



**Язык
программирования
Паскаль**

Языки программирования

Алгоритм на естественном языке компьютеру анализировать очень сложно, т. к. они являются недостаточно строгими. Одно и то же слово может обозначать разные понятия. Поэтому для создания программ (инструкций компьютеру) используются специально разработанные языки программирования (или алгоритмические языки).

Уровни языков программирования

Языки низкого уровня (ЯНУ) – машинный язык, ассемблер, - предназначены для обращения к процессору и др. устройствам. Для человека неудобны.

Переносимости на другие аппаратные платформы нет. По быстродействию и экономности использования ресурсов ЯНУ являются наиболее эффективными.

Языки программирования

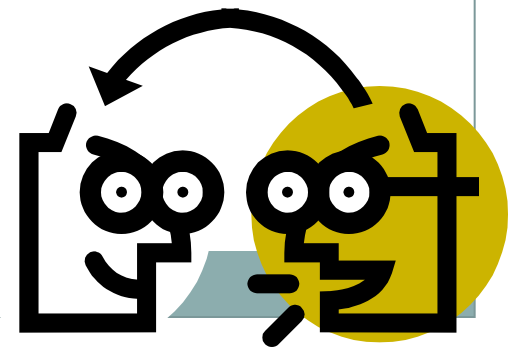
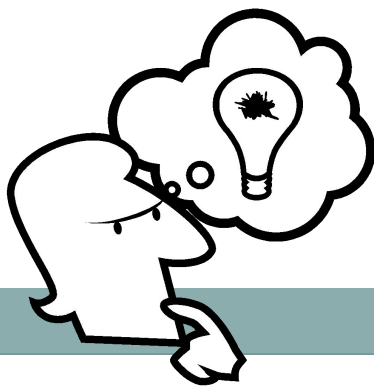
Языки высокого уровня (ЯВУ) – ближе к человеку. Используют термины решаемых задач и не требуют знания устройства компьютера. Одна команда ЯВУ выполняет множество команд ЯНУ. Есть переносимость на другие аппаратные и программные платформы. К ЯВУ относятся: Pascal, Delphi, Basic, Fortran, Prolog, Lisp, ...

Языки среднего уровня (ЯСУ) – сочетают возможности ЯВУ и ЯНУ. Более эффективные, чем ЯВУ, для программиста проще, чем ЯНУ, но сложнее, чем ЯВУ. К ЯСУ относятся: C, C++, Forth...

Трансляторы

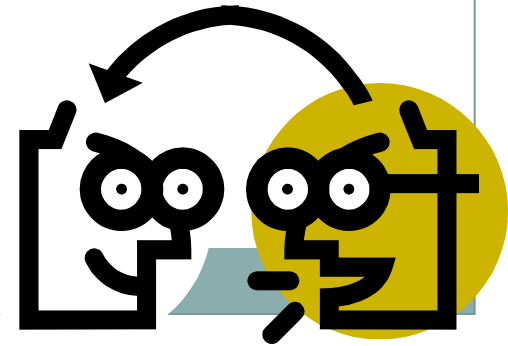
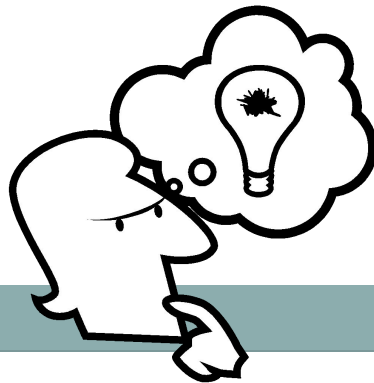
Трансляторы – программы, переводящие исходный текст программы с символьных языков (команды которых записываются словами) на машинный язык. Различают такие виды трансляторов:

Интерпретаторы – После запуска анализируют исходный текст построчно. Выделив отдельную команду, сразу выполняют её или сообщают об ошибке. Позволяют выполнить отдельную команду, не связывая её с другими. Удобны при отладке программы, но неэффективны при работе с готовой программой, так как тратят время на перевод.



Трансляторы

Компиляторы – После запуска анализируют весь исходный текст и записывают на диск готовый перевод в машинных кодах (независимую от с/п исполняемую программу), либо сообщают обо всех обнаруженных ошибках. Полученная исполняемая программа работает эффективно, но для отладки такой транслятор менее удобен.



Из истории



Среди современных языков программирования одним из самых популярных является язык Паскаль. Этот язык разработан в 1971 году и назван в честь **Блеза Паскаля** – французского ученого, изобретателя механической вычислительной машины. Автор языка Паскаль – швейцарский профессор **Никлаус Вирт**.

