

Лабораторная работа №02 по дисциплине "Программирование"

ТЕМА: Основные понятия языка программирования. Создание простых программ.

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПЕК -----

1. В Вашей папке создайте папку с именем П_лр02, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.
2. При выполнении задания создавайте в папке П_лр02 для каждой программы папку с именем, по смыслу соответствующим действиям, выполняемым составляемой программой, и сохраняйте все файлы, имеющие отношение к этой программе, в этой созданной папке (имена сохраняемых файлов должны отличаться от имен, присваиваемых по умолчанию, и должны по смыслу соответствовать действиям, выполняемым составляемой программой).
3. ПРИ ВЫПОЛНЕНИИ СЛЕДУЮЩИХ ЛАБОРАТОРНЫХ РАБОТ ДЕЙСТВУЙТЕ АНАЛОГИЧНО ПРЕДЫДУЩЕМУ.

АЛФАВИТ ЯЗЫКА ПРОГРАММИРОВАНИЯ -----

4. В алфавит языка программирования входят:
цифры "0 1 2 3 4 5 6 7 8 9";
большие и маленькие латинские буквы
"A B C D E F G H I J K L M N O P R S T U V W X Y Z
a b c d e f g h i j k l m n o p r s t u v w x y z"
и символ подчеркивания "_";
специальные символы "+ - * / = , . : ; ' < > () [] { } ^ @ \$ #"
и символ "пробел".
Символ ";" используется при записи программы для разделения операторов, объявлений и описаний.

ИДЕНТИФИКАТОРЫ -----

5. Идентификаторы предназначены для обозначения переменных, функций и других объектов в программе.
Идентификатор может состоять из букв, цифр и символов подчеркивания "_", но **начинаться он должен только с буквы или символа подчеркивания "_"**.
Большие и маленькие буквы в идентификаторах не различаются.
Между двумя идентификаторами в программе должен быть хотя бы один **разделитель** (пробел, символ конца строки, комментарий или другой специальный символ или комбинация специальных символов).
Идентификаторы различаются по их первым 63 символам.
6. Примеры идентификаторов: a A x bc ff vv1 v_W summa DELTA2 alfa5

ПЕРЕМЕННЫЕ И КОНСТАНТЫ -----

7. Переменная – это область памяти, которая отводится для данных, используемых программой.
Переменная в программе обозначается идентификатором.
Константа – это целое или дробное число, строка символов или отдельный

СИМВОЛ.

8. Примеры констант: числовых - 123 0.1 0.0 -5.03 0 1 1500 1.5e3 1.5E3
символьных - '***Result: ' 's='

ТИПЫ ДАННЫХ

9. Пока будем использовать следующие идентификаторы для обозначения типов и соответствующие типы данных:
Integer - целый тип;
Real - вещественный тип;
String - строковый тип.

ОПИСАНИЕ (ОБЪЯВЛЕНИЕ) ПЕРЕМЕННЫХ

10. Описание (объявление) переменных начинается с ключевого слова **var** и **обязательно должно предшествовать использованию этих переменных** в программе.
11. Примеры описания (объявления) переменных:
var s,i: Integer;
x,y: Real;
st: String;

ОПЕРАЦИИ

12. Математические операции:

Операция	Действие	Типы операндов	Тип результата	Приоритет
+	Сложение	Хотя бы один из операндов real	real	Более низкий
		Оба операнда integer	integer	
-	Вычитание	Те же, что для +	Тот же, что для +	Более высокий
*	Умножение	Те же, что для +	Тот же, что для +	
/	Деление	real или integer	real	
div	Деление нацело	integer	integer	
mod	Остаток от деления			

СТАНДАРТНЫЕ ФУНКЦИИ

13. Некоторые математические функции:

Функция	Описание	Тип	
		аргумента	результата
Abs(x)	Абсолютное значение x	real integer	real integer
Sqrt(x)	Квадратный корень из x	real integer	real
Sqr(x)	Квадрат x	real integer	real integer
Sin(x)	Синус x	real integer	real
Cos(x)	Косинус x	real integer	real
Arctan(x)	Арктангенс x	real integer	real
Pi	Число π	-	real
Exp(x)	Экспонента x	real integer	real
Ln(x)	Натуральный логарифм x	real integer	real

14. Некоторые функции преобразования:

Функция	Описание	Тип	
		аргумента	результата
Int(x)	Целая часть x	real	real
Frac(x)	Дробная часть x	real	real
Trunc(x)	Отбрасывает дробную часть x	real	integer
Round(x)	Округление x до ближайшего целого числа по общепринятым правилам	real	integer

ВЫРАЖЕНИЯ

15. Для вычисления значений по формулам применяются выражения, состоящие из операндов (переменных, констант, функций), связанных между собой знаками математических операций.

При записи выражений можно не ставить пробел между операндами и знаками операций (за исключением операций **div** и **mod**).

При вычислении значений выражений в первую очередь вычисляются значения функций, затем операции умножения и деления, затем операции сложения и вычитания.

Порядок действий в выражениях может быть изменен с помощью круглых скобок, при этом операции в скобках будут выполняться в обычном порядке, но раньше, чем операции за скобками.

При записи выражений, содержащих скобки, число открывающих скобок должно быть равно числу закрывающих скобок.

Тип выражения определяется типами входящих в него операндов и выполняемыми в нем операциями.

