



Presentación

Nombres: Cristopher Manuel

Apellidos: Guerrero Tapia

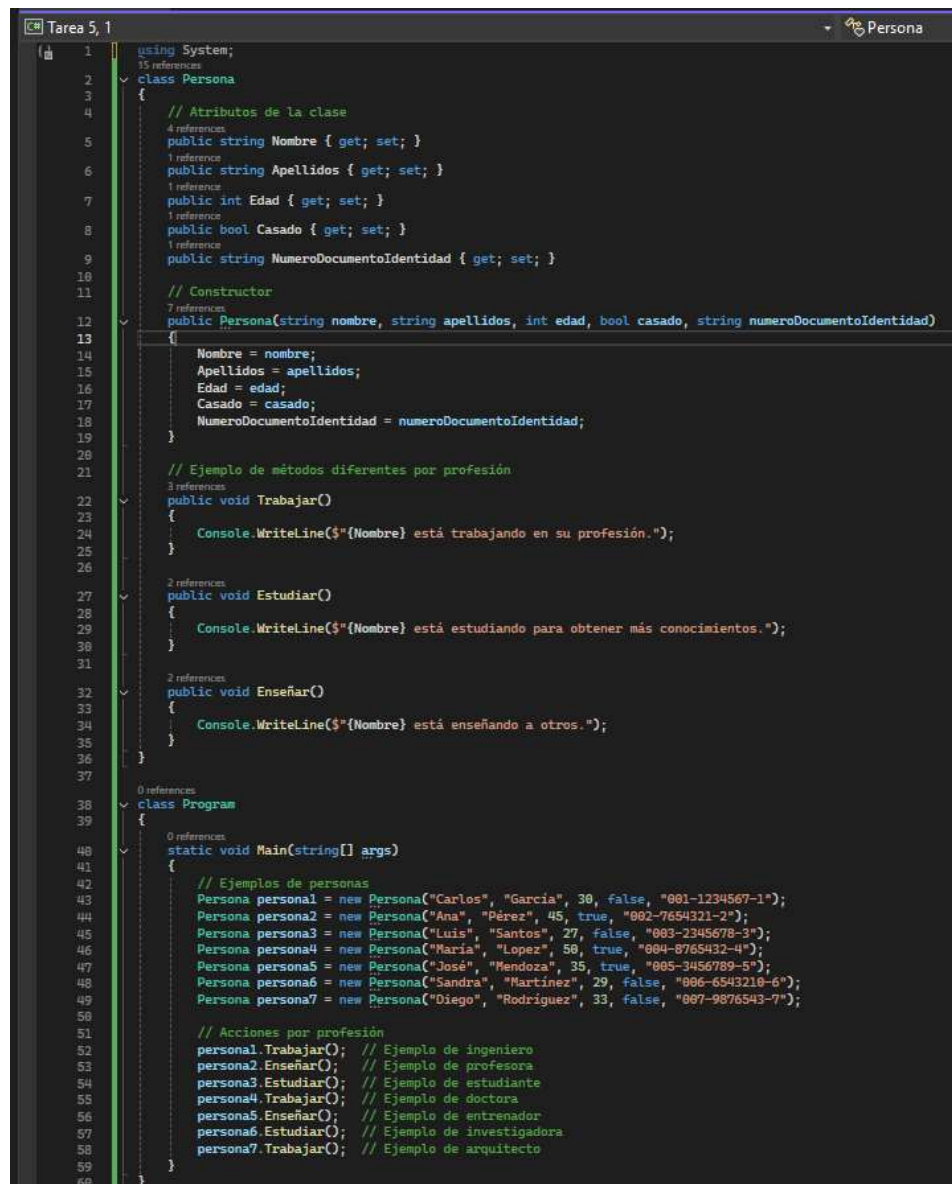
Matricula: 2024-2052

Materia: Programación I

Docente: Starling Alfredo Germosén Reynoso

Tarea 5

Considera estás desarrollando un programa donde necesitas trabajar con objetos de tipo Persona. Define una clase Persona, pero en este caso considerando los siguientes atributos de clase: nombre (String), apellidos (String), edad (int), casado (boolean), numeroDocumentoIdentidad(String) y 3 metodos como acciones diferentes por persona de acuerdo a una profesión. Define un constructor y los métodos para poder establecer y obtener los valores de los atributos. Mínimo 7 personas diferentes con acciones diferentes.



```
1 using System;
2 class Persona
3 {
4     // Atributos de la clase
5     public string Nombre { get; set; }
6     public string Apellidos { get; set; }
7     public int Edad { get; set; }
8     public bool Casado { get; set; }
9     public string NumeroDocumentoIdentidad { get; set; }
10
11     // Constructor
12     public Persona(string nombre, string apellidos, int edad, bool casado, string numeroDocumentoIdentidad)
13     {
14         Nombre = nombre;
15         Apellidos = apellidos;
16         Edad = edad;
17         Casado = casado;
18         NumeroDocumentoIdentidad = numeroDocumentoIdentidad;
19     }
20
21     // Ejemplo de métodos diferentes por profesión
22     public void Trabajar()
23     {
24         Console.WriteLine($"{Nombre} está trabajando en su profesión.");
25     }
26
27     public void Estudiar()
28     {
29         Console.WriteLine($"{Nombre} está estudiando para obtener más conocimientos.");
30     }
31
32     public void Enseñar()
33     {
34         Console.WriteLine($"{Nombre} está enseñando a otros.");
35     }
36 }
37
38 class Program
39 {
40     static void Main(string[] args)
41     {
42         // Ejemplos de personas
43         Persona persona1 = new Persona("Carlos", "Garcia", 30, false, "001-1234567-1");
44         Persona persona2 = new Persona("Ana", "Pérez", 45, true, "002-7654321-2");
45         Persona persona3 = new Persona("Luis", "Santos", 27, false, "003-2345678-3");
46         Persona persona4 = new Persona("Maria", "Lopez", 50, true, "004-8765432-4");
47         Persona persona5 = new Persona("José", "Mendoza", 35, true, "005-3456789-5");
48         Persona persona6 = new Persona("Sandra", "Martínez", 29, false, "006-6543210-6");
49         Persona persona7 = new Persona("Diego", "Rodríguez", 33, false, "007-9876543-7");
50
51         // Acciones por profesión
52         persona1.Trabajar(); // Ejemplo de ingeniero
53         persona2.Enseñar(); // Ejemplo de profesora
54         persona3.Estudiar(); // Ejemplo de estudiante
55         persona4.Trabajar(); // Ejemplo de doctora
56         persona5.Enseñar(); // Ejemplo de entrenador
57         persona6.Estudiar(); // Ejemplo de investigadora
58         persona7.Trabajar(); // Ejemplo de arquitecto
59     }
60 }
```

Github: <https://github.com/CristopherGuerrero/Tareas-ITLA/blob/main/Tarea%205%2C%201/Tarea%205%2C%201/Program.cs>

