

PROGRAMACIÓN ORIENTADA A OBJETOS 2o. PARCIAL

Practica #4 Estructuras en C#

Daniel Vázquez de la Rosa
d_vazquez@outlook.com

Instituto Politécnico Nacional



Escuela Superior de Cómputo

Septiembre 2018

Datos Personales

Crear un Documento en Word que incluya una caratula que contenga tus datos. Dicho documento será guardado como: Grupo_ApellidoPaterno_NombreAlumno_Practica3_POO y deberá contener toda la información de la practica creada.

Objetivos de la Práctica

- Conocer el concepto de estructura y cómo se utilizan en C#.
- Conocer las operaciones con estructuras y sus aplicaciones.
- Desarrollar programas utilizando estructuras.

Introducción

En el desarrollo de aplicaciones se necesita trabajar con diferentes tipos de datos asignados dependiendo el tipo de información con la que se trabaja.

En clases anteriores, se ha trabajado con la primera estructura de datos: los arreglos. Dicha estructura permite trabajar con una colección de elementos del mismo tipo; sin embargo, hay problemas que requiere la utilización de varios tipos de datos que describen una misma entidad.

¿Qué es una entidad?

Una entidad es todo aquello que se puede describir. La descripción puede hacerse en términos de atributos y métodos.

Definición

Una estructura es un tipo de valor que se suele utilizar para encapsular pequeños grupos de variables relacionadas. Las variables que se agrupan no necesariamente deben ser del mismo.

Un registro o estructura se define con la palabra clave **struct**.

¿Qué pueden contener las Estructuras?

- Constructores.
- Constantes.
- Campos.
- Métodos.
- Propiedades.
- Indizadores.
- Operadores.
- Eventos.

- Tipos anidados.

Para declarar una estructura, se utiliza la siguiente sintaxis:

```
Modificador_Acceso struct Nombre_Estructura
{
    Miebros; // Variables, Funciones y/o Procedimientos.
}
```

En C#, primero deberemos declarar cual va a ser la estructura de nuestro registro, lo que no se puede hacer dentro de "Main". Más adelante, ya dentro de "Main", podremos declarar variables de ese nuevo tipo.

Los datos que forman un "**struct**" pueden ser públicos o privados. Por ahora, a nosotros nos interesará que sean accesibles desde el resto de nuestro programa, por lo que les añadiremos delante la palabra "**public**" para indicar que queremos que sean públicos.

Ya desde el cuerpo del programa, para acceder a cada uno de los datos que forman el registro, tanto si queremos leer su valor como si queremos cambiarlo, se debe indicar el nombre de la variable y el del dato (o campo) separados por un punto:

En este contexto, al operador punto lo llamaremos **operador de acceso**, ya que permitirá acceder a cada uno de los miembros que forman parte de la estructura, es decir, podremos acceder a las variables.

¿Cómo se realiza el acceso a los miembros de una estructura?

Variable_tipo_estructura.nombre_del_miembro

Desarrollo

Ejemplo 1

Forma de capturar y mostrar datos utilizando estructuras.

```
1  using System;
2
3  namespace Ejemplo 1
4  {
5      class Program
6      {
7          struct Estudiante
8          {
9              public String Nombre;
10             public int Edad;
11             public String Asignatura;
12             public int Tel;
13         }
14
15         public static void Main(string[] args)
16         {
17             String Opcion;
18             //Instanciamos la estructura
19             Estudiante Objeto = new Estudiante();
20             int Op;
21             do {
22                 Console.WriteLine("\t1.Registrar");
23                 Console.WriteLine("\t2.Mostrar");
24                 Console.WriteLine("\t3.salir");
25                 Console.WriteLine("\tOpción?");
26                 Op = int.Parse(Console.ReadLine());
27                 switch (Op)
28                 {
```

```

29         case 1:
30             Console.Clear();
31             Console.WriteLine("\tINGRESE LA INFORMACIÓN DEL ESTUDIANTE");
32             Console.WriteLine("\nNombre: ");
33             Objeto.Nombre = Console.ReadLine();
34             Console.WriteLine("\nEdad: ");
35             Objeto.Edad = int.Parse(Console.ReadLine());
36             Console.WriteLine("\nAsignatura: ");
37             Objeto.Asignatura = Console.ReadLine();
38             Console.WriteLine("\nTeléfono: ");
39             Objeto.Tel = int.Parse(Console.ReadLine());
40             break;
41
42         case 2:
43             Console.Clear();
44             Console.WriteLine("\tNombre:" + Objeto.Nombre);
45             Console.WriteLine("\tEdad:" + Objeto.Edad);
46             Console.WriteLine("\tAsignatura:" + Objeto.Asignatura);
47             Console.WriteLine("\tTeléfono:" + Objeto.Tel);
48             break;
49     }
50     Console.WriteLine("\tDesea continuar? (S/N)");
51     Opcion = Console.ReadLine();
52 } while (Opcion == "s");
53 Console.ReadKey();
54 }
55 }
56 }

