

1. 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.

```
using System;

15 referencias
class Persona
{
    public string Nombre;
    public string Apellidos;
    public int Edad;
    public bool Casado;
    public string NumeroDocumentoIdentidad;

    7 referencias
    public Persona(string nombre, string apellidos, int edad, bool casado, string numeroDocumentoIdentidad)
    {
        this.Nombre = nombre;
        this.Apellidos = apellidos;
        this.Edad = edad;
        this.Casado = casado;
        this.NumeroDocumentoIdentidad = numeroDocumentoIdentidad;
    }

    3 referencias
    public void Enseñar()
    {
        Console.WriteLine(Nombre + " " + Apellidos + " está enseñando.");
    }

    2 referencias
    public void OperarPaciente()
    {
        Console.WriteLine(Nombre + " " + Apellidos + " está operando un paciente.");
    }

    2 referencias
    public void ProgramarSoftware()
    {
        Console.WriteLine(Nombre + " " + Apellidos + " está programando.");
    }
}

0 referencias
class Program
{
    0 referencias
    static void Main()
    {
        Persona p1 = new Persona("Juan", "Pérez", 45, true, "12345678A");
        Persona p2 = new Persona("Ana", "Gómez", 38, false, "87654321B");
        Persona p3 = new Persona("Carlos", "López", 30, true, "56781234C");
        Persona p4 = new Persona("Pedro", "Martínez", 40, false, "11223344D");
        Persona p5 = new Persona("María", "Rodríguez", 35, true, "55667788E");
        Persona p6 = new Persona("Luis", "Fernández", 33, false, "99887766F");
        Persona p7 = new Persona("Sofía", "García", 28, true, "33445566G");

        p1.Enseñar();
        p2.OperarPaciente();
        p3.ProgramarSoftware();
        p4.Enseñar();
        p5.OperarPaciente();
        p6.ProgramarSoftware();
        p7.Enseñar();
    }
}
```

```
Juan Pérez está enseñando.
Ana Gómez está operando un paciente.
Carlos López está programando.
Pedro Martínez está enseñando.
María Rodríguez está operando un paciente.
Luis Fernández está programando.
Sofía García está enseñando.
```

```
C:\PROGRAMACION\Codigo de programacion\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase\bin\Debug\net8.0\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase.exe (proceso 24084) se cerró con el código 0 (0x0).
```

```
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración -> Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```

2. Crea una clase Cuenta con los métodos ingreso, reintegro y transferencia. La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos getters y setters para mostrar e ingresar.

```
using System;

7 referencias
class Cuenta
{
    public string Titular;
    public double Saldo;

    0 referencias
    public Cuenta()
    {
        Titular = "Desconocido";
        Saldo = 0.0;
    }

    2 referencias
    public Cuenta(string titular, double saldo)
    {
        this.Titular = titular;
        this.Saldo = saldo;
    }

    1 referencia
    public void Ingreso(double cantidad)
    {
        if (cantidad > 0)
        {
            Saldo += cantidad;
            Console.WriteLine("Ingreso realizado. Nuevo saldo: " + Saldo);
        }
        else
        {
            Console.WriteLine("La cantidad debe ser positiva.");
        }
    }

    1 referencia
    public void Reintegro(double cantidad)
    {
        if (cantidad > 0 && cantidad <= Saldo)
        {
            Saldo -= cantidad;
            Console.WriteLine("Reintegro realizado. Nuevo saldo: " + Saldo);
        }
        else
        {
            Console.WriteLine("Saldo insuficiente o cantidad inválida.");
        }
    }

    1 referencia
    public void Transferencia(Cuenta destino, double cantidad)
```

```
    {
        if (cantidad > 0 && cantidad <= Saldo)
        {
            Saldo -= cantidad;
            destino.Saldo += cantidad;
            Console.WriteLine("Transferencia realizada. Nuevo saldo: " + Saldo);
        }
        else
        {
            Console.WriteLine("No se pudo realizar la transferencia.");
        }
    }

    4 referencias
    public void Mostrar()
    {
        Console.WriteLine("Titular: " + Titular + ", Saldo: " + Saldo);
    }
}

0 referencias
class Program
{
    static void Main()
    {
        Cuenta cuenta1 = new Cuenta("Juan Pérez", 1000);
        Cuenta cuenta2 = new Cuenta("Ana Gómez", 500);

        cuenta1.Mostrar();
        cuenta2.Mostrar();

        cuenta1.Ingreso(200);
        cuenta1.Reintegro(100);
        cuenta1.Transferencia(cuenta2, 300);

        cuenta1.Mostrar();
        cuenta2.Mostrar();
    }
}
```

```
Titular: Juan Pérez, Saldo: 1000
Titular: Ana Gómez, Saldo: 500
Ingreso realizado. Nuevo saldo: 1200
Reintegro realizado. Nuevo saldo: 1100
Transferencia realizada. Nuevo saldo: 800
Titular: Juan Pérez, Saldo: 800
Titular: Ana Gómez, Saldo: 800
```

C:\PROGRAMACION\Codigo de programacion\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase\bin\Debug\net8.0\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase.exe (proceso 4544) se cerró con el código 0 (0x0).

Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración -> Cerrar la consola automáticamente al detenerse la depuración.

3. 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.