16. Примеры выражений:

```
x+1
y+sin(abs(x)*Pi/180)-1
I mod 2+1
-x+y/z
(a-b)/(c+d/(e+f))
```

ОПЕРАТОР (ИНСТРУКЦИЯ) ПРИСВАИВАНИЯ

17. Примеры оператора присваивания:

```
z:=x+1;
f:=y+sin(abs(x)*Pi/180)-1;
k:=I mod 2+1;
uu := -x+y/z1z2;
t:=(a-b)/(c+d/(e+f));
s:=s+1;
```

18. При выполнении оператора присваивания сначала вычисляется значение выражения справа от знака присваивания " :=" и затем результат присваивается переменной слева от знака " :=". Переменная и выражение должны иметь совместимые типы (например, real и integer, но не наоборот). Пробелы внутри знака присваивания недопустимы.

ОПЕРАТОРЫ (ИНСТРУКЦИИ) ВВОДА И ВЫВОДА В РЕЖИМЕ КОНСОЛЬНОГО ПРИЛОЖЕНИЯ

19. При выполнении, например, оператора ввода Readln(x1,x2,x3) программа ждет, пока на клавиатуре будут набраны три числа, разделенные пробелом, табуляцией или нажатиями клавиши Enter, и затем, после ввода последнего из этих чисел и нажатия на клавишу Enter, введенные значения присваиваются переменным, имена которых указаны в операторе ввода.

Если было введено больше чисел, чем указано переменных в операторе ввода, то остальные числа будут потеряны.

Для того, **чтобы окно с результатами выполнения консольного приложения неожиданно не исчезло с экрана**, рекомендуется в конце программы использовать оператор Readln **без аргументов** (этот оператор будет ждать нажатия на клавишу Enter).

Во избежание недоразумений при вводе данных не рекомендуется использовать оператор Read, который передает излишне введенные данные следующему оператору ввода.

20. При выполнении, например, оператора Write('*Results: ',s,' ',i,' ',j,' ',k), на экран будут выведены в одной строке текст *Results: и значения переменных s,i,j,k, разделенные пробелами, после чего курсор останется в этой же строке после последнего выведенного символа значения переменной k.

В операторе Write можно использовать выражения, например, Write(I mod 2+1,' Argument ',x,y+sin(abs(x)*Pi/180)-1).

Оператор Writeln, в отличие от оператора Write, после вывода значений выполняет переход на начало следующей строки.

Оператор Writeln без аргументов выполняет переход в начало следующей строки.

КОММЕНТАРИИ

21. Комментарий – это пояснительный текст, который можно записывать в любом месте программы, где разрешен пробел. Наличие или отсутствие комментариев никак не влияет на работу программы. Текст комментария может быть ограничен символами { и } или их эквивалентами (* и *) и может содержать любые символы, включая русские буквы. Комментарий, ограниченный указанными символами, может размещаться в нескольких строках. Однострочный комментарий должен содержать символы // в начале места расположения комментария в строке.

22. Примеры комментариев:

```
{ Ниже находится оператор,  
увеличивающий значение s на 1 }  
s:=s+1; // Здесь значение s увеличивается на 1  
// Ниже находится еще один комментарий  
(* Выше находится оператор, увеличивающий значение s на 1 *)
```

ПРИМЕР ТЕКСТА ПРОГРАММЫ

```
program Primer;  
  
{ $APPTYPE CONSOLE }  
  
uses  
  SysUtils;  
  
var a,b,d: integer; // Объявление переменных  
      x,y: real;  
begin  
  { TODO -oUser -cConsole Main : Insert code here }  
  readln (a,b); { Ввод чисел }  
  d:=a div b; // Целая часть от деления  
  writeln(a, ' ',b, ' ', 'div ',d); (* Вывод результатов*)  
  readln (x); // Ввод x  
  y:=sin(abs(x)*Pi/180)-1; //Вычисление y  
  writeln(x, ' ',abs(x)*Pi, ' ',y); { Вывод x,  
                                     произведения модуля x на число "пи",  
                                     y }  
  readln; //Ожидание нажатия на клавишу Enter  
end.
```

СОЗДАНИЕ ПРОСТЕЙШИХ ПРОГРАММ

ПРОГРАММА 2-1

23. Создайте консольное приложение, при выполнении которого происходит:

- ввод двух целых чисел;
- вычисление целой части и остатка от деления первого из введенных чисел на второе;
- вывод в следующей строке слева направо значений введенных чисел и полученных результатов вычислений с расположенным левее каждого результата названием соответствующей операции (DIV и MOD);
- в строках текста программы с операторами, выполняющими вычисления, должны также находиться комментарии на русском языке с названиями действий, выполняемыми соответствующими операциями.

ПРОГРАММА 2-2

24. Создайте консольное приложение, при выполнении которого происходит:
- а) вывод текста "X= " и затем ввод в этой же строке вещественного аргумента;
 - б) вычисление и печать в отдельных строках значений всех перечисленных в пп.13,14 функций от введенного аргумента с расположенным левее каждого результата названием соответствующей функции (кроме, разумеется функции Pi);
 - в) для функций $\text{Sqrt}(x)$ и $\text{Ln}(x)$ следует вычислять значения $\text{Sqrt}(\text{Abs}(x))$ и $\text{Ln}(\text{Abs}(x))$;
 - г) в тексте программы перед строками с соответствующими операторами должны быть вставлены строки с комментариями "Математические функции" и "Функции преобразования".

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

25. Продемонстрируйте тексты созданных программ, а также их выполнение в среде программирования и независимо от нее.
-