Facultad: Ingeniería
Escuela: Computación
Asignatura: Programación II

Tema: Arreglos de objetos en C#.

Objetivos

- Describir la implementación de arreglos de Objetos
- Implementar programas en C# que incluyan el uso de arreglos de objetos.

Materiales y Equipo

- Computadora con el software Visual Studio 2013.
- Guía Número 6.

Introducción Teórica

Arreglo de objetos.
La función básica de un arreglo es almacenar en una variable más de un valor de un mismo tipo de dato, por ejemplo la siguiente declaración int[ ] numero= new int [5]; permite almacenar en la variable numero, 5 valores enteros.

En las clases el concepto de arreglos es el mismo, con la diferencia que ahora se almacenarán objetos de una clase o de diferentes clases.

Los objetos se pueden estructurar como un array. Los objetos son variables y tienen las mismas capacidades y atributos que cualquier tipo de variables, por tanto es posible disponer objetos en un array.

La sintaxis es exactamente igual a la utilizada para declarar y acceder al array. También disponemos de arrays bidimensionales.

Cuando se crea un array de objetos éstos se inicializan llamando al constructor sin

argumentos. Por consiguiente, siempre que se prevea organizar los objetos en un array, la clase debe tener un constructor que pueda llamarse sin parámetros.

Sintaxis para la definición del arreglo:

```
nombre_clase [ ] nombrevector = new nombre_clase[tamaño]; /*creación del espacio de memoria para el vector*/
nombrevector[x]= new clase( ); /*creación de las clases*/
```

Cuando necesitamos invocar algún elemento (propiedad o método) de la clase desde el Programa principal lo hacemos así:

```
nombrevector[x].elementoinvocado;  //si es una propiedad
nombrevector[x].elementoinvocado( ); //si es un método (si tiene parámetros no olvidarlos)
```

Recordemos que cada variación de x representa un nuevo objeto dentro del arreglo, con todos los atributos y métodos que implique.

Procedimiento

Ejemplo No. 1:

En el ejercicio de ejemplo se plantea un registro de información básica de clientes, en donde se almacenarán el código, nombre, apellido y NIT. La diferencia fundamental de este ejercicio es que tenemos muchos clientes que debemos registrar, por lo cual se hará por medio de un arreglo de objetos. Después debemos mostrar esos clientes registrados.

a. Para ello primero creamos la clase cliente de la siguiente forma:

b. Con ello estamos declarando únicamente sus atributos y las respectivas propiedades para acceder a ellos.

c. Posteriormente en la clase principal (Program) vamos a crear el arreglo de objetos

```
class cliente
{
    string codigo;
    string nombre;
    string apellido;
    string NIT;

    public string Codigo
    {
        get { return codigo; }
        set
        { codigo = value; }
     }

    public string Nombre
    {
        get { return nombre; }
        set
        { nombre = value; }
    }

    public string Apellido
    {
        get { return apellido; }
        set
        { apellido = value; }
    }

    public string Nit
    {
        get { return NIT; }
        set
        { NIT = value; }
    }
```

```
static void Main(string[] args)
{
    int tamvec;

    Console.WriteLine("Ingrese número de clientes que registrará");
    tamvec = int.Parse(Console.ReadLine());
    cliente[] vectorCliente = new cliente[tamvec];

  for (int i = 0; i < vectorCliente.Length; i++)
    {
        vectorCliente[i] = new cliente();

        Console.WriteLine("Codigo: ");
        vectorCliente[i].Codigo = Console.ReadLine();
        Console.WriteLine("Nombre: ");
        vectorCliente[i].Nombre = Console.ReadLine();
        Console.WriteLine("Apellido: ");
        vectorCliente[i].Apellido = Console.ReadLine();
        Console.WriteLine("NIT: ");
        vectorCliente[i].Nit = Console.ReadLine();
    }
    Console.WriteLine("\n \n LISTADO DE CLIENTES EN ARREGLO");
    for (int j = 0; j < vectorCliente.Length; j++)
    {
        Console.WriteLine(vectorCliente[j].Codigo + "\t" + vectorCliente[j].Nombre + "\t" + vectorCliente[j].Apellido + "\t"
         + vectorCliente[j].Nit);
    }
    Console.ReadKey();
}
```

Asigna espacio de memoria para arreglo completo

Crea los objetos del arreglo

Ejemplo No. 2:

El programa crea una clase denominada alumno, la cual contiene arreglo para sus notas. Se piden los datos básicos del alumno y en el menú de opciones se ingresan a los estudiantes, se consultan todos los estudiantes inscritos y finalmente se puede ver el registro de todos.