Considera estás desarrollando un programa donde necesitas trabajar con objetos de tipo Persona. Define una clase Persona, pero en este caso considerando los siguientes atributos de clase: nombre (String), apellidos (String), edad (int), casado (boolean), numeroDocumentoIdentidad (String) y 3 metodos como acciones diferentes por persona de acuerdo a una profesión. Define un constructor y los métodos para poder establecer y obtener los valores de los atributos. Mínimo 7 personas diferentes con acciones diferentes.

```
Tarea 5, 2
1  using System;
2  15 references
3  class Persona
4  {
5      // Atributos de la clase
6      4 references
7      public string Nombre { get; set; }
8      1 reference
9      public string Apellidos { get; set; }
10     1 reference
11     public int Edad { get; set; }
12     1 reference
13     public bool Casado { get; set; }
14     1 reference
15     public string NumeroDocumentoIdentidad { get; set; }
16
17     // Constructor
18     7 references
19     public Persona(string nombre, string apellidos, int edad, bool casado, string numeroDocumentoIdentidad)
20     {
21         Nombre = nombre;
22         Apellidos = apellidos;
23         Edad = edad;
24         Casado = casado;
25         NumeroDocumentoIdentidad = numeroDocumentoIdentidad;
26
27     }
28
29     // Métodos que representan acciones según la profesión
30     3 references
31     public void Cocinar()
32     {
33         Console.WriteLine($"{Nombre} está cocinando deliciosos platillos.");
34     }
35
36     2 references
37     public void Construir()
38     {
39         Console.WriteLine($"{Nombre} está trabajando en un proyecto de construcción.");
40     }
41
42     2 references
43     public void AtenderPacientes()
44     {
45         Console.WriteLine($"{Nombre} está atendiendo a sus pacientes.");
46     }
47 }
48
49 0 references
50 class Program
51 {
52     0 references
53     static void Main(string[] args)
54     {
55         // Crear objetos de tipo Persona
56         Persona persona1 = new Persona("Juan", "López", 30, false, "123-4567890-1");
57         Persona persona2 = new Persona("María", "Gómez", 45, true, "987-6543210-2");
58         Persona persona3 = new Persona("Carlos", "Rodríguez", 35, true, "456-7890123-3");
59         Persona persona4 = new Persona("Ana", "Martínez", 28, false, "789-1234567-4");
60         Persona persona5 = new Persona("Pedro", "Santos", 40, true, "321-6549870-5");
61         Persona persona6 = new Persona("Sofía", "Castro", 32, false, "654-9873210-6");
62         Persona persona7 = new Persona("Luis", "Fernández", 50, true, "159-7538642-7");
63
64         // Acciones específicas de cada persona
65         persona1.Cocinar(); // Ejemplo de chef
66         persona2.AtenderPacientes(); // Ejemplo de doctora
67         persona3.Construir(); // Ejemplo de ingeniero
68         persona4.Cocinar(); // Otra chef
69         persona5.Construir(); // Ejemplo de albañil
70         persona6.AtenderPacientes(); // Ejemplo de enfermera
71         persona7.Cocinar(); // Ejemplo de panadero
72     }
73 }
```