```

Ejemplo 2

Arreglos de Estructuras: En ocasiones se trabajará con datos o campos que provienen de varias estructuras, entonces lo que se hace es introducir una estructura dentro de otra. Veamos el siguiente ejemplo:

```

1  using System;
2
3
4  namespace Ejemplo 2
5  {
6      class Program
7      {
8          struct Persona
9          {
10             public String Nombre;
11             public int Edad;
12             public String Direccion;
13         }
14
15         public static void Main(string[] args)
16         {
17             int max;
18             Console.WriteLine("\t¿Cuántos registros?");
19             max = int.Parse(Console.ReadLine());
20             Persona[] Acceso = new Persona[max];
21             for(int i=0; i<max; i++)
22             {
23                 Console.WriteLine("\tPersona con ID: ["+(i+1)+"]");
24                 Console.WriteLine("\tNombre: ");
25                 Acceso[i].Nombre = Console.ReadLine();
26                 Console.WriteLine("\tEdad: ");
27                 Acceso[i].Edad = int.Parse(Console.ReadLine());
28                 Console.WriteLine("\tDirección: ");
29                 Acceso[i].Direccion = Console.ReadLine();
30             }

```

```

31
32         for (int i = 0; i < max; i++)
33         {
34             Console.WriteLine("\tPersona con ID: ["+(i+1)+"]");
35             Console.WriteLine("\tNombre: " + Acceso[i].Nombre);
36             Console.WriteLine("\tEdad: " + Acceso[i].Edad);
37             Console.WriteLine("\tDirección: " + Acceso[i].Direccion);
38             Console.WriteLine("\t-----");
39         }
40         Console.ReadKey();
41     }
42 }
43

```

Ejemplo 3: Estructuras Anidadas

```

1     using System;
2
3     namespace Ejemplo 3
4     {
5         class Program
6         {
7             struct Programa
8             {
9                 public string Universidad;
10                public string Carrera;
11                public string Asignatura;
12            }
13
14            struct Estudiante
15            {
16                public string Nombre;
17                public int Edad;
18                public Programa Estudios;
19            }
20
21            public static void Main(string[] args)
22            {
23                Estudiante Student = new Estudiante();
24
25                Console.WriteLine("\tIngrese Nombre: ");
26                Student.Nombre = Console.ReadLine();
27                Console.WriteLine("\tEdad: ");
28                Student.Edad = int.Parse(Console.ReadLine());
29                Console.WriteLine("\tCarrera: ");
30                //Accedemos a miembros de otra estructura:
31                Student.Estudios.Carrera = Console.ReadLine();
32                Console.WriteLine("\tUniversidad: ");
33                Student.Estudios.Universidad = Console.ReadLine();
34
35                Console.WriteLine("\tMostrando los datos: ");
36                Console.WriteLine("\tEstudiante: " + Student.Nombre);
37                Console.WriteLine("\tEdad: " + Student.Edad);
38                Console.WriteLine("\tCarrera: " + Student.Estudios.Carrera);
39                Console.WriteLine("\tUniversidad: " + Student.Estudios.Universidad);
40                Console.Read();
41            }
42        }
43    }

