Программирование

Введение

Программирование и программисты

Программирование – это создание программ для компьютеров. Людей, которые этим занимаются, называют **программистами**.

Программист должен уметь:

- *анализировать* поставленную задачу:;
- разрабатывать алгоритм решения;
- писать тексты программ на различных языках программирования;
- отлаживать и тестировать программы;
- готовить описания программ и инструкции для пользователей;
- дорабатывать и сопровождать программы после сдачи заказчику.

Первая программа

Программа на языке Паскаль

```
program qq;
begin
  { основная программа }
end.
```

Оператор — это команда языка программирования.

Комментарии — это пояснения для человека внутри текста программы.

Вывод текста

Программа на языке Паскаль

```
program Hello;
begin
  write('Привет!')
end.
```

Команда **вывод** (*оператор вывода*) в алгоритмическом языке выводит на экран символы, заключенные в апострофы. В языке Паскаль для вывода данных используют стандартную процедуру **write**.

Вывод текста

В программе на языке Паскаль каждая команда (оператор) заканчиваются точкой с запятой, но перед словом **end** её можно не ставить.

При использовании оператора ***Writeln*** после вывода на экран осуществляется переход на следующую строку, при использовании оператора ***Write*** перехода на следующую строку не происходит.

Примеры вывода данных

Например,

1. **Write(a)** – вывод на экран значения переменной (или константы) a;
2. **Write('a')** – вывод на экран буквы **a**;
3. **Write('a', b, c)** – вывод на экран буквы **a** и значений переменных (или констант) **b** и **c**;
4. **Write('a,', b, ',', c)** – вывод на экран буквы **a** и значений **b** и **c** через запятую;
5. **Write('a');** – вывод на экран букв **Write('b') a** и **b** (в строку)
6. **Writeln('a');** – вывод на экран букв **Write('b') a** и **b** (в столбик)

© К.Ю. Поляков, Е.А. Ерёмин, 2013

Системы программирования

Транслятор – это программа, которая переводит тексты других программ в машинные коды (команды процессора).

Трансляторы бывают двух типов:

- ***компиляторы***, которые переводят в машинные коды сразу всю программу и строят исполняемый файл;
- ***интерпретаторы***, которые выполняют программу по частям: обработав очередной фрагмент программы, интерпретатор сразу исполняет его.

Системы программирования

Отладчик – программа для поиска ошибок в разрабатываемых программах.

Отладчик позволяет:

- выполнять программу в пошаговом режиме (по одной строчке);
- просматривать значения переменных в памяти;
- устанавливать *точки останова*, то есть отмечать места в программе, в которых выполнение программы временно приостанавливается.

Программирование

Линейные программы

© К.Ю. Поляков, Е.А. Ерёмин, 2013

Переменные

Если команды в программе будут выполняться последовательно, одна за другой, то такие алгоритмы (и программы) называются ***линейными.***

Программа по сложению двух чисел должна:

1. запросить у пользователя два целых числа;
2. сложить их;
3. вывести результат сложения.

Переменные

Исходные данные (числа), которые введёт человек, нужно сохранить в памяти компьютера. Для этого используют *переменные*.

Переменная — это величина, которая имеет имя, тип и значение. Значение переменной может изменяться во время выполнения программы.

Типы переменных

Существуют *стандартные* и *пользовательские* (создаваемые самим программистом) типы.
Некоторые стандартные типы:
Integer – целые числа(-10; 0; 1; 2; …)
Real – вещественные числа (-10; -7,241; 0; 1; 4,25; …)
Boolean – логический тип (**true** и **false**)
Char – символьный тип (’а’; ’б’; ’1’; ’*’ …)
String – строковый тип (’а’; ’б’; ’1’; ’123’; ’абв’; ’*’ …)

Например,
Var x:integer; Y:char; a:integer; b:integer;
Переменные одного типа можно описывать вместе:
Var x,a,b:integer; Y:char;

Переменные

Переменная может хранить только одно значение. При записи в неё нового значения «старое» стирается и его уже никак не восстановить.

