****

***Algoritmos y Programación I***

***Laboratorio #1***

**Introducción al Lenguaje Pascal**

**Fecha de envío:** Viernes 12-04-2018 **Fecha tope de entrega:** Miércoles 17-04-2019 11:59 pm

**Objetivo**

* Introducción al lenguaje Pascal y sus características.
* Conocer el entorno de programación.
* Identificar las partes de un programa en Pascal
* Conocer las características de los tipos de datos elementales predefinidos en Pascal.
* Saber evaluar expresiones simples (tanto aritméticas como lógicas) correctamente.
* Conocer y aprender a utilizar las instrucciones básicas en Pascal.
* Apreciar la necesidad de la documentación exhaustiva de los programas que se implementen, conociendo algunos criterios adicionales de documentación.
* Aprender a construir, en Pascal, correctamente la solución de algunos problemas simples.

**Pre-laboratorio**

**Leer el siguiente contenido:**

Pascal es un lenguaje de programación de alto nivel, y de propósito general desarrollado por el profesor suizo Niklaus Wirth entre los años 1968 y 1969 y publicado en 1970. Su objetivo era crear un lenguaje que facilitara el aprendizaje de programación a sus alumnos, utilizando la programación estructurada. Sin embargo con el tiempo su utilización excedió el ámbito académico para convertirse en una herramienta para la creación de aplicaciones de todo tipo.

Pascal se caracteriza por ser un lenguaje de programación estructurado fuertemente tipificado. Esto implica que:

1. Utiliza la técnica de programación estructurada

2. El tipo de dato de todas las variables debe ser declarado previamente para que su uso quede habilitado.

Partes de un programa Todo programa en Pascal cuenta con tres partes:

|  |
| --- |
| **program identificador; Cabecera** |
| **uses**  identificadores  **label**  lista de etiquetas  **const**  definición de constantes  **type Declaraciones**  declaración de tipos de datos definidos por el usuario  **var**  declaración de variables  **Procedure**  definición de procedimientos  **function**  definición de funciones |
| **begin**  sentencias/Instrucciones; **Cuerpo**  **end.** |

**I.- Cabecera:**

La cabecera del programa únicamente lleva una sola instrucción donde se indica el nombre del programa.

PROGRAM nombre del programa;

**II.- Declaraciones:**

En la sección de declaraciones se le indica al compilador todos los identificadores y unidades que se utilizarán durante la ejecución del programa.

**Declaración de unidades**

Las unidades o librerías son módulos independientes del programa que, aunque no son ejecutables por sí mismos, pueden ser utilizados por el programa principal sin necesidad de reescribir el código que contienen. Tiene funciones para trabajar con archivos, memoria, directorios, servicios del DOS, servicios de video o pantalla, etc. Para la utilización de estos "subprogramas o librerías" es necesaria su declaración.

La palabra reservada **USES** cumple el propósito de declarar las unidades en el formato siguiente:

USES

**crt, dos;**

Esta línea declara y habilita para su uso a la unidad **crt** y a la unidad **dos**.

Cada unidad que se declara deberá estar separada de la siguiente por una coma. Al final de todas las unidades

declaradas se deberá colocar un punto y coma ";".

Algunas de las librerías de Pascal son:

* **Crt (catode ray tube)**

Esta unidad provee rutinas de control sobre el modo de la pantalla, códigos extendidos de teclado, colores, ventanas y sonidos.

* **Graph**

Esta es una de las más divertidas unidades, pues con la diversidad de subprogramas que provee, sumados a unos sencillos algoritmos, permite hacer desde una simple línea o un círculo hasta una simulación.

* **Dos**

Esta unidad nos brinda las rutinas necesarias para interactuar con el sistema operativo D.O.S. permitiendo tomar desde la hora o fecha del sistema hasta ejecutar programas o comandos.

* **Printer**

Permite imprimir líneas y archivos de tipo texto.

**Declaración de Constantes y Variables**

Para declarar las constantes se utiliza la palabra reservada **CONST** seguida de un identificador al que se le

dará un valor determinado, un signo de igual "=", el valor que recibirá el identificador y al final un punto y

coma ";".

CONST

pi = 3.1415926;

iva = 12;

Descuento = 5.0;

En la declaración de variables utilizamos la palabra **VAR** seguido por el identificador de la variable seguido

de dos puntos y el tipo de dato. Ejemplo:

VAR

Identificador: Integer;

Identificador: String;

Identificadores

**Los identificadores** son palabras que representan constantes, variables, tipos de datos, procedimientos, funciones y algunos otros datos. Existen dos tipos de identificadores: los predefinidos en Pascal y los definidos por el programador.

Algunos de los identificadores predefinidos en Pascal, son: integer, real, byte que son **Palabras Reservadas** y no pueden usarse como nombres de variables, programas, constantes.

Por su parte, los identificadores definidos por el programador son los elementos del lenguaje que se usan para nombrar el programa, las variables, los procedimientos, las funciones, etc.

**Para crear estos identificadores, por parte del programador, se deben seguir estas reglas :**

* Debe comenzar con una letra (A a Z, mayúsculas o minúsculas) y no puede contener blanco.
* Las letras , números y el guión bajo (\_), están permitidos después del primer carácter.
* Un identificador puede tener cualquier longitud, pero solo se tomarán en cuenta los primeros 63 caracteres, no debe contener espacios ni caracteres especiales como : &, !, \*, etc., tampoco puede ser alguna palabra reservada, como : PROGRAM, BEGIN, INTEGER, ELSE, IF, CASE, FUNCTION, INTEGER etc.
* Para Pascal no existen diferencias entre mayúsculas y minúsculas, así que a un identificador denominado "valor" se le puede referir como "VALOR" o "VaLoR" o “valor”.
* Todo identificador en Pascal debe ser definido previamente a su utilización.

