

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.

```
namespace Person
```

```
{
```

```
    using System;
```

```
    using System.Timers;
```

```
    public class Persona
```

```
    {
```

```
        public Persona() { }
```

```
        public Persona(string nombre, string apellido, int edad, bool casado,  
string docID)
```

```
        {
```

```
            Nombre = nombre;
```

```
            Apellido = apellido;
```

```
            Edad = edad;
```

```
            Casado = casado;
```

```
            DocID = docID;
```

```
        }
```

```
public string Nombre { get; set; }
```

```
public string Apellido { get; set; }
```

```
public int Edad { get; set; }
```

```
public bool Casado { get; set; }
```

```
public string DocID { get; set; }
```

```
public void Accion()
```

```
{
```

```
    Console.WriteLine($"{Nombre} realiza una accion.");
```

```
}
```

```
public string Hablar(string discurso)
```

```
{
```

```
    return discurso;
```

```
}
```

```
public string Comer(string comida)
```

```
{
```

```
    return $"La persona come {comida}";
```

```
}
```

```
}
```

```
public class Medico : Persona
```

```
{
```

```
    public Medico()
```

```

    {
    }

    public Medico(string nombre, string apellido, int edad, bool casado,
string docID) : base(nombre, apellido, edad, casado, docID) { }

    public void Pastilla()
    {
        Console.WriteLine("El doctor receta un medicamento.");
    }

    public void Consulta()
    {
        Console.WriteLine("El doctor realiza una consulta medica.");
    }

    public void Cirugia()
    {
        Console.WriteLine("El doctor hace una cirugia.");
    }
}

public class Ingeniero : Persona
{
    public Ingeniero()
    {

```

```

    }

    public Ingeniero(string nombre, string apellido, int edad, bool casado,
string docID) : base(nombre, apellido, edad, casado, docID) { }

    public void DisenarObra()
    {
        Console.WriteLine("El ingeniero disena la obra.");
    }

    public void SupervisarObra()
    {
        Console.WriteLine("El ingeniero observa la obra.");
    }

    public void Problema()
    {
        Console.WriteLine("El ingeniero esta resolviendo un problema.");
    }
}

public class Profesor : Persona
{
    public Profesor()
    {
    }
}

```

```
    public Profesor(string nombre, string apellido, int edad, bool casado,
string docID) : base(nombre, apellido, edad, casado, docID)
```

```
    { }
```

```
    public void Enseñar()
```

```
    {
```

```
        Console.WriteLine("El maestro esta enseñando una clase.");
```

```
    }
```

```
    public void HacerExamen()
```

```
    {
```

```
        Console.WriteLine("El maestro esta escribiendo un examen.");
```

```
    }
```

```
    public void EvaluarEstudiantes()
```

```
    {
```

```
        Console.WriteLine("El profesor califica los estudiantes.");
```

```
    }
```

```
}
```

```
public class Abogado : Persona
```

```
{
```

```
    public Abogado()
```

```
    {
```

```
    }
```

```
public Abogado(string nombre, string apellido, int edad, bool casado,
string docID) : base(nombre, apellido, edad, casado, docID) { }
```

```
public void Cliente()
{
    Console.WriteLine("El abogado defiende a sus cliente.");
}
```

```
public void Robar()
{
    Console.WriteLine("El abogado roba a su cliente.");
}
```

```
public voidCodigoProceduralPenal()
{
    Console.WriteLine("El abogado esta defendiendo a un posible
criminal, en base a lo estipulado dentro del Codigo Procedural Penal de la
Republica dominicana, en donde se garantiza la defensa legal completa e
integral para cualquier ciudadano dominicano.");
}
}
```

```
public class Chef : Persona
{
    public Chef()
    {
```

```

    }

    public Chef(string nombre, string apellido, int edad, bool casado,
string docID) : base(nombre, apellido, edad, casado, docID) { }

    public void Cocinar()
    {
        Console.WriteLine("El chef esta cocinando.");
    }

    public void SupervisarCocina()
    {
        Console.WriteLine("El chef esta supervisando la cocina.");
    }

    public void CrearReceta()
    {
        Console.WriteLine("El chef esta creando una nueva receta para
cocinar fritos con salami.");
    }
}

public class Policia : Persona
{
    public Policia()
    {

