|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Rodríguez Espino Claudia |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1122, Bloque: 139 |
| *No de Práctica(s):* | Quinta práctica |
| *Integrante(s):* | Fonseca Ramírez Gadiel Moisés |
|  |  |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | 14/Septiembre/2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Guía práctica de estudio 05: Pseudocódigo**

**Objetivo:**

Elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

**Actividades:**

* Elaborar un pseudocódigo que represente la solución algorítmica de un problema en el cual requiera el uso de la estructura de control de flujo condicional.
* A través de un pseudocódigo, representar la solución algorítmica de un problema en el cual requiera el uso de la estructura de control iterativa.

**Actividades de clase:**

* Al leer la práctica se comenzó con el tema de pseudocódigo, el cual es una representación algorítmica de la solución a un problema, la cual se hace después del algoritmo, puesto que el pseudocódigo es una codificación del algoritmo, la cual es más cercana a los lenguajes de programación.
* Después la profesora explicó la correcta sintaxis que debe tener el pseudocódigo, para poder realizar distintos procesos y tareas de un programa. Estos procesos son iguales a los que ya se vieron en la clase de teoría y son representables a través de los diagramas de flujo.
* Todo pseudocódigo tiene un inicio y un fin, al igual que los algoritmos y diagramas de flujo, y entre estos se deben escribir las instrucciones a realizar. Las palabras reservadas en el pseudocódigo se escriben en mayúsculas, se debe usar sangría para que el código tenga orden y sea fácilmente entendible y depurable.
* Luego la profesora fue explicando de la práctica cada una de las siguientes instrucciones:
  + LEER: indica que el usuario ingrese información.
  + ESCRIBIR: escribe información en la pantalla (en el algoritmo lo llaman imprimir).
  + La variable se declaran como, nombre de variable: tipo de dato

Los tipos de datos van en mayúsculas y pueden ser:

* + - ENTERO valor entero positivo y/o negativo
    - REAL valor con punto flotante y signo
    - BOOLEANO valor de dos estados: verdadero o falso
    - CARÁCTER valor tipo carácter
    - CADENA para cadenas de caracteres
  + Para declarar más de una variable con el mismo nombre se escribe nombre de variable[n]: tipo de dato, dónde ‘n’ es el número de variables.
  + También se pueden crear variables tipo “registros o estructuras”, que son tipos de datos que contienen varias variables de distintos tipos. Estos se hacen de la forma:

nombreRegistro:REG

var1: tipo de dato

var2: tipo de dato

var3: tipo de dato

FIN REG

* + Las variables constantes se escriben con mayúsculas y se deben inicializar al momento de declararlas, como:

Nombre de variable := n: tipo de dato, CONST, dónde ‘n’ es el calor de la constante.

* En el pseudocódigo se utilizan operadores aritméticos como suma (+), resta (-), multiplicación (\*), división real (/), división entera (div), módulo (mod), exponenciación (^), asignación (:=) y lógicos igualdad (=), y-lógica o AND (&), o-lógica u OR (|), negación o NOT (!), relaciones de orden (<, >, <=, >=) y diferente (<>).
* Las variables y funciones se declaran con la “notación de camello”, en la forma lower y upper, las palabras se separan con la inicial de cada palabra en mayúsculas.

Ejemplo de variables:

realAreaDelTriangulo: REAL // lower camel case

EnteroRadioCirculo: REAL // upper camel case

Ejemplo de funciones:

calcularArea()

obtenerPerimetro()

* Se dio la explicación de las estructuras de control, que se dividen en:
  + Secuencial: Se refiere a que las instrucciones se realizan una a continuación de la otra en el orden que están escritas.

INICIO

Instrucción 1

Instrucción 2

Instrucción 3

FIN

* + Condicionales (o selectivas): En estas primero evalúa una expresión lógica, y realizan una u otra instrucción dependiendo de si la condición es verdadera o falsa. En estas estructuras, sólo se realiza una u otra instrucción.

Estas instrucciones pueden ser de la forma SI, en la que si se cumple, se realiza un conjunto de instrucciones, si no se continúa con el programa normalmente:

SI condición ENTONCES

Instrucciones

FIN SI

O de la forma SI-DE LO CONTRARIO: En esta estructura, cuando la condición es falsa o verdadera, en ambos casos se realiza un conjunto de instrucciones

SI condición ENTONCES

Instrucciones

FIN SI

DE LO CONTRARIO

Instrucciones

FIN DE LO CONTRARIO

Además existe la estructura de tipo SLECCIONAR-CASO la cual funciona evaluando el valor de una variable y dependiendo de su valor, realiza un conjunto de instrucciones, esta estructura puede elegir una de varias opciones, y se escribe:

SELECCIONAR (variable) EN

CASO valor1 Instrucciones

CASO valor2 Instrucciones

CASO valor3 Instrucciones

DEFECTO Instrucciones

FIN SELECCIONAR

* Estructuras Iterativas o repetitivas (o cíclicas): Estas permiten realizar una serie de instrucciones mientras se siga cumpliendo una expresión lógica, tienen un inicio y un fin, y son de dos formas:
  + Estructura MIENTRAS, esta primero valida la condición y si es verdadera, ejecuta un conjunto de instrucciones, en caso contrario rompe el ciclo y sigue con el flujo del programa.