**III.- Cuerpo del programa:**

El cuerpo del programa empieza con la palabra "BEGIN" y finaliza con "END."; entre ambas se sitúan las instrucciones del programa.

**Instrucciones de asignación**

La instrucción de asignación es una sentencia que permite reemplazar el valor de una variable durante la ejecución de un programa. Es una operación destructiva, esto es, el nuevo valor sustituye al anterior. Su sintaxis es la siguiente:

**identificador := expresión;**

Se evalúa la expresión de la derecha y el resultado se "*asigna*" a la variable que aparece a la izquierda del "**:=**".

Ha de existir una consistencia de tipos entre la expresión y la variable a la que se le asigna.

**Instrucciones de entrada/salida (e/s).**

Estas instrucciones permiten comunicar al entorno (usuario) con el programa tanto para introducirle (entrada) datos como para obtener (salida) resultados.

**Instrucciones de entrada.**

READ y READLN, son las dos instrucciones que vamos a utilizar para realizar la entrada de los datos. Estas instrucciones leen los datos del dispositivo de entrada (input) y los asigna a las variables que tengan como argumentos.

Su sintaxis es la siguiente:

READ(v1,v2,...., vn)

READLN(v1,v2,...., vn)

La instrucción readln requiere que se introduzca por teclado un salto de línea y la siguiente instrucción readln ignora todo lo que se haya escrito anteriormente.

Diferencia entre READ y READLN-

En READ , después de pulsar la tecla ENTER, el cursor permanece inmediatamente después del último carácter introducido. Con READLN, el cursor se envía al principio de la siguiente línea, tas pulsar la tecla ENTER.

**Instrucciones de salida**

WRITE y WRITELN escriben valores a un fichero de salida, en nuestro caso la pantalla (fichero output). Una instrucción de salida puede imprimir:

* Comentarios o cadenas de caracteres entre apóstrofes. Por ejemplo Write(‘hola’)
* Constantes, valores de variables, resultados de expresiones y funciones.

WRITELN obliga a la pantalla (dispositvo de salida) a pasar a la línea siguiente después de realizar la impresión de los argumentos que tiene entre paréntesis.

La diferencia entre WRITE y WRITELN , es que el cursor pasa al comienzo de la siguiente línea después de ejecutar WRITELN, mientras que con WRITE el cursor se queda al fina de la línea.

**Laboratorio**

En el laboratorio vamos a trabajar con un en entorno de desarrollo integrado llamado DEV-Pascal el cual está disponible para trabajar con el Sistema Operativo Windows. El mismo lo puede descargar en el siguiente link:

**http://www.bloodshed.net/devpascal.html**

**Actividad 1:**

Ingrese en el entorno escogido por usted y familiarícese con el entorno de desarrollo, e investigue lo siguiente:

* Las opciones del menú que tiene disponible.
* Los tipos de datos que soporta Pascal
* Los tipos de operadores para los distintos tipos de datos que utiliza
* Cómo escribir comentarios en los programas.

**Actividad 2:**

1.- Dado el nombre completo, cédula de identidad, dirección, fecha de nacimiento y tres calificaciones de un estudiante, hacer un programa en Pascal que determine el promedio de las tres calificaciones y emita la siguiente salida:

Reporte de Notas

Bachiller: xxxxxxxxxxxxxxxx Cédula : Vxxxxxxxxx

Nacido el: xx/xx/xx y residenciado en : xxxxxxxxxxxxxxxxxxxx.

Promedio final de notas : xx,xx puntos

2.- Desarrolla un programa que permita intercambiar el valor se dos números.

**Post Laboratorio**

Asignación para ser enviada antes del Miércoles 17-04-2019 a las 11:59 pm.

**Implementar los siguientes programas en Pascal (5 ptos c/u).**

1.- Escribir un programa en Pascal que: sume, reste, multiplique, divida, calcule el resto y el cociente de dos números dados por teclado. Para el caso de la división, resto y cociente, el primer número es el dividendo y el segundo es el divisor.

2.- Escribir un programa en Pascal que calcule el área y el volumen de un cilindro, a partir de las siguientes fórmulas:

A = (2\* (pi + **r2**)) + ((2 + pi + r) \* h)

V = (pi \* **r2**) \* h

3.-Escribir un programa en Pascal que calcule la siguiente expresión:

**X1= (-b + √(b2-4\*a\*c)/(2\*a)**

Las cual constituye la raíz real positiva de una ecuación de segundo grado.

4.- Dado el nombre de un trabajador, el sexo (M/F), fecha de ingreso, sueldo mensual y el porcentaje de retención del I.S.L.R, calcule el monto neto a cobrar, una vez deducido el impuesto. El programa debe mostrar la siguiente salida:

RETENCION DE I.S.L.R

Nombre del trabajador: xxxxxxxxxxxxxxxxxxx Sexo : X

Sueldo mensual (Bs) : xxxxxx,xx Porcentaje retención : xx,xx%

Impuesto retenido (Bs) : xxxxxx,xx

Total a cobrar : xxxxxx,xx .

**Instrucciones para el envío de la tarea**:

Crear una carpeta comprimida que incluya:

* Un archivo .doc con los algoritmos de los problemas propuesto en el Postlaboratorio
* Los archivos .pas de los problemas resuelto (un archivo .pas por cada programa)
* Un archivo .doc con las capturas de pantalla de las corridas de los programas.

Además debe enviar la HOJA DE CONTROL DE PRACTICAS con la autoevaluación de la práctica de la semana. (NO INCLUYA ESTA HOJA EN LA CARPETA COMPRIMIDA)

Adjuntar archivos a la tarea correspondiente en Módulo 7