

# **Первые шаги в программировании**

# План урока

1. Из истории
2. Пустая программа
3. Вывод текста на экран
4. Переменные и их объявление
5. Тип переменной
6. Первая программа
7. Алфавит языка. Арифметические операции
8. Закрепление материала

# Из истории



- Язык программирования Паскаль был разработан в 1968 - 1970 годах швейцарским профессором Никлаусом Виртом и назван в честь великого французского математика и философа Блеза Паскаля (1623-1662г).
- Созданный для обучения студентов, язык оказался настолько хорош, что стал использоваться профессионалами.

# Пустая программа

Кумир

```
алг Куку  
нач  
| основная  
  программа  
кон
```

Паскаль

```
Program qq;  
Begin  
{основная программа}  
End.
```

# Вывод текста на экран

Напишем программу, которая выводит на экран такие строки:

**2+2=?    Ответ:4**

**алг** ВЫВОД

**нач**

ВЫВОД ' 2+'

ВЫВОД ' 2=?', НС

ВЫВОД ' ответ: 4'

**кон**

**Program** vivod;

**Begin**

Write('2+');

Writeln ('2=?');

Write ('ответ: 4')

**End.**

# Программа на псевдокоде

Напишем программу, которая выполняет сложение двух чисел. Компьютер не может выполнить псевдокод, нужно расшифровать его команды через операторы программирования. Здесь надо хранить данные в памяти. Для этого используют переменные.

1. Запрашивает у пользователя 2 числа
2. Складывает их
3. Выводит результат сложения

**алг** сумма

**нач**

1. Ввести 2 числа
2. Сложить их
3. Вывести результат

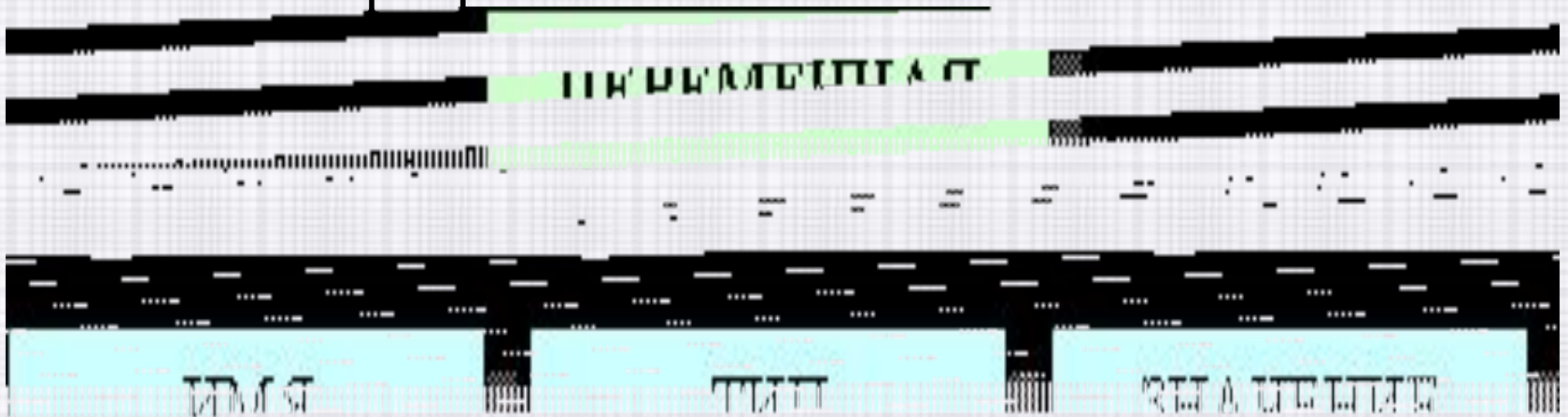
**кон**

# Переменные

**Переменная** - это область в оперативной памяти компьютера для хранения какой-либо информации.

**Переменная** – это величина, которая имеет имя, тип и значение.

**Значение переменной** может измениться во время выполнения программы.





# Объявление переменных

- ✓ Переменная определяет область памяти, где хранится только одно значение. При записи в неё нового значения «старое» стирается, и его уже никак не восстановить.
- ✓ Переменные в программе надо объявлять, то есть указать тип переменной и её имя (идентификатор).
- ✓ Например: **var a,b,c: integer;**