Языки программирования, имитирующие естественные языки, называются языками «высокого уровня». Насчитывается несколько сотен языков.

Одним из первых языков высокого уровня был Фортран. Используется для инженерных и научных расчетов.

В 1958 г. появилась первая версия языка АЛГОЛ. По сути, является родоначальником всех языков.

Наиболее известный язык BASIC, созданный в 1965 г. с появлением персональных компьютеров.

В начале 70 –х был разработан язык Паскаль. Важное достоинство – это структурный язык программирования (в ПАСКАЛЕ предусмотрены управляющие структуры – последовательность, выбор, итерация и структуры данных – массивы, записи, файлы, наборы и классы).

Разработан в одно время с ПАСКАЛЕМ и язык СИ.

Язык АДА.

Наибольшая популярность у языка СИ ++ и Java – непосредственного наследника.

Основные средства языка



Любой естественный язык состоит из нескольких основных элементов: символов, слов, словосочетаний и предложений. В любом языке программирования имеются аналогичные структурные элементы: символы, слова, выражения (словосочетания) и операторы (предложения).

Символы языка – это элементарные знаки, используемые при составлении любых текстов. Набор таких символов называют **алфавитом языка**.

Алфавит языка Паскаль

- 1) Все латинские прописные и строчные буквы (A-Z, a-z);
- 2) Арабские цифры (0,1, 2, ... 9);
- 3) Буквы русского алфавита (в сообщениях, комментариях);
- 4) Знаки арифметических действий + - * / ;
- 5) Знаки отношений = > <;
- 6) Знак присваивания := ;
- 7) Скобки () { } [] ;
- 8) Разделители . , : ; ‘ пробел;
- 9) Специальные символы
! ? % \$ # _ @

Служебное слово – слово, имеющее определенное смысловое значение.

Например, PROGRAM, Begin, End, Var

Составные части программы



заголовок программы

program имя программы;

раздел описаний;

begin

раздел операторов;

end.

Идентификаторы

- **Состоят из букв, цифр и знака подчёркивания.**
- **Русские буквы не допустимы.**
- **Начинаются с буквы или знака подчёркивания.**
- **Не различают прописные и строчные буквы.**

Составные части программы

Разделы программы отделяются друг от друга ;

Заголовок программы задает её имя, состоит из служебного слова Program, имени программы, образованного по правилам использования идентификаторов Паскаля, и точки с запятой.

НАПРИМЕР: `program my_prog01;`

Составные части программы

В разделе описаний объявляются все данные, встречающиеся в программе и их параметры: имя, тип, возможные значения.

Раздел команд. В нём описывается алгоритм решения задачи. Этот раздел заключён между словами `begin` и `end`. Операторы всегда должны отделяться друг от друга точкой с запятой независимо от того, располагаются они в одной строке текста программы или в разных. После `end`, последнего в программе, обязательно ставится точка.

Структура программы на Паскале

Program имя;

label; - раздел меток

const; - раздел констант

type; - раздел типов

var ; - раздел переменных

procedure; - раздел процедур

function; - раздел функций

Begin

оператор 1;

оператор 2;

- - - - -

оператор n;

End.

Структура программы

В языке Паскаль должны быть описаны все константы, типы, переменные, которые будут использоваться программой. Как правило придерживаются строгого порядка следования описаний.

Тело программы начинается со слова **begin**, затем следуют операторы языка, реализующие решение задачи.

Операторы отделяются друг от друга точкой с запятой, могут располагаться в одну строчку или начинаться с новой строки (в этом случае их также необходимо разделять точкой с запятой). Заканчивается тело программы словом **end**.

Рекомендуют операторы размещать по одному в строке.

Структура программы

НАПРИМЕР:

Program имя программы;
 const раздел констант;
 type раздел типов;
 var раздел переменных;

Begin

 оператор 1;

 оператор 2;

 - - - - -

 оператор n;

End.

Структура программы

В тексте программы могут быть комментарии в фигурных скобках или в круглых скобках в сопровождении символа *. Комментарии не выполняются программой, а служат для пояснения её отдельных частей.

НАПРИМЕР:

```
program one;
```

```
Const a=7;
```

```
Var b, c: real;
```

```
Begin
```

```
  C:=a+2;
```

```
  B:=c-a*sin(a)
```

```
end.
```


Стандартные типы данных



Тип данных определяет возможные значения констант, переменных, функций, выражений, принадлежащих к этому типу, форму представления в ПК и операции, которые могут выполняться над ними.