```

Ejemplo 4: Estructuras y Procedimientos

```
1  using System;
2
3  class Datos
4  {
5
6      struct Alumno
7      {
8          public String carnet;
9          public String Nombre;
10         public void Mostrar(Alumno X)
11         {
12             Console.WriteLine("\tNombre: " + X.Nombre);
13             Console.WriteLine("\tCarnet: " + X.carnet);
14         }
15     }
16
17     public static void Main()
18     {
19         Alumno Var = new Alumno();
20         int op,0;
21         Console.Title = "UTILIZANDO ESTRUCTURAS Y PROCEDIMIENTOS";
22         do {
23             Console.Clear();
24             Console.WriteLine("\tMENU");
25             Console.WriteLine("\t1.Ingresar.");
26             Console.WriteLine("\t2.Mostrar.");
27             Console.WriteLine("\t3.Ingresar opción.");
28             op = int.Parse(Console.ReadLine());
29             switch (op)
30             {
31                 case 1:
32                     Console.Clear();
33                     Console.WriteLine("\tNombre: ");
34                     Var.Nombre = Console.ReadLine();
35                     Console.WriteLine("\tCarnet: ");
36                     Var.carnet = Console.ReadLine();
37                     break;
38
39                 case 2:
40                     Console.Clear();
41                     Var.Mostrar(Var);
42                     break;
43             }
44             Console.WriteLine("\tSi desea continuar presione 1, sino presione 2");
45             0 = int.Parse(Console.ReadLine());
46         } while (0 == 1);
47         Console.ReadKey();
48     }
49 }
```

Ejemplo_5: Paso de parámetros a método miembro de estructura:

```
1  using System;
2
3  namespace Ejemplo 5
4  {
5      class Program
6      {
7          struct Ejemplo
8          {
9              public string Name;
10             public string Genero;
11             public int Edad;
12             public void Registrar(string n, string g, int e)
13             {
14                 Name = n;
15                 Genero = g;
16                 Edad = e;
17             }
18         }
19
20         static void Main(string[] args)
21         {
22             int Opcion, Op;
23             int tam;
24             Console.Title = "Estructuras con paso de parámetros.";
25             Console.WriteLine("\t\tIngrese el número de estudiantes a registrar: ");
26             tam = int.Parse(Console.ReadLine());
27             Ejemplo[] E = new Ejemplo[tam];
28             do {
29                 Console.Clear();
30                 Console.WriteLine("\nMenú:");
31                 Console.WriteLine("\t\t1.Registrar.");
32                 Console.WriteLine("\t\t2.Mostrar.");
33                 Console.WriteLine("\t\t3.Salir.");
34                 Console.WriteLine("\t\tIngrese su opción: ");
35                 Opcion = int.Parse(Console.ReadLine());
36                 switch (Opcion)
37                 {
38                     case 1:
39                         for (int i = 0; i < tam; i++)
40                         {
41                             Console.Clear();
42                             Console.WriteLine("\tNombre: ");
43                             string n = Console.ReadLine();
44                             Console.WriteLine("\tGénero:");
45                             string g = Console.ReadLine();
46                             Console.WriteLine("\tEdad: ");
47                             int e = int.Parse(Console.ReadLine());
48                             E[i].Registrar(n, g, e);
49                         }
50                     break;
```

```

51
52         case 2:
53             Console.WriteLine("\tInformación ingresada...");
54             Console.WriteLine("\t-----");
55             for (int i = 0; i < tam; i++)
56             {
57                 Console.WriteLine("\tNombre: " + E[i].Name);
58                 Console.WriteLine("\tGénero: " + E[i].Genero);
59                 Console.WriteLine("\tEdad: " + E[i].Edad);
60                 Console.WriteLine("\n\t-----");
61             }
62             break;
63         }
64         Console.WriteLine("\tPara continuar presione 1...");
65         Op = int.Parse(Console.ReadLine());
66     } while (Op == 1);
67
68     Console.ReadKey();
69 }
70
71

```

Responda

Desarrollar las siguientes aplicaciones en C#.

1. Modifique el ejemplo 4 para que acepte más de un registro.
2. Desarrolle un programa haciendo uso de estructuras que simule una agenda. La aplicación debe solicitar los datos y mantenerlos en tiempo de ejecución (el programa debe estar activo hasta que se escoja la opción salir). Los datos que se deben solicitar son los siguientes: Nombre, DUI, Dirección, Teléfono, Email, Profesión u Oficio. El menú debe tener las opciones: 1. Ingresar Datos. 2. Mostrar Datos. 3. Buscar Persona. 4. Salir. En la opción 3, debe buscar por DUI, ya que es un dato que es único para cada usuario, al encontrar dicha persona el programa debe mostrar los datos encontrados.

Esta Práctica se deberá de entregar a más tardar el día lunes 01/10/2018 antes de las 10:00 PM, toda práctica que no sea enviada dentro de este horario será evaluada con cero.

Al Enviar la Practica, deberás poner en el asunto del correo que envíes la siguiente información: Practica3_Grupo_ApellidoPaterno_ApellidoMaterno_Nombre

Bibliografía

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