

# Основы программирования

1. Правила программирования;
2. Структура программы;
3. Операции;
4. Типы данных;
5. Структура имен переменных, констант, процедур, функции;
6. Структурное программирование.

# 1. Правила программирования

## **Методы программирования программных средств.**

Этапы:

### 1. Определение условий задачи

определяются условия задачи и необходимо ясно понять, что требуется для её решения. Основная цель в данном случае - отсеять второстепенные аспекты от основной сути задачи

### 2. Анализ задачи

определяются входные данные, выходные, промежуточные и какие дополнительные трудности могут возникнуть при решении поставленной задачи

### 3. Создание алгоритма решения задачи

со схемой алгоритма гораздо проще; для сложных задач часто применяется метод декомпозиции

# 1. Правила программирования

## 4. Реализация алгоритма

реализация алгоритма на языке программирования

## 5. Тестирование и отладка готовой программы

поиске всевозможных ошибок и позволяет добиться правильности работы программы

# Основы программирования

1. Правила программирования;
2. **Структура программы;**
3. Операции;
4. Типы данных;
5. Структура имен переменных, констант, процедур, функции;
6. Структурное программирование.

## 2. Структура программы

Структура программы на языке  
Pascal

**Program** <имя программы>

**Uses** список модулей

**Label** описание меток

**Const** описание констант

**Type** описание типов

**Var** описание переменных

**Procedure, function** описание процедур, функций

**Begin**

Тело программы

**end.**

Разделы описаний  
могут идти в разном  
порядке

## 2. Структура программы

Раздел описания переменных начинается с ключевого слова **var**, после которого следуют элементы описания вида.

*список имен переменных :  
тип;*

Имена в списке перечисляются через запятую. Например:

**var**

a, b, c: integer;

i : integer;

d: real := 3.7;

## 2. Структура программы

Раздел описания именованных констант начинается со служебного слова **const**, после которого следуют элементы описания вида

ИМЯ КОНСТАНТЫ =  
значение;

Пример:

**const**

val = 123.456;

Count = 10;

Surname = 'Алебардин';

## 2. Структура программы

**Комментарии** - это участки кода, игнорируемые компилятором и используемые программистом для пояснения текста программы.

```
{ Это комментарий }
```

```
(* Это  
тоже  
комментарий *)
```

Комментарием также считается любая последовательность символов после символов `//` и до конца строки:

```
var Version: integer; // Версия продукта
```

Комментарии разных типов могут быть вложенными:

```
{ Это еще один  
(* комментарий *) }
```

## 2. Структура программы

Пример

```
:  
program MySuperProgram;  
const      // объявление констант  
  N = 10;  
var       // объявление переменных  
  a, b: integer;  
  x: real;  
begin  
  readln(a,b); // чтение значений переменных с клавиатуры  
  x := a/b + N; // выполнение действия  
  writeln(x);  // вывод значения переменной на экран  
end.
```

# Основы программирования

1. Правила программирования;
2. Структура программы;
3. **Операции;**
4. Типы данных;
5. Структура имен переменных, констант, процедур, функции;
6. Структурное программирование.

# 3. Операции

## Арифметические операции

Для вещественных и целых чисел:

- сложение (+)
- вычитание (−)
- умножение (\*)
- деление (/)

Для целых чисел:

- div – целочисленное деление
- mod – остаток от целочисленного деления

# 3. Операции

## Операции сравнения

Для вещественных и целых чисел:

- строгие  $<$ ,  $>$
- нестрогие  $<=$ ,  $>=$
- равно  $=$
- не равно  $<>$

## 3. Операции

### Логические операции

К логическим относятся операции **and**, **or** и **xor**, а также унарная операция **not**, имеющие операнды типа `boolean` и возвращающие значение типа `boolean`.

# 3. Операции

Конъюнкция – И, AND,  $\wedge$ , &,  $\times$

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкция – ИЛИ, OR,  $\vee$ , +

A	B	A $\vee$ B
0	0	0
0	1	1
1	0	1
1	1	1

Отрицание – НЕ, NOT

A	$\bar{A}$
0	1
1	0

Сложение по модулю 2 – исключающее "ИЛИ", XOR,  $\oplus$

A	B	A $\oplus$ B
0	0	0
0	1	1
1	0	1
1	1	0

# 3. Операции

## Приоритет операций

Приоритет определяет порядок выполнения операций в выражении. Первыми выполняются операции, имеющие высший приоритет. Операции, имеющие одинаковый приоритет, выполняются слева направо.

Операция	Приоритет
@, not, ^, +, - (унарные), new	1
*, /, div, mod, and, shl, shr	2
+, - (бинарные), or, xor	3
=, <>, <, >, <=, >=, =>	4

Для изменения порядка выполнения операций в выражениях используются скобки.

# Основы программирования

1. Правила программирования;
2. Структура программы;
3. Операции;
4. **Типы данных;**
5. Структура имен переменных, констант, процедур, функции;
6. Структурное программирование.

