```
write(SQRT(58)/0.5);
writeln(a, b, c);
```

2.

Список
переменных
через
запятую.

```
writeln('Чётное');
```

3.

Текст в
апострофах

Примеры:

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 );
```

Арифметические выражения

3 1 2 4 5 6

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

Приоритет (*старшинство*):

1) функция

2) скобки

3) умножение и деление

4) сложение и вычитание

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

Стандартные функции

abs (x) — модуль

sqrt (x) — квадратный корень

sqr (x) — квадрат

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

```
writeln ( 'x^8  ', exp (ln (x) * 8) : 0 : 3) ;
```

Задания

«А»: Вывести на экран значение переменной **V** (объёма шара), если **R=5** (R-радиус).

По виду : **V = {значение}**

$$V = \frac{4}{3} \pi R^3$$

«В»: Вывести на экран рисунок из букв

```

      Ж
     ЖЖЖ
    ЖЖЖЖЖ
   ЖЖЖЖЖЖЖ
  НН  НН
  ZZZZZ
  
```

«С»: Вывести на экран значение переменной **V** (объёма куба), если **H = 7** (H – длина ребра куба) .

По виду : **V = {значение}**

Сложение чисел

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

Протокол:

Введите два целых числа

25 30

25+30=55

компьютер

пользователь

компьютер считает сам!

?

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

Сумма: псевдокод

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

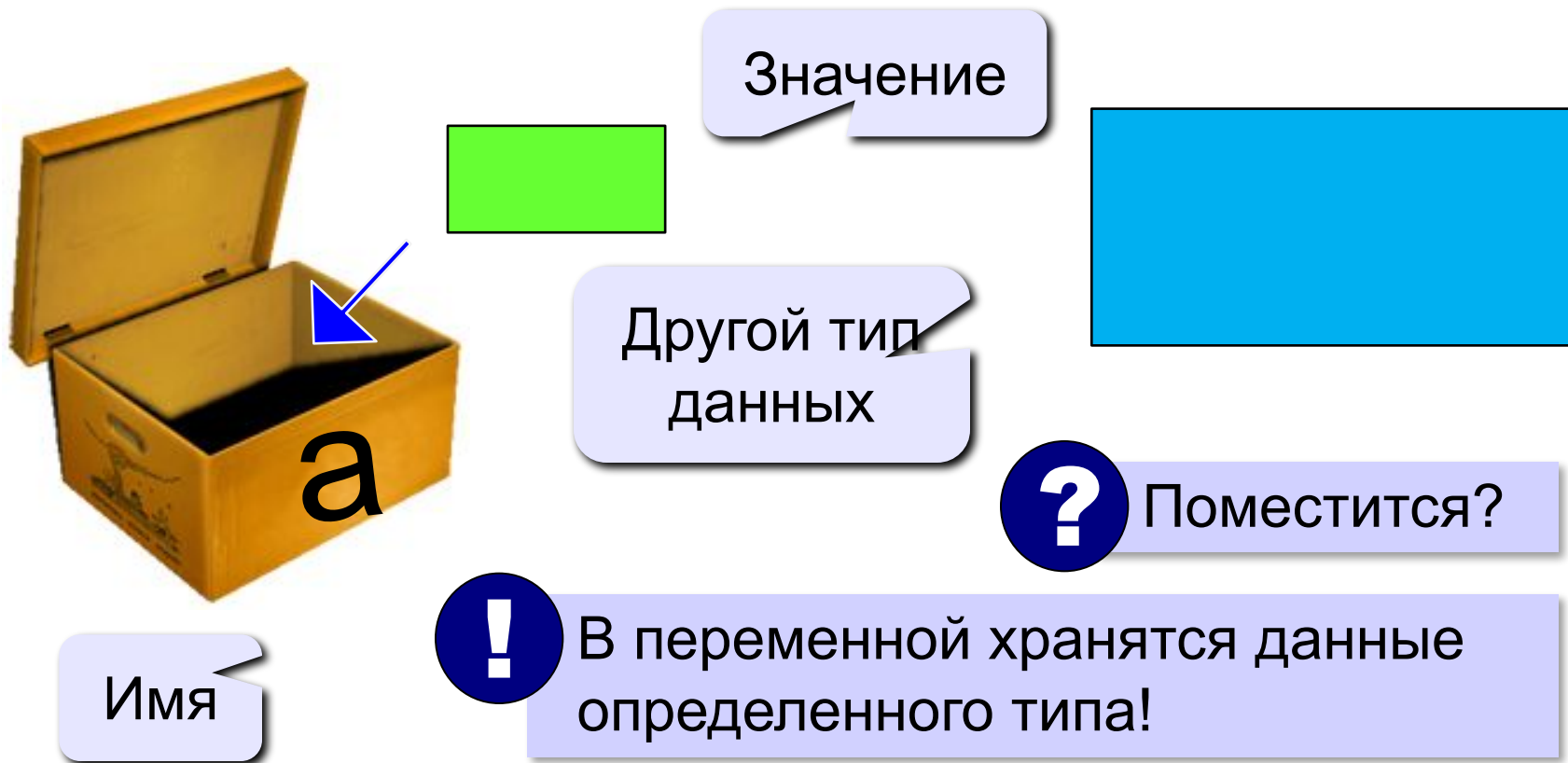
Псевдокод: алгоритм на русском языке с элементами Паскаля.



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

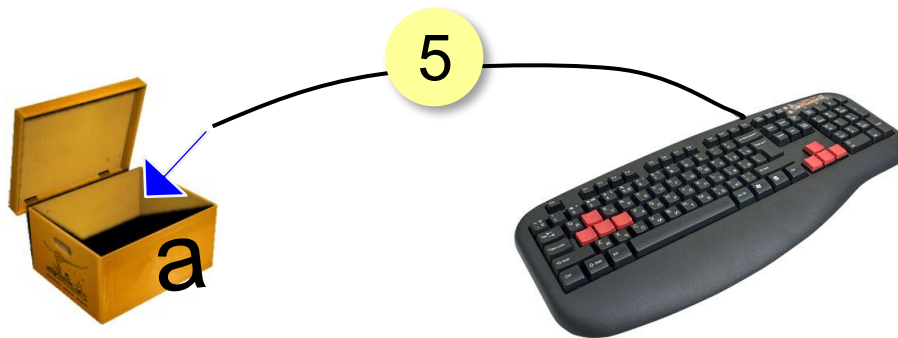
Тип переменной

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

Ввод значения в переменную

оператор
ввода

