

# Программирование на языке Паскаль

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.
```

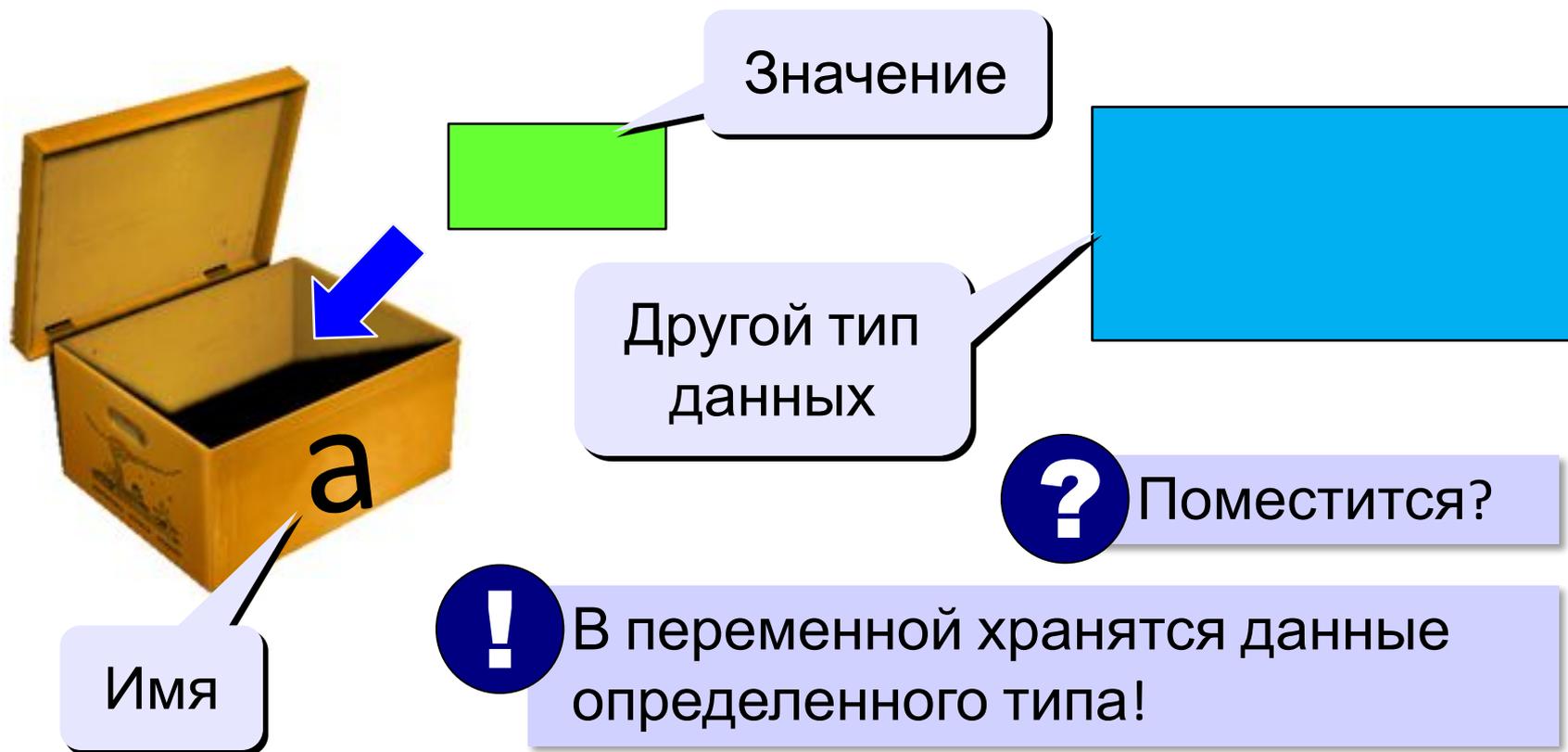
**Псевдокод:** алгоритм на русском языке с элементами Паскаля.



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...

## Объявление переменных:

выделение  
места в памяти

*variable* – переменная

тип – целые

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

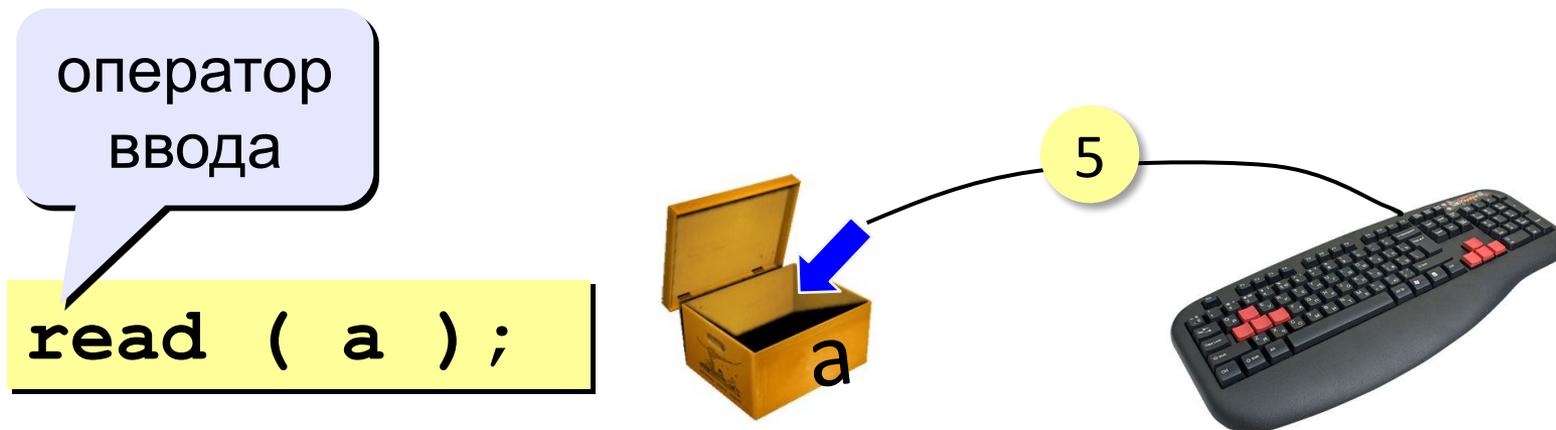
СПИСОК ИМЕН  
переменных

# Тип переменной

---

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

# Ввод значения в переменную



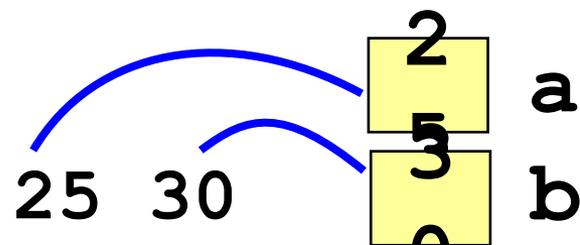