MIENTRAS condición ENTONCES

Instrucciones

FIN MIENTRAS

* Estructura HACER-MIENTRAS, es igual que la anterior, pero en esta primero se realizan las instrucciones y luego se valida la expresión lógica, se escribe como:

HACER

Instrucciones

MIENTRAS condición

* Las últimas instrucciones que se vieron fueron las “funciones”, las cuales son subprocesos (como pequeños programas) que generan datos que devuelven datos para el resto del programa. La función tiene un identificador (nombre), un conjunto de parámetros de entrada y un valor de retorno. La función puede recibir o no datos de entrada, y los datos que reciba la función van entre paréntesis. Pero toda función puede regresar un valor.

INICIO

FUNC identificador (var:TipoDato,..., var:TipoDato) RET: TipoDato

[Acciones]

FIN FUNC

FIN

* Para hacer comentarios o anotaciones en el pseudocódigo (como para decir de qué trata la instrucción del programa o qué hace) se usan dos asteriscos y rodo lo que vaya después de los dos asteriscos en la misma línea es un comentario, de la forma:

\*\*Indicaciones, explicaciones, ext.

* Para esta práctica se dejó como trabajo en casa hacer los programas de:

1. “La chicharronera” incluyendo la solución para números imaginarios.

INICIO

a, b, c, d, z, r, n, u1, u2: REAL

ESCRIBIR “Introduzca el valor de ‘a’ de su ecuación. El valor de ‘a’ no debe ser cero.”

LEER a

MIENTRAS a = 0 ENTONCES

ESCRIBIR “Error, ‘a’ debe ser distinto de cero”

ESCRIBIR “Introduzca el valor de ‘a’ de su ecuación.”

LEER a

FIN MIENTRAS

ESCRIBIR “Introduzca el valor de ‘b’ y ‘c’ de su ecuación”

LEER b, c

d := b\*b-4\*a\*c

SI d < 0 ENTONCES

z := (d\*-1)^1/2

r := -b/2\*a

n := z/2\*a

ESCRIBIR “x1 = r + n i”

ESCRIBIR “x2 = r - n i”

FIN SI

DE LO CONTRARIO

u1 := (-b+(d)^1/2)/2\*a

u2 := (-b-(d)^1/2)/2\*a

ESCRIBIR “x1 = u1”

ESCRIBIR “x2 = u2”

FIN DE LO CONTRARIO

FIN

1. Resolver si y<2; pero, si y>2 resolver

INICIO

i, u: REAL

ESCRIBIR “Introduce el valor de y”

LEER y

SELECCIONAR (y) EN

CASO y > 2

u = 4\*(i^2) – 3\*i+0

ESCRIBIR “ x = u”

CASO y < 2

u = i^2 + 4\*i - 25

ESCRIBIR “ x = u”

CASO y = 2

ESCRIBIR “No hay solución para y = 2

FIN SELECCIONAR

FIN

1. Pedir tres valores al usuario y decir si se puede armar un triángulo equilátero, isósceles o escaleno.

INICIO

a, b, c: REAL

ESCRIBIR “Introduzca el valor de los lados del triángulo.”

LEER a, b, c

MIENTRAS a = 0

ESCRIBIR “Ningún lado de un triángulo puede medir cero, introduzca el valor de a.”

LEER a

FIN MIENTRAS

MIENTRAS b = 0

ESCRIBIR “Ningún lado de un triángulo puede medir cero, introduzca el valor de b.”

LEER b

FIN MIENTRAS

MIENTRAS c = 0

ESCRIBIR “Ningún lado de un triángulo puede medir cero, introduzca el valor de c.”

LEER c

FIN MIENTRAS

SI a = b ENTONCES

SI b = c ENTONCES

ESCRIBIR “Es un triángulo equilátero”

FIN SI

FIN SI

DE LO CONTRARIO

SI a < > c ENTONCES

SI b < > c ENTONCES

ESCRIBIR “Es un triángulo escaleno”

FIN SI

FIN SI

FIN DE LO CONTRARIO

SI a = b ENTONCES

SI a < > c ENTONCES

SI b < > c ENTONCES

ESCRIBIR Es un triángulo isósceles”

FIN SI

FIN SI

FIN SI

SI b = c ENTONCES

SI b < > a ENTONCES

SI c < > a ENTONCES

ESCRIBIR Es un triángulo isósceles”

FIN SI

FIN SI

FIN SI

SI a = c ENTONCES

SI a < > b ENTONCES

SI c < > b ENTONCES

ESCRIBIR Es un triángulo isósceles”

FIN SI

FIN SI

FIN SI

FIN

1. Pedir tres valores al usuario y decir si la suma de los 2 primeros es igual al tercer número.

INICIO

a, b, c, d: REAL

ESCRIBIR “Introduzca tres valores”

LEER a, b, c

d := a + b

SI d = c ENTONCES

ESCRIBIR “La suma de los dos primeros valores es igual al tercer valor”

FIN SI

DE LO CONTRARIO

ESCRIBIR “La suma de los dos primeros valores es diferente del tercer valor”

FIN DE LO CONTRARIO

FIN

**Conclusión**

Esta práctica permitió poner en práctica la creación del pseudocódigo que se usa para representar la solución de un problema de una forma más cercana a la que utilizan los lenguajes de programación, para así facilitar la creación del programa en cualquier lenguaje, puesto que el pseudocódigo es independiente del lenguaje. Además de que trabajar el pseudocódigo en variados ejercicios permite aprender bastante acerca de cuándo se debe usar cada comando y cuáles son más convenientes para hacer el pseudocódigo de forma más eficiente y corta.