```
read ( a );
```



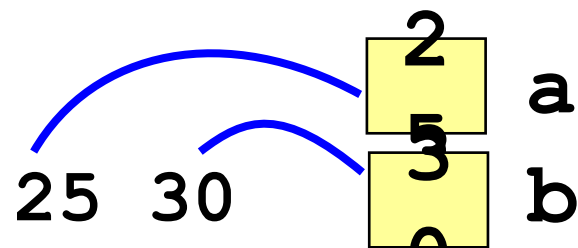
1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную **a**.

Ввод значений переменной

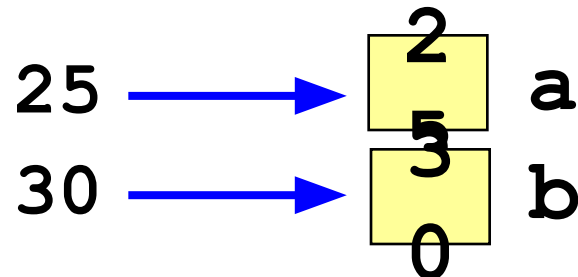
```
read ( a , b ) ;
```

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

через пробел:



через *Enter*:



Изменение значений переменной

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

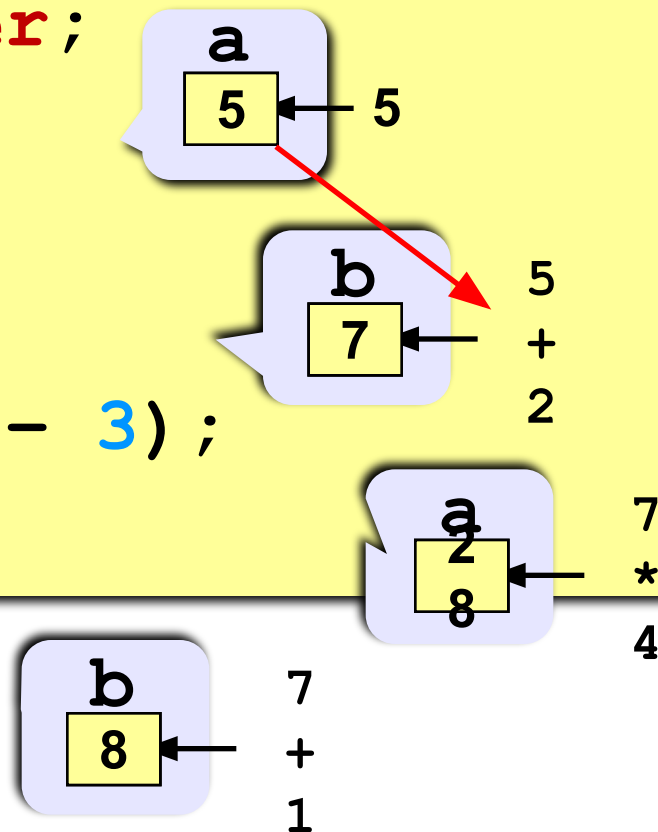
```
...
```

```
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

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

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

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