- ! 1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
- 2. Введенное значение записывается в переменную **a**.

# Ввод значений переменной

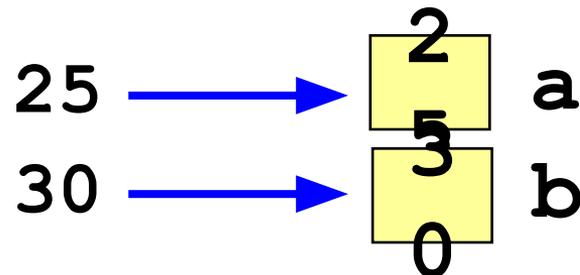
```
read ( a , b ) ;
```

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

через пробел:



через *Enter*:



# Изменение значений переменной

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

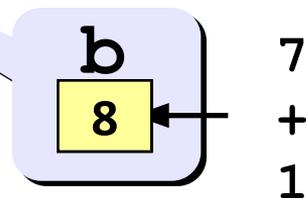
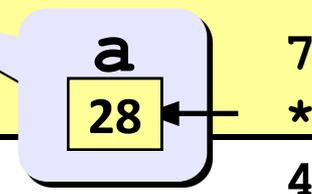
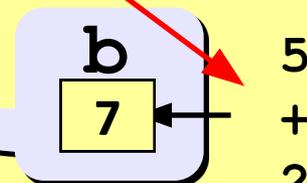
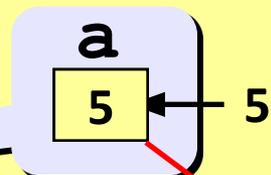
```
...
```

```
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