Github: <https://github.com/CristopherGuerrero/Tareas-ITLA/blob/main/Tarea%205%2C%202/Tarea%205%2C%202/Program.cs>

Crea una clase Contador con los métodos para incrementar y decrementar el contador. La clase contendrá un constructor por defecto, un constructor con parámetros, y los métodos getters y setters.

```
C# Tarea 5, 3
1  using System;
2  6 references
3  class Contador
4  {
5      // Atributo de la clase
6      private int valor;
7
8      // Constructor por defecto
9      1 reference
10     public Contador()
11     {
12         valor = 0; // Inicializamos el contador en 0
13     }
14
15     // Constructor con parámetros
16     1 reference
17     public Contador(int valorInicial)
18     {
19         valor = valorInicial;
20     }
21
22     // Método para incrementar el contador
23     1 reference
24     public void Incrementar()
25     {
26         valor++;
27     }
28
29     // Método para decrementar el contador
30     1 reference
31     public void Decrementar()
32     {
33         if (valor > 0) // Evitar valores negativos
34         {
35             valor--;
36         }
37         else
38         {
39             Console.WriteLine("El valor no puede ser menor que cero.");
40         }
41     }
42
43     // Getter para obtener el valor del contador
44     3 references
45     public int ObtenerValor()
46     {
47         return valor;
48     }
49
50     // Setter para establecer un valor en el contador
51     1 reference
52     public void EstablecerValor(int nuevoValor)
53     {
54         if (nuevoValor >= 0) // Validación básica
55         {
56             valor = nuevoValor;
57         }
58         else
59         {
60             Console.WriteLine("El valor debe ser mayor o igual a cero.");
61         }
62     }
63 }
```

```
57 0 references
58 class Program
59 {
60     0 references
61     static void Main(string[] args)
62     {
63         // Uso del constructor por defecto
64         Contador contador1 = new Contador();
65         contador1.Incrementar();
66         Console.WriteLine("Valor del contador1: " + contador1.ObtenerValor());
67
68         // Uso del constructor con parámetros
69         Contador contador2 = new Contador(10);
70         contador2.Decrementar();
71         Console.WriteLine("Valor del contador2: " + contador2.ObtenerValor());
72
73         // Usando el setter para establecer un valor inicial
74         contador1.EstablecerValor(5);
75         Console.WriteLine("Nuevo valor del contador1: " + contador1.ObtenerValor());
76     }
77 }
```

Github: <https://github.com/CristopherGuerrero/Tareas-ITLA/blob/main/Tarea%20%2C%203/Tarea%20%2C%203/Program.cs>

Crea una clase Libro con los métodos préstamo, devolución y ToString. La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos getters y setters.

```
C# Tarea 5, 4
1  using System;
2  6 references
3  class Libro
4  {
5      // Atributos de la clase
6      7 references
7      public string Titulo { get; set; }
8      3 references
9      public string Autor { get; set; }
10     7 references
11     public bool Prestado { get; private set; } // Indica si el libro está prestado
12
13     // Constructor por defecto
14     1 reference
15     public Libro()
16     {
17         Titulo = "Sin título";
18         Autor = "Desconocido";
19         Prestado = false;
20     }
21     // Constructor con parámetros
22     1 reference
23     public Libro(string titulo, string autor)
24     {
25         Titulo = titulo;
26         Autor = autor;
27         Prestado = false; // Por defecto, el libro no está prestado
28     }
29     // Método para realizar el préstamo
30     1 reference
31     public void Prestar()
32     {
33         if (!Prestado)
34         {
35             Prestado = true;
36             Console.WriteLine($"El libro '{Titulo}' ha sido prestado.");
37         }
38         else
39         {
40             Console.WriteLine($"El libro '{Titulo}' ya está prestado.");
41         }
42     }
43     // Método para realizar la devolución
44     1 reference
45     public void Devolver()
46     {
47         if (Prestado)
48         {
49             Prestado = false;
50             Console.WriteLine($"El libro '{Titulo}' ha sido devuelto.");
51         }
52         else
53         {
54             Console.WriteLine($"El libro '{Titulo}' no estaba prestado.");
55         }
56     }
57     // Método ToString para mostrar la información del libro
58     4 references
59     public override string ToString()
60     {
61         string estado = Prestado ? "Prestado" : "Disponible";
62         return $"Titulo: {Titulo}, Autor: {Autor}, Estado: {estado}";
63     }
64 }
```



```
56 0 references
57 class Program
58 {
59     0 references
60     static void Main(string[] args)
61     {
62         // Uso del constructor por defecto
63         Libro libro1 = new Libro();
64         Console.WriteLine(libro1.ToString());
65
66         // Uso del constructor con parámetros
67         Libro libro2 = new Libro("Cien años de soledad", "Gabriel Garcia Márquez");
68         Console.WriteLine(libro2.ToString());
69
70         // Préstamo y devolución
71         libro2.Prestar();
72         Console.WriteLine(libro2.ToString());
73         libro2.Devolver();
74         Console.WriteLine(libro2.ToString());
75     }
76 }
```