Переменные в программе необходимо объявлять. При объявлении указывается тип переменной и ее имя.

Идентификатор — это имя программы или переменной.

Идентификаторы

Идентификаторы задает сам программист
- В качестве идентификатора может использоваться любая ***последовательность латинских букв, цифр и знака подчеркивания, начинающаяся с буквы***
- Имена идентификаторов не могут совпадать
- Строчные и прописные буквы не различаются

Определите могут ли быть идентификаторами следующие последовательности символов:
1) prog 1 2) prog_1 3) prog1 4) Prog 5) прог1
6) 1_prog 7) prog~1 8) PROG1 9) PrOg 10) p

Переменные

Вот так объявляются целочисленные переменные (*в которых могут храниться только целые значения*) с именами **a**, **b** и **c**:

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

В языке Паскаль описание переменных начинается с ключевого слова **var**, после него записывают список переменных и в конце через двоеточие – их тип.

Можно использовать в именах латинские буквы (строчные и заглавные буквы не различаются), цифры и знак подчеркивания «_». Имя не может начинаться с цифры.

Переменные

Тип переменной нужен для того, чтобы определить:

- какие значения может принимать переменная;
- какие операции можно выполнять с этой переменной;
- сколько памяти нужно выделить для хранения значения.

© К.Ю. Поляков, Е.А. Ерёмин, 2013 http://kpolyakov.spb.ru

Работа с переменными

Присвоить значение переменной можно двумя способами. Во-первых, можно записать нужное значение прямо в программе:

```
a:= 5;
```

Оператор, содержащий символы «**:=**» – это **оператор присваивания**, с его помощью присваивают новое значение переменной.

Для вывода значения переменной на экран используют тот же оператор **вывод** (в Паскале –**write**), который раньше применяли для вывода текста

Работа с переменными

Очень часто программа хранится как *исполняемый файл* – готовые к выполнению машинные коды.

В этом случае текст программы нам недоступен и в нём ничего нельзя исправить. Для того чтобы пользователь смог как-то поменять исходные данные, программист может предусмотреть их ***ввод с клавиатуры***

Работа с переменными

Для ввода данных с клавиатуры используется оператор **ввод** (в Паскале – **read**). Например:

```
read(a);
```

Выполняя эту команду, компьютер ожидает, пока пользователь наберёт значение и введёт его, нажав клавишу *Enter*. Это значение будет присвоено переменной *a*, которая указана в операторе ввода.

Работа с переменными

Полная программа сложения двух чисел:

```
program Sum;
var a, b, c: integer;
begin
  read(a, b);
  c := a + b;
  write(c);
end.
```

У этой программы есть два недостатка:

- перед вводом данных пользователь не знает, что от него требуется
- результат выдается в виде числа, которое означает неизвестно что.

Работа с переменными

Хотелось бы, чтобы диалог программы с пользователем выглядел так:

Введите два числа: 2 3

2+3=5

```
write(a,'+',b,'=',c);
```

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

Арифметические выражения

Арифметические выражения обычно записываются в одну строчку. Они могут содержать константы, имена переменных, знаки арифметических операций, круглые скобки.

$$a \leftarrow \frac{c+b-1}{2} \cdot d$$

```
a:=(c+b-1)/2*d;
```

Арифметические выражения

Арифметические действия выполняются в следующем порядке:

- действия в скобках;
- умножение и деление, слева направо;
- сложение и вычитание, слева направо.

Операция умножения обозначается знаком «*», а операция деления – знаком «/».

Арифметические выражения

Результат деления (операции «/») может быть нецелым числом, такие числа называются **вещественными.** Вещественное значение нельзя записать в целочисленную переменную. Вещественную переменную *x* объявляют так:

```
var x: real;
```