a. Para ello creamos la clase alumno

```
class alumno
{
    string carnet;

public string Carnet...
    string nombre;

public string Nombre...
    string apellido;

public string Apellido...
    string materia;

public string Materia...
    float [] calificaciones = new float[3];

public float[] Calificaciones...

    public void ingresardatos()
    {
        Console.WriteLine("\nIngrese el carnet del estudiante");
        carnet = Console.ReadLine();

        Console.WriteLine("\nIngrese el nombre del estudiante");
        nombre=Console.ReadLine();

        Console.WriteLine("\nIngrese el apellido del estudiante");
        apellido=Console.ReadLine();

        Console.WriteLine("\nIngrese la materia del estudiante");
        materia=Console.ReadLine();

        int i;
        for (i = 0; i < 3; i++)
        {
            Console.WriteLine("\n Ingrese la nota {0} de la materia {1} del estudiante {2}: ", i+1, materia,nombre);
            calificaciones[i] = float.Parse(Console.ReadLine());
        }
    }


    public void mostrar()
    {
        float acumula =0;
        Console.WriteLine("\nEl alumno: " + Nombre +" "+ Apellido + " con carnet "+ Carnet);
        Console.WriteLine("\nEstá cursando la materia de " + materia);
        Console.WriteLine("\nSus notas en esa asignatura son: ");
        for (int i = 0; i < 3; i++)
        {
           Console.Write(calificaciones[i] +"    " );
            acumula= calificaciones[i] + acumula;

        }
        float promedio = acumula / 3;
        Console.WriteLine("\n\nY su promedio es: " + promedio);
        Console.WriteLine("\n\n---------------------------- ");

        }
    }

}
```

b. Hacemos un menú en la clase Program para crear el arreglo, los objetos e invocar métodos

```
static void Main(string[] args)
{
    int tamvec;
    int op;

    Console.WriteLine("Ingrese número de estudiantes en su grupo");
    tamvec = int.Parse(Console.ReadLine());
    Console.Clear();
    alumno[] Estudiante = new alumno[tamvec];

    do{
        Console.WriteLine("\t ****** MENU *******");
        Console.WriteLine("1.Ingresar datos del nuevo estudiante");
        Console.WriteLine("2.Ver lista de estudiantes inscritos");
        Console.WriteLine("3. Reporte de estudiantes");
        Console.WriteLine("4. Salir");


        op=int.Parse(Console.ReadLine());
        Console.Clear();

        switch(op)
        {
            case 1:
                Console.WriteLine("SECCIÓN DE INGRESO");
                for (int i = 0; i < Estudiante.Length; i++)
                {
                 Estudiante[i] = new alumno();
                 Estudiante[i].ingresardatos();
                 Console.Clear();
                }
                 break;

  case 2:
      Console.WriteLine("\n---------------------- ");
      Console.WriteLine("\nLISTADO ALUMNOS");
      Console.WriteLine("\n---------------------- ");
      for (int i = 0; i < Estudiante.Length; i++)
      {
          Console.WriteLine("Estudiante número "+ (i+1) +": ");
          Console.WriteLine(Estudiante[i].Nombre + " " +Estudiante[i].Apellido);
          Console.WriteLine("\n");
      }
      Console.WriteLine("\n");
      Console.ReadKey();
      Console.Clear();

       break;

  case 3:
      Console.WriteLine("\nREPORTE DE ESTUDIANTES");
      for(int i=0;i<Estudiante.Length;i++)
      {
      Estudiante[i].mostrar();
      }
      Console.ReadKey();
      Console.Clear();
      break;

  case 4:
      break;

        default:
            Console.WriteLine("\n Escriba opción válida");
            Console.ReadKey();
            break;
    }
    }
   while(op!=4);
Console.ReadKey();
}
```

Análisis de Resultados

Ejercicio No.1:

Basados en los ejercicios de ejemplo modifíquelos de tal forma que:

a) El ejemplo 1 pueda ingresar los datos y también mostrarlos valiéndose de métodos que estén en la clase. (debe crear los métodos correspondientes)

b) El ejemplo 2 permita ingresar varias materias para un alumno p.ej. [matemática, química, física….] y cada materia tenga disponibles tres notas para promediar

c) Siempre tomando de base el ejemplo 2; agregue una opción similar a la opción 3, con la condicionante que solo me muestre un alumno en particular (ya sea por su correlativo, carnet u otro y muestre todos los datos de él únicamente).

Investigación Complementaria

Tarea No.1:

Investigar sobre ArrayList de C# ¿Es igual que tener un arreglo de objetos? Explique y haga un ejemplo de ello utilizando clases.

Tarea No.2:

Modifique el ejemplo 2 de forma que las materias y toda la información relacionadas a ella (profesor asignado, nombre, UV, notas y promedio) estén en otra clase que se relacione con la clase alumno. Adicional a ello genere usted el carnet de los estudiantes usando el mismo formato de la UDB primeras letras de apellido y nombre y un correlativo. El formato debe tener dos letras y seis números (XX-000000).