Стандартные типы данных

<u>Название</u>	<u>Диапазон значений</u>	<u>Память, байт</u>
Byte <i>байтовый</i>	0 - 255	1
Shortint <i>короткий целый</i>	(-128) - 127	1
Integer <i>целый</i>	(-32768)-32767	2
Word <i>слово</i>	0 - 65535	2
Longint <i>длинный целый</i>	(-2147483648)-2147483647	4
Single <i>с одинарной точностью</i>	1,5E-45 – 3,4E38	4
Real <i>вещественный</i>	2,9E-39 – 1,7E38	6
Double <i>с двойной точностью</i>	5,0E-324 – 1,7E308	8
Extended <i>с повышенной точности</i>	3,4E-4932 – 1,1E4932	10
Comp <i>сложный</i>	(-2E63+1) - 2E63-1	8
Char <i>символьный</i>	Символы кодовой таблицы	1
Boolean <i>логический</i>	true, false	1

Типы данных делятся на

- Скалярные (простые)
 - Стандартные – предлагаются разработчиками ТР.
 - Пользовательские – перечисляемые и интервальные.
- Структурированные (составные)
 - Строки
 - Массивы
 - Множества
 - Записи
 - файлы

Данные целого типа (integer)



1. **Константа целого типа** – любое десятичное число, записанное без точки («+» или «-»).

Диапазон от -32768 до +32767

2. **Переменные целого типа** – это переменные, принимающие в качестве своих значений константы целого типа.

3. Над данными целого типа можно выполнять **арифметические операции**, которые дают целый результат:

а) сложение;

б) вычитание;

в) умножение;

г) деление с отбрасыванием дробной части – DIV;
(например, $5 \text{ div } 3 = 1$)

д) получение целого остатка при делении целого данного на целое – MOD (например, $5 \text{ mod } 3 = 2$).

Данные действительного типа (real)



- 1. Константы действительного типа** могут быть представлены в двух видах: с фиксированной точкой и плавающей точкой. Константа с фиксированной точкой может изображаться десятичным числом с дробной частью. Дробная часть отделяется от целой точкой.
В математике для изображения очень больших и малых чисел используется запись числа с десятичным порядком. В Паскале они имеют следующий вид: mEr , где m – мантисса, E – признак записи числа с десятичным порядком, r – порядок числа.
Например, $4E-5$, $0.547E+3$, $5.47E+1$, $54.7E+1$
- 2. Переменные действительного типа** – это переменные, принимающие в качестве своих значений числа с фиксированной точкой или плавающей.
- 3. Арифметические операции:**
 - а) сложение;
 - б) вычитание;
 - в) умножение;
 - г) деление.

Данные логического типа (boolean)



1. **Логические константы** – TRUE (истина), FALSE (ложь).
2. **Переменные логического типа** принимают одно из этих значений.
3. Для сравнения: $>$, $<$, $=$, $<>$, $>=$, $<=$
Операции:
 - а) OR - логическое сложение (или);
 - б) AND - логическое умножение (и);
 - г) NOT – логическое отрицание (не).

Данные символьного типа (char)



Данные символьного типа позволяют представлять в программах тексты и производить над ними некоторые редакционные операции.

1. **Символьная или литерная константа** – это любой символ языка, заключенный в апострофы.
2. **Символьная переменная** - это переменная, принимающая значение символьной константы.
3. **Операции** : >, <, =, <>, >=, <=

Стандартные функции:

<u>Функция</u>	<u>Обозначение</u>
Модуль числа	Abs(x)
Синус	Sin(x)
Косинус	Cos(x)
Арктангенс	Arctan(x)
ПИ	Pi
e^x	Exp(x)
Натуральный логарифм	Ln(x)
x^2	Sqr(x)
Корень квадратный из x	Sqrt(x)
Целая часть числа	Int(x)
Дробная часть числа	Frac(x)
Округление числа	Round(x)
Отсекание дробной части	Trunc(x)
Случайное число от 0 до 1	Random
Случайное число от 0 до n	Random (n)

Остальные часто встречающиеся функции (тангенс, арксинус и т.д.) моделируются из уже определенных с помощью известных математических соотношений.

НАПРИМЕР:

$$\text{tg}(x)=\sin(x)/\cos(x)$$

Определенную проблему представляет возведение в степень. Если значение степени n – целое, то можно n раз перемножить x или воспользоваться формулой:

Для положительного x

$$\text{exp}(n*\ln(x))$$

Для отрицательного x

$$-\text{exp}(n*\ln(\text{abs}(x)))$$