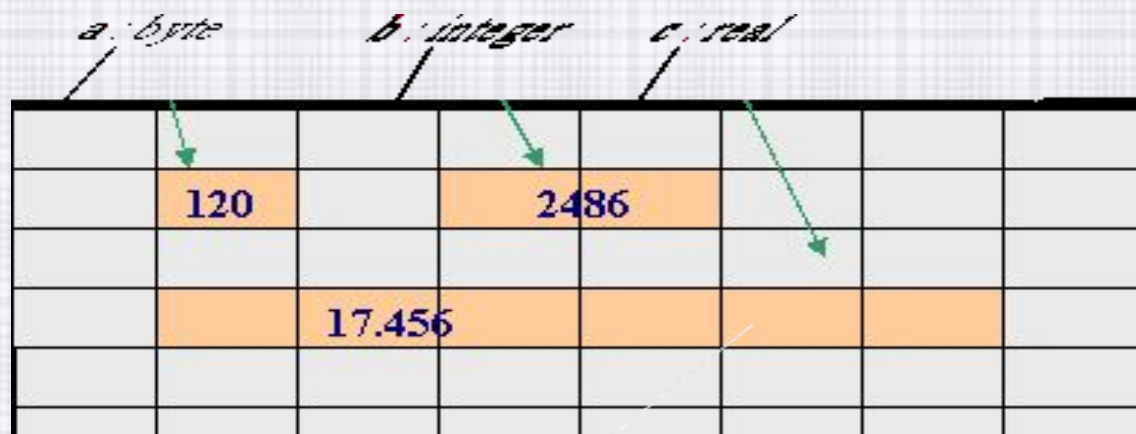
# ПРИМЕРЫ ОПИСАНИЯ ПЕРЕМЕННЫХ

Слово **var** - это сокращение от слова *variable* (переменная) и является зарезервированным словом.

В разделе описания переменных может содержаться несколько строчек, каждая из которых заканчивается точкой с запятой.

Для описания переменных необходимо перечислить через запятую их имена, а затем после двоеточия «:» указать их тип.

**Var**    **a:** byte;  
          **b:** integer;  
          **c, d, e:** real;  
          **f:** double;

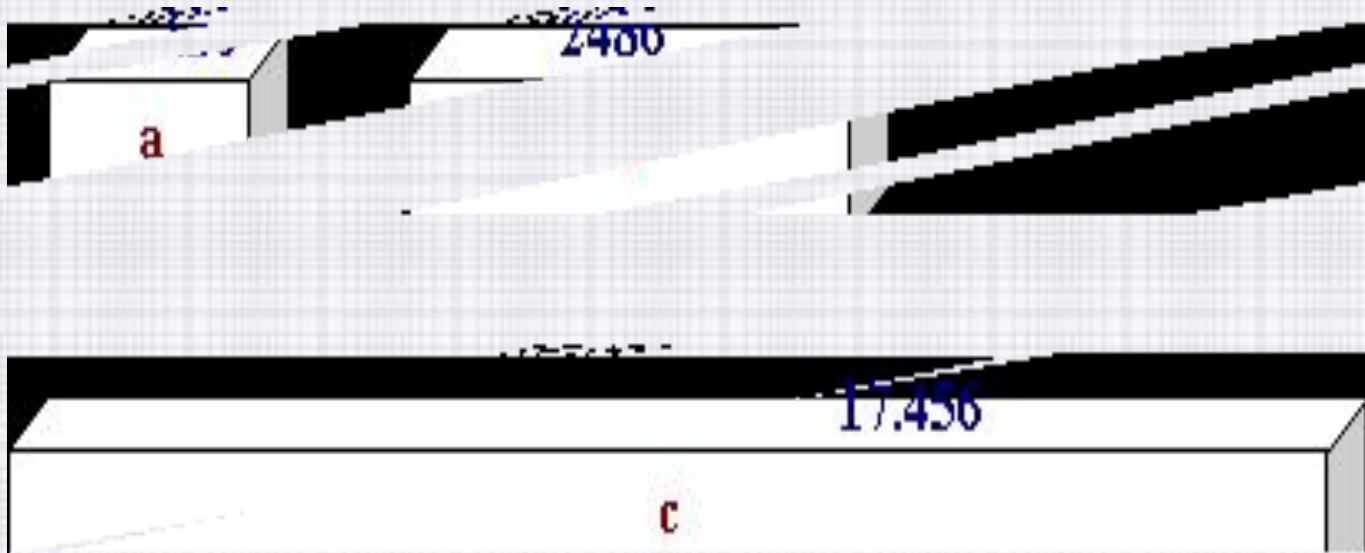


Наглядно переменную можно представить как коробочку, в которую можно положить на хранение что-либо.

