

PROGRAMACIÓN ORIENTADA A OBJETOS

1er. PARCIAL

Practica #2

Estructuras de Bucles en C#

Daniel Vázquez de la Rosa
d_vazquez@outlook.com

Instituto Politécnico Nacional



Escuela Superior de Cómputo

Agosto 2018

Datos Personales

Crear un Documento en Word que incluya una caratula que contenga tus datos. Dicho documento será guardado como: Apellido Paterno_NombreAlumno_Practica2_POO y deberá contener toda la información de la practica creada.

Objetivos de la Práctica

- Utilizar las estructuras de repetición para ejecutar instrucciones repetidas en una aplicación.
- Solucionar problemas con programas que integren la estructura de selección y estructuras repetitivas **for**.

Introducción

Los bucles o ciclos repiten una sentencia o una secuencia de sentencias un número de veces. A dichas sentencias se les conoce como **cuerpo del bucle**. A cada repetición del bucle se le llama **iteración**.

El bucle for

Ejecuta una sentencia o conjunto de sentencias un número finito de veces. En inicialización se escribe una variable de control de bucle (la cual actúa como un contador), con su valor inicial. La condición iteración contiene una expresión lógica que hace que el bucle realice las iteraciones de las sentencias mientras sea verdadera. Finalmente, en incremento se modifica la variable de control.

El bucle while

Repite la ejecución de una sentencia o de un grupo de sentencias mientras la condición que se evalúa (al principio del bucle) sea verdadera.

El bucle do while

Repite la ejecución de una sentencia o de un grupo de sentencias mientras se cumpla la condición especificada (al menos una vez). La condición del do while evalúa al final del bucle.

Contadores: un contador es una variable casi siempre de tipo entero cuyo valor se incrementa o decrementa en cada repetición del bucle. Es habitual llamar a esta variable "cont" (contador) o "i" (índice); El contador suele utilizarse de este modo:

1. Se inicializa antes de que comience el bucle. Es decir, se le da un valor inicial- Por ejemplo: **cont = 1**

- Se modifica dentro del cuerpo del bucle. Lo más habitual es que se incremente su valor en una unidad:
Por ejemplo: `cont = cont + 1;`

Esto quiere decir que el valor de la variable **cont** se incrementa en una unidad y es asignado de nuevo al contador. Es decir, si **cont** valía 1 antes de esta instrucción, **cont** valdrá 2 después. Otra forma típica del contador es: `cont = cont - 1.`

Formas de incrementar los contadores:

Primera Forma	Segunda Forma	Tercera Forma
<code>cont = cont + 1</code>	<code>cont++</code>	<code>cont += 1</code>

Acumuladores: Las variables acumuladores tienen la misión de almacenar los resultados sucesivos, es decir, de acumular resultados. Las variables acumuladores también deben ser inicializadas. Si llamamos "acum" a un acumulador, escribiremos antes de iniciar el bucle algo como esto: `acum = 0.`

Por supuesto, el valor inicial puede cambiar, dependiendo de la naturaleza del problema. Más tarde, en el cuerpo del bucle, la forma en la que solemos encontrarla es:

`acum = acum + N;` otra manera de escribir esta sentencia sería: `acum += N.`

Desarrollo

Ejemplo 1

Vamos a realizar un programa que sea capaz de leer 5 números y mostrar el cubo de cada uno de ellos. Hacer uso de una estructura for.

