

Программирование на языке Паскаль

Алгоритм

Алгоритм – это четко определенный план действий для исполнителя.

Свойства алгоритма

- **дискретность**: состоит из отдельных шагов (команд)
- **понятность**: должен включать только команды, известные исполнителю
- **конечность**: заканчивается за конечное число шагов
- **массовость**: может применяться многократно при различных исходных данных
- **корректность**: дает верное решение при любых допустимых исходных данных

Программа

Программа – это

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

Команда – это описание действий, которые должен выполнить компьютер.

Оператор – это команда языка программирования высокого уровня.

Языки программирования

//Даже при наличии десятков тысяч программ разработанных для IBM PC пользователям может потребоваться что-то такое, чего не делают (или делают что-то не то) имеющиеся программы. В этих случаях следует использовать системы программирования, т.е. системы для разработки новых программ.

Языки программирования

Язык программирования - это система команд, понятных ЭВМ.

Языки программирования делятся на:

- ***машинноориентированные языки*** (языки низкого уровня): Автокоды, Ассемблеры – позволяют управлять вычислительным процессом напрямую, при помощи машинных команд
- ***языки программирования высокого уровня*** - разработаны для удобства использования программистом.

Классы языков высокого уровня

- ***Процедурные*** – Fortran, Pascal, Basic, C;
- ***Логические*** – Лисп, Пролог и др.;
- ***Объектно-ориентированные*** – C++, Java, Delphi и др.

Классы языков высокого уровня

Среди программистов, пишущих программы для персональных компьютеров, наибольшей популярностью пользуются языки **Си**, **Си++**, **Бейсик** и **Паскаль**.

Язык СИ

Язык **Си** был изобретен в 1972 году **Денисом Ричи** и **Кеном Томпсоном**

Цель - для использования в написании весьма ныне популярной операционной системы Unix.

Си соединяет свойства языка высокого уровня с возможностью эффективного использования ресурсов компьютера, которое обычно обеспечивается только при программировании на языке Ассемблера.

Си не очень прост в обучении и требует тщательности в программировании, но позволяет писать сложные и весьма высокоэффективные программы.

Язык Бейсик

Язык Бейсик был создан в 1964 году **Томасом Куртом** и **Джоном Кемени** как язык для начинающих, облегчающий написание простых программ. Существует много различных версий Бейсика. Это язык очень широко распространен на микрокомпьютерах. На IBM PC широко используются Quick Basic и Visual Basic фирмы Microsoft и Turbo Basic фирмы Borland.

Основная идея авторов языка Бейсик - снабдить простым языком программирования непрофессиональных программистов - оказалась очень привлекательной для большого числа разработчиков различных фирм.

В феврале 1975 года на рынке программных средств появилась первая версия языка Бейсик для персональных компьютеров, авторами которой были Билл Гейтс и Пол Аллен, сотрудники корпорации Microsoft - ведущего лидера поставщиков программного обеспечения.

Именно этот год стал, по существу, годом, когда Basic вышел в свет. Сочетая в себе простоту, гибкость и универсальность, этот язык стал прообразом многих других языков программирования.

Язык Паскаль

Язык **Паскаль** был разработан в 1970 году **Никлаусом Виртом** как язык для обучения программированию. Паскаль позволяет писать программы, легко читаемые даже новичком, и содержит в себе все элементы, необходимые для соблюдения хорошего строгого стиля программирования (называемого структурным программированием), упрощающего разработку сложных программ. Системы программирования на Паскале для IBM PC также реализуют расширенные варианты этого языка. Из этих реализаций наиболее популярны - Turbo Pascal, Borland Pascal и Borland Pascal for Windows

Простейшая программа

Заголовок программы

```
program primer;
```

```
Var;
```

Раздел описания переменных

```
begin { начало программы }
```

```
. . .
```

Тело программы

```
end. { конец программы }
```

Комментарии в фигурных скобках
не обрабатываются

Вывод текста на экран

```
program primer;  
Begin  
Writeln('hello world!!!')  
end.
```

hello world!!!

Раздел описания

Переменная - именованный участок памяти компьютера для хранения данных определенного типа.

Значение переменной (информация в соответствующих ячейках памяти) в ходе выполнения программы может быть изменено. Каждая переменная должна быть уникальной и отличаться именем. Переменные в программе задаются в разделе описания между заголовком программы и ее основным текстом



Основные типы данных:

INTEGER – тип целых чисел (-32768..32767)

//LONGINT – тип «длинных» целых чисел
(-2147483648..2147483647)

REAL – тип действительных чисел (с дробной частью)

STRING – строковый тип

//CHAR – символьный тип

BOOLEAN – логический тип [TRUE, FALSE]

Имена переменных

В именах **МОЖНО** использовать

- латинские буквы (A-Z)

заглавные и строчные буквы не различаются

- цифры

имя не может начинаться с цифры

- знак подчеркивания _

В именах **НЕЛЬЗЯ** использовать

- русские буквы
- пробелы
- скобки, знаки +, =, !, ? и др.

Какие имена правильные??

AXby R&B 4Wheel Вася “PesBarbos”

TU154 [QuQu] _ABBA A+B

Пример описания переменных:

```
program qq;  
Var a,b: integer;  
    c: longint;  
    str: string;  
    temperature: real;  
    x: 1..10;  
    y: 'a'..'z';  
Begin  
...  
end.
```

Операторы ввода-вывода

В процессе работы программы переменные могут меняться как самой программой так и пользователем.

Для этого существуют специальные команды ввода данных в программу.

Команда присваивания используется для изменения значений переменных. В языке Паскаль эта команда обозначается так:

<имя переменной>:=<выражение>

```
Begin  
a := 8 ;  
b := 2 ;  
S := a + b ;  
End ;
```

При этом тип выражения должен совпадать с типом переменной, заданной в разделе описания

Операторы ввода-вывода

Команды вывода на экран текста или значения переменных
`WRITE ()`; - выводит на экран, указанные в скобках параметры, курсор остается в этой же строке.

`WRITELN ()`; - выводит на экран, указанные в скобках параметры, после вывода курсор переходит в начало следующей строки экрана.

В качестве параметров в круглых скобках может быть указан текст сообщения (текст записывается в апострофах `'`) и имя переменной (записывается без апострофов), значение которой нужно вывести. Между выводимыми элементами ставится запятая.

```
Begin
```

```
a:=8; b:=2;
```

```
Writeln('a = ',a, ' b = ',b);
```

```
End;
```

```
A = 8 b = 2
```

Операторы ввода-вывода

`READ();` - считывает в перечисленные в скобках переменные значения, введенные с клавиатуры через пробел. В конце ввода следует нажать утук

`READLN();` - переменные вводятся через Enter. Если значения вводились через пробел, то после ввода следует нажать Enter.

В качестве параметров в круглых скобках через запятую указываются имена переменных, значения которых считываются.

При работе с простыми переменными рекомендуется использовать команду `Readln ();`

```
Begin
```

```
Writeln( 'введите a' );
```

```
Readln(a) ;
```

```
End;
```

```
введите a  
8
```

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?

Переменные

```
program summ;  
var a, b, c: integer;  
begin  
    writeln('Введите два целых числа');  
    read ( a, b );  
    c := a + b;  
    writeln ( a, '+', b, '=', c );  
end.
```

**Презентация
окончена**