**Имя переменной** – это надпись на коробочке

**Значение переменной** – это то, что хранится в ней в данный момент.

**Тип переменной** это то, что допустимо класть в эту коробочку.



# ИМЯ ПЕРЕМЕННОЙ

должно удовлетворять следующим требованиям:

- 1) **всегда начинаться с буквы**, после которой могут другие буквы или цифры;
- 2) **не должно быть пробелов, запятых** или других непредусмотренных знаков;
- 3) **нельзя использовать** служебные слова или операторы.

Например: **A1, spisok, A\_2**

# ОСНОВНЫЕ ТИПЫ ПЕРЕМЕННЫХ

## 1). нат - **word**

целые положительные числа

[0, 65535]

## 2). цел - **integer**

целые числа из интервала

[-32768, 32767];

## 3). вещ - **real**

дробные числа из интервала

$[-2.9 \cdot 10^{-39}, - + 2.9 \cdot 10^{-39}]$

## 4). лит - **CHAR** -1 символа (буквы, знака или кода, занимает 1 байт)

## 5). лит - **Cstring** - хранит слова, занимает 16 байт)

# 1. ЦЕЛЫЙ ТИП

<b>Тип</b>	<b>Диапазон значений</b>	<b>Размер памяти</b>
<b>Shortin</b>	-128 ... 127	1 байт
<b>Integer</b>	-32768 ... 32767	2 байта
<b>Longint</b>	+2147483647	4 байта
<b>Byte</b>	0 ... 255	1 байт
<b>Word</b>	0 ... 65535	2 байта

## 2. ВЕЩЕСТВЕННЫЙ ТИП

Тип	Диапазон значений	Длина мантиссы	Размер памяти
<b>Real</b>	$29\text{E-}39 \dots 1.7\text{E}39$	10-12	6 байт
<b>Single</b>	$1.5\text{E-}35 \dots 3.4\text{E}38$	7-8	4 байта
<b>Double</b>	$5\text{E-}324 \dots 1.7\text{E}302$	15-16	8 байт
<b>Extendent</b>	$3.4\text{E-}4932 \dots 1.1\text{E}4931$	19-20	10 байт
<b>Comp</b>	$-2\text{E}+63 \dots +2\text{E}63-1$		8 байт



## 3. СИМВОЛЬНЫЙ ТИП

**CHAR** – представляет собой тип данных, предназначенный для хранения 1 символа (буквы, знака или кода, занимает 1 байт)

**Cstring** – хранит слова, занимает 16 байт)



# 4. БУЛЕВСКИЙ (ЛОГИЧЕСКИЙ) ТИП

**BOOLEAN** - Имеется два значения, которые могут принимать переменные этого типа **“true”** - истина и **“false”** - ложь.

Допустимые операции сравнения: FALSE < TRUE.

Стандартные логические операции:

- **or** - логическое сложение
- **and** - логическое умножение
- **xor** - сложение по модулю
- **not** - логическое отрицание.

# Алфавит языка Арифметические операции

---

# АЛФАВИТ ЯЗЫКА

- **Заглавные и строчные латинские буквы и символы;**
- **Цифры 0, 1, 2, 3, 4, 5,...9;**
- **22 спец. символа + - < > : ; / . , @ \* ( ) { } # \$ ^ [ ] “**

# АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ В ПАСКАЛЕ

## 1. Операции с дробными числами (обычные, как в математике).

- вычитание;

+ сложение;

\* умножение;

/ деление.

## 2. Операции с целыми числами.

- вычитание; + сложение; \* умножение;

Деление представлено двумя операциями:

**div** - целая часть от деления, **mod** - остаток.

$$23 \text{ div } 3 = 7$$

$$23 \text{ mod } 3 = 2$$

$$9 \text{ div } 4 = 2$$

$$9 \text{ mod } 4 = 1$$

$$35 \text{ div } 10 = 3$$

$$35 \text{ mod } 10 = 5$$

# ПРИОРИТЕТ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

Порядок выполнения действий - слева направо.

1. возведение в степень
2. умножение (\*) и деление ( /, div, mod).
3. сложение и вычитание.

Выражения в скобках вычисляются в первую очередь.

# КОНСТАНТЫ

- Значения констант не меняются в данной задаче, задаются в разделе описания констант.
- Этот раздел начинается со слова `const`, входящего в число зарезервированных.
- При попытке изменить значение константы компилятор выдаст сообщение об ошибке.

Пример описания констант

```
Const a = 10; b = 1.6;
```