Github: <https://github.com/CristopherGuerrero/Tareas-ITLA/blob/main/Tarea%20%2C%204/Tarea%20%2C%204/Program.cs>

Crea una clase Fracción con métodos para sumar, restar, multiplicar y dividir fracciones.

```
Tarea 5, 5 Program
1 using System;
2 class Fraccion
3 {
4     // Atributos de la clase
5     public int Numerador { get; set; }
6     public int Denominador { get; set; }
7
8     // Constructor por defecto
9     public Fraccion()
10    {
11        Numerador = 0;
12        Denominador = 1; // Evitamos divisiones entre 0
13    }
14
15    // Constructor con parámetros
16    public Fraccion(int numerador, int denominador)
17    {
18        if (denominador == 0)
19        {
20            throw new ArgumentException("El denominador no puede ser cero.");
21        }
22
23        Numerador = numerador;
24        Denominador = denominador;
25    }
26
27    // Método para sumar dos fracciones
28    public Fraccion Sumar(Fraccion otra)
29    {
30        int nuevoNumerador = (Numerador * otra.Denominador) + (otra.Numerador * Denominador);
31        int nuevoDenominador = Denominador * otra.Denominador;
32        return new Fraccion(nuevoNumerador, nuevoDenominador);
33    }
34
35    // Método para restar dos fracciones
36    public Fraccion Restar(Fraccion otra)
37    {
38        int nuevoNumerador = (Numerador * otra.Denominador) - (otra.Numerador * Denominador);
39        int nuevoDenominador = Denominador * otra.Denominador;
40        return new Fraccion(nuevoNumerador, nuevoDenominador);
41    }
42
43    // Método para multiplicar dos fracciones
44    public Fraccion Multiplicar(Fraccion otra)
45    {
46        int nuevoNumerador = Numerador * otra.Numerador;
47        int nuevoDenominador = Denominador * otra.Denominador;
48        return new Fraccion(nuevoNumerador, nuevoDenominador);
49    }
50
51    // Método para dividir dos fracciones
52    public Fraccion Dividir(Fraccion otra)
53    {
54        if (otra.Numerador == 0)
55        {
56            throw new ArgumentException("No se puede dividir entre una fracción con numerador 0.");
57        }
58
59        int nuevoNumerador = Numerador * otra.Denominador;
60        int nuevoDenominador = Denominador * otra.Numerador;
61        return new Fraccion(nuevoNumerador, nuevoDenominador);
62    }
63 }
```



```

63
64 // Método ToString para representar la fracción como texto
65 public override string ToString()
66 {
67     return $"{Numerador}/{Denominador}";
68 }
69 }
70
71 0 references
72 class Program
73 {
74     0 references
75     static void Main(string[] args)
76     {
77         // Crear dos fracciones
78         Fraccion fraccion1 = new Fraccion(1, 2); // 1/2
79         Fraccion fraccion2 = new Fraccion(3, 4); // 3/4
80
81         // Operaciones entre fracciones
82         Fraccion suma = fraccion1.Sumar(fraccion2);
83         Console.WriteLine($"Suma: {suma}");
84
85         Fraccion resta = fraccion1.Restar(fraccion2);
86         Console.WriteLine($"Resta: {resta}");
87
88         Fraccion multiplicacion = fraccion1.Multiplicar(fraccion2);
89         Console.WriteLine($"Multiplicación: {multiplicacion}");
90
91         Fraccion division = fraccion1.Dividir(fraccion2);
92         Console.WriteLine($"División: {division}");
93     }
94 }

```

Github: <https://github.com/CristopherGuerrero/Tareas-ITLA/blob/main/Tarea%20%2C%20Tarea%20%2C%20Program.cs>