# 4. Типы данных



## 4. Типы данных

### Целые типы (точные)

Длина, байт	Название	Диапазон значений	Операции
1	Byte	0..255	+, -, /, *, Div, Mod, >=, <=, =, <>, <, >
1	Shortint	-128..+127	
2	Word	0..65535	
2	Smallint	-32768..32767	
4	Integer, Longint	-2147483648..2147483647	

## 4. Типы данных

Вещественные типы (приближенные)

Длина, байт	Название	Диапазон значений	Значащих цифр	Операции
6	Real	$2,9 \times 10^{-39} - 1,7 \times 10^{38}$	11-12	+, -, /, *, >=, <=, =, <>, <, >
4	Single	$1,5 \times 10^{-45} - 3,4 \times 10^{38}$	7-8	
8	Double	$5 \times 10^{-324} - 1,7 \times 10^{308}$	15-16	
10	extended	$3,4 \times 10^{-4932} - 1,1 \times 10^{4932}$	19-20	

## 4. Типы данных

Особенности в использовании различных типов данных

```
var  
  rVal : single;  
begin  
  rVal := 1000000 + 0.1;  
  writeln(rVal);  
end.
```

Результат: 1 000 000

```
var  
  bVal : byte;  
begin  
  bVal := 255+2;  
  writeln(bVal);  
end.
```

Результат: 1

# 4. Типы данных

Особенности в использовании различных типов данных

```
var  
  siVal : ShortInt;  
begin  
  siVal := 127+1;  
  writeln(siVal);  
end.
```

Результат: -128

## 4. Типы данных

### Представление чисел без знака

$72_{10} = 1001000_2$  в однобайтовом формате (Byte)

Номера разрядов	7	6	5	4	3	2	1	0
Биты числа	0	1	0	0	1	0	0	0

$72_{10} = 1001000_2$  в двухбайтовом формате (Word)

Номера разрядов	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Биты числа	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

# Представление чисел со знаком

Три формы записи:

- прямой код
- обратный код
- дополнительный код

## 4. Типы данных

### Представление чисел со знаком

Прямой код числа - 1

1	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Знак числа "-"

Прямой код числа - 127

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

Знак числа "-"

Число: -1

Код модуля числа: 0 0000001

Обратный код числа: 1 1111110

1	1	1	1	1	1	1	0
---	---	---	---	---	---	---	---

Число: -127

Код модуля числа: 0 1111111

Обратный код числа: 1 0000000

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Дополнительный код числа - 1

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

Дополнительный код числа - 127

1	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

## 4. Типы данных

### Вычисление в дополнительном коде

Десятичная запись

$$\begin{array}{r} + 3 \\ -10 \\ \hline -7 \end{array}$$

Двоичные коды

$$\begin{array}{r} + 0\ 0000011 \\ 1\ 1110110 \\ \hline 1\ 1111001 \end{array}$$

Дополнительный код числа -10  
Дополнительный код числа -7

Десятичная запись

$$\begin{array}{r} + 10 \\ -3 \\ \hline 7 \end{array}$$

Двоичные коды

$$\begin{array}{r} + 0\ 0001010 \\ 1\ 1111101 \\ \hline 0\ 0000111 \end{array}$$

Дополнительный код числа -3  
перенос отбрасывается

Десятичная запись

$$\begin{array}{r} + -3 \\ -7 \\ \hline -10 \end{array}$$

Двоичные коды

$$\begin{array}{r} + 1\ 1111101 \\ 1\ 1111001 \\ \hline 1\ 1110110 \end{array}$$

Дополнительный код числа -3  
Дополнительный код числа -7  
Дополнительный код числа -10  
перенос отбрасывается

## 4. Типы данных

### Вычисление в дополнительном коде

Тип данных: один байт со знаком (ShortInt  
-128..127)  
 $127 + 1 = ?$

127 : 0111 1111

1 : 0000 0001

1000 0001  $\rightarrow -1$

$$127 + 1 = -1$$

# Основы программирования

1. Правила программирования;
2. Структура программы;
3. Операции;
4. Типы данных;
5. **Структура имен переменных, констант, процедур, функции;**
6. Структурное программирование.

# 5. Структура имен переменных, констант, процедур, функции

## Имена переменных

Имя переменной должно отражать суть хранимых данных

Структура имени переменной <тип данных> <суть>

Фамилия – stSurName

День рождения – dtBDate

Итоговая сумма - rSumTotal

## Константы

cNum = 25

## Функции

a = fCalcResult(12,4)

## Процедуры

pSetValue(25,'Name')

# Основы программирования

1. Правила программирования;
2. Структура программы;
3. Операции;
4. Типы данных;
5. Структура имен переменных, констант, процедур, функции;
6. **Структурное программирование.**

## 6. Структурное программирование

```
Var
intA, I : Integer;
intMin : Integer;
Begin
intMin:=32767;
For I:=1 to 10 do
Begin
Write('Vvedi chislo ');
Readln(intA);
If intA < intMin then
intMin:=intA;
end;
Writeln('MIN=',intMin);
Readln;
end.
```

```
Var
intA, I : Integer;
intMin : Integer;
Begin
intMin:=32767;
For I:=1 to 10 do
Begin
Write('Vvedi chislo ');
Readln(intA);
If intA < intMin then intMin:=intA;
end;
Writeln('MIN=',intMin);
Readln;
end.
```