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace For
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             Double cubo;
14             int i;
15             Console.Title = "UTILIZANDO EL FOR";
16             Console.WriteLine("Cálculo de Cubos");
17             for (i=0; i < 5; i++)
18             {
19                 Console.WriteLine("Escriba una cantidad: ");
20                 Double Cantidad = Double.Parse(Console.ReadLine());
21                 cubo = Math.Pow(Cantidad,3);
22                 Console.WriteLine("El cubo de la cantidad es: "+cubo);
23             }
24             Console.WriteLine("Saliendo del programa...");
25             Console.ReadKey();
26         }
27     }
28 }

```

Ejemplo 2

Se pide desarrollar un programa en C# que que lea 10 números desde teclado y presente la sumatoria de los mismos. Hacer uso de la estructura repetitiva **while**.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Guía_5_Ejemplo_2
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             int contador = 1, suma=0;
14             int N;
15             Console.Title = "UTILIZANDO EL BUCLE WHILE";
16             while (contador <= 10)
17             {
18                 Console.WriteLine("\tIngrese un número  "+contador);
19                 N = int.Parse(Console.ReadLine());
20                 suma = suma + N;
21                 contador++;
22             }
23             Console.WriteLine("\tLa suma de los 10 números ingresados es:"+suma);
24             Double Promedio = suma / 10;
25             Console.WriteLine("\tEl promedio de los números ingresados es: "+Promedio);
26             Console.ReadKey();
27         }
28     }
29 }
```

Ejemplo 3

Desarrollar un Programa que combine el uso de for y do while. Este programa capturara desde teclado los datos de uno o varios empleados y calculará el total invertido en sueldos.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Guía_5_Ejemplo_3
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             int op;
14             string Nombre, Ocupacion;
15             Double Sueldo, Cantidad, Total=0;
16             Console.Title = "UTILIZANDO EL DO WHILE";
```

```

17         do {
18             Console.Clear();
19             Console.WriteLine("\t¿Cuántos empleados va a registrar?");
20             Cantidad = int.Parse(Console.ReadLine());
21             for (int i = 1; i <= Cantidad; i++)
22             {
23                 Console.WriteLine("\tIngrese el nombre del empleado(a): "+i);
24                 Nombre = Console.ReadLine();
25                 Console.WriteLine("\tIngrese la ocupación del empleado(a): ");
26                 Ocupacion = Console.ReadLine();
27                 Console.WriteLine("\tIngrese el sueldo: ");
28                 Sueldo = Double.Parse(Console.ReadLine());
29                 Total = Total + Sueldo;
30             }
31             Console.WriteLine("\tEl total de dinero invertido en sueldos es: "+Total);
32             Console.WriteLine("\tSi desea continuar, presione 1, sino presione 0");
33             op = int.Parse(Console.ReadLine());
34         } while (op == 1);
35         Console.ReadKey();
36     }
37 }
38 }

```

Responda

Una vez que has hecho estos ejemplos, deberás de contestar los siguientes ejercicios

1. Desarrolle un programa que presente el siguiente menú:
 - a) Elevar potencia.
 - b) Extraer Raíz cuadrada.
 - c) Verificar si un número es par o impar.
 - d) Salir del programa.

Consideraciones:

- ✓ El menú tiene que estar activo hasta que el usuario decida salir del programa.
 - ✓ Cada vez que se seleccione una opción deberá realizar una limpieza de pantalla para más orden en la aplicación.
 - ✓ Se recomienda la utilización de la estructura switch case y la estructura repetitiva do while.
2. Desarrollar un programa que capture un número desde teclado e imprima su tabla de multiplicar.
 3. Desarrollar un programa que permita ingresar n números enteros y luego nos imprima cuántos valores fueron pares y cuántos impares.

4. Desarrollar un programa que, dados como datos 25 números enteros, obtenga la suma de los números impares y el promedio de los números pares.
5. En un banco se procesan datos de las cuentas corrientes de sus clientes. De cada cuenta corriente se conoce: número de cuenta y saldo actual. El ingreso de datos debe finalizar al ingresar un valor negativo en el número de cuenta.

Se pide realizar un programa que lea los datos de las cuentas corrientes e informe:

- a) De cada cuenta: número de cuenta y estado de la cuenta según su saldo, sabiendo que:
Del Acreedor si el saldo es >0 . Del Deudor' si el saldo es <0 . Nulo' si el saldo es $=0$.
- b) La suma total de los saldos acreedores.

Esta Práctica se deberá de entregar a más tardar el día jueves 23/08/2018 antes de las 10:00 PM, toda práctica que no sea enviada dentro de este horario será evaluada con cero.

Bibliografía

- Deitel, Harvey M. y Paul J. Deitel, Cómo Programar en C#, Segunda Edición, México, 2007.