```
using System;

6 referencias
class Contador
{
    public int Valor;

    1 referencia
    public Contador()
    {
        Valor = 0;
    }

    1 referencia
    public Contador(int valorInicial)
    {
        Valor = valorInicial;
    }

    2 referencias
    public void Incrementar()
    {
        Valor++;
        Console.WriteLine("Valor incrementado: " + Valor);
    }

    2 referencias
    public void Decrementar()
    {
        Valor--;
        Console.WriteLine("Valor decrementado: " + Valor);
    }

    0 referencias
    public int GetValor()
    {
        return Valor;
    }

    0 referencias
    public void SetValor(int nuevoValor)
    {
        Valor = nuevoValor;
    }
}

0 referencias
class Program
{
    0 referencias
    static void Main()
    {
        Contador contador1 = new Contador();
        Contador contador2 = new Contador(10);

        contador1.Incrementar();
        contador1.Decrementar();

        contador2.Incrementar();
        contador2.Decrementar();
    }
}
```

```
Valor incrementado: 1
Valor decrementado: 0
Valor incrementado: 11
Valor decrementado: 10
```

4. 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.

```
using System;

class Libro
{
    public string Titulo;
    public string Autor;
    public int Ejemplares;
    public int Prestados;

    // constructor por defecto
    public Libro()
    {
        Titulo = "Desconocido";
        Autor = "Desconocido";
        Ejemplares = 0;
        Prestados = 0;
    }

    // constructor con parámetros
    public Libro(string titulo, string autor, int ejemplares)
    {
        Titulo = titulo;
        Autor = autor;
        Ejemplares = ejemplares;
        Prestados = 0;
    }

    // método préstamo
    public void Prestamo()
    {
        if (Prestados < Ejemplares)
        {
            Prestados++;
            Console.WriteLine("Libro prestado: " + Titulo);
        }
        else
        {
            Console.WriteLine("No hay ejemplares disponibles.");
        }
    }

    // método devolución
    public void Devolucion()
    {
        if (Prestados > 0)
        {
            Prestados--;
            Console.WriteLine("Libro devuelto: " + Titulo);
        }
        else
        {
            Console.WriteLine("No hay ejemplares prestados.");
        }
    }

    // método ToString
    public override string ToString()
    {
        return "Título: " + Titulo + ", Autor: " + Autor + ", Ejemplares: " + Ejemplares + ", Prestados: " + Prestados;
    }
}

class Program
{
    static void Main()
    {
        Libro libro1 = new Libro("1984", "George Orwell", 3);
        Libro libro2 = new Libro("Cien años de soledad", "Gabriel García Márquez", 2);

        Console.WriteLine(libro1.ToString());
        Console.WriteLine(libro2.ToString());

        libro1.Prestamo();
        libro1.Prestamo();
        libro1.Devolucion();
        libro1.Prestamo();

        Console.WriteLine(libro1.ToString());
    }
}
```

```
Título: 1984, Autor: George Orwell, Ejemplares: 3, Prestados: 0
Título: Cien años de soledad, Autor: Gabriel García Márquez, Ejemplares: 2, Prestados: 0
Libro prestado: 1984
Libro prestado: 1984
Libro devuelto: 1984
Libro prestado: 1984
Título: 1984, Autor: George Orwell, Ejemplares: 3, Prestados: 2
```

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

```
class Fraccion
{
    public int Numerador;
    public int Denominador;

    6 referencias
    public Fraccion(int numerador, int denominador)
    {
        if (denominador == 0)
        {
            throw new ArgumentException("El denominador no puede ser cero.");
        }
        Numerador = numerador;
        Denominador = denominador;
    }

    1 referencia
    public Fraccion Sumar(Fraccion otra)
    {
        int nuevoNumerador = (Numerador * otra.Denominador) + (otra.Numerador * Denominador);
        int nuevoDenominador = Denominador * otra.Denominador;
        return new Fraccion(nuevoNumerador, nuevoDenominador);
    }

    1 referencia
    public Fraccion Restar(Fraccion otra)
    {
        int nuevoNumerador = (Numerador * otra.Denominador) - (otra.Numerador * Denominador);
        int nuevoDenominador = Denominador * otra.Denominador;
        return new Fraccion(nuevoNumerador, nuevoDenominador);
    }

    1 referencia
    public Fraccion Multiplicar(Fraccion otra)
    {
        int nuevoNumerador = Numerador * otra.Numerador;
        int nuevoDenominador = Denominador * otra.Denominador;
        return new Fraccion(nuevoNumerador, nuevoDenominador);
    }

    1 referencia
    public Fraccion Dividir(Fraccion otra)
    {
        if (otra.Numerador == 0)
        {
            throw new ArgumentException("No se puede dividir por una fracción con numerador 0.");
        }
        int nuevoNumerador = Numerador * otra.Denominador;
        int nuevoDenominador = Denominador * otra.Numerador;
        return new Fraccion(nuevoNumerador, nuevoDenominador);
    }

    0 referencias
    public override string ToString()
    {
        return Numerador + "/" + Denominador;
    }
}
```

Consola de depuración de Mi

Suma: 10/8
Resta: -2/8
Multiplicación: 3/8
División: 4/6

C:\PROGRAMACION\Codigo de programacion\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase\bin\Debug\net8.0\Prog1_CSharp_Tarea5_Nombre_Matricula_DiaDeClase.exe (proceso 7284) se cerró con el código 0 (0x0).

Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración -> Cerrar la consola automáticamente al detenerse la depuración.

Presione cualquier tecla para cerrar esta ventana. . .|