```

```

    }

    public Policia(string nombre, string apellido, int edad, bool casado,
string docID) : base(nombre, apellido, edad, casado, docID) { }

    public void Patrullaje()
    {
        Console.WriteLine("El policía esta patrullando la 42.");
    }

    public void ProcesoPenal()
    {
        Console.WriteLine("El policía se lleva un kitipo.");
    }

    public void DetenerSospechoso()
    {
        Console.WriteLine("El policía para al motorista.");
    }
}
}

```

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.

namespace cuenta


```

{
    public class Cuenta
    {
        public Cuenta()
        {
        }

        public Cuenta(int Num, string Titular, decimal Value,
List<string> Trans, List<string> Benef)
        {
            int _num = Num;
            string _titular = Titular;
            decimal _value = Value;
            List<string> _Trans = Trans;
            List<string> _benef = Benef;
        }
        public int Num { get; set; }
        public string Titular { get; set; }
        public decimal Value { get; set; }
        public List<string> Trans { get; set; }
        public List<string> Benef { get; set; }

        public void Ingreso( decimal deposito )
        {
            this.Value += deposito;
        }

        public void Reintegro(decimal retiro ) //Retiro??
        {
            this.Value -= retiro;
        }

        public void Transferencia(Cuenta Receptor, decimal
valorDepositado )
        {

```

```

        try
        {
            decimal impuesto = Convert.ToDecimal(valorDepositado >=
50.0m ? (valorDepositado * 0.05m) : 0m); //No solicitado jaja

            Value -= (valorDepositado + impuesto);
            Receptor.Value += valorDepositado;
        }
        catch (NullReferenceException)
        {
            Console.WriteLine("Algo salio mal al
encontrar el receptor. Intente nuevamente.");
        }
    }
}

```

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.

```

namespace counter
{
    public class Contador
    {

        public Contador()
        {
        }
        public Contador(int start = 0)
        {
        }
    }
}

```

```

        int _start = start;
    }

    public int Start { get; set; }

    public int Incremento(int leap = 1)
    {
        Console.WriteLine(Start);
        return this.Start += leap;
    }

    public int Decremento(int leap = 1)
    {
        Console.WriteLine(Start);
        return this.Start -= leap;
    }
}
}

```

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.

```

namespace Libreria
{
    public class Libro
    {

        public Libro()
        {

```

```
}
```

```
public Libro(int isbn, string nombre, string autor, bool disponibilidad =  
true)
```

```
{
```

```
    ISBN = isbn;
```

```
    Nombre = nombre;
```

```
    Autor = autor;
```

```
    Disponibilidad = disponibilidad;
```

```
}
```

```
public int ISBN { get; set; }
```

```
public string Nombre { get; set; }
```

```
public string Autor { get; set; }
```

```
public bool Disponibilidad { get; set; }
```

```
public void Prestamo(string solicitador)
```

```
{
```

```
    Console.WriteLine($"El libro {this.ISBN} ({this.Nombre}) de {this.Autor}  
ha sido prestado a {solicitador}.");
```

```
    this.Disponibilidad = false;
```

```
    //Juro que el codigo no fue hecho por LLM, solo es asi de malo.
```

```
}
```

```

    public void Devolucion(string solicitador)
    {
        Console.WriteLine($"El libro {this.ISBN} ({this.Nombre}) de {this.Autor}
ha sido devuelto por {solicitador}.");
        this.Disponibilidad = true;
    }

    public override string ToString()
    {
        string isbn = this.ISBN.ToString();
        string disponibilidad = this.Disponibilidad.ToString();

        return $"{isbn}, {Nombre}, {Autor}, {disponibilidad}";
    }
}

```

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

```

namespace Fraccionario
{
    public class Fraccion
    {
        public Fraccion(double num = 0, double den = 0)
        {

```

```

        Numerador = num;
        Denominador = den;
    }

    public double Numerador { get; set; }
    public double Denominador { get; set; }

    public Fraccion Sumar(Fraccion fraccion)
    {
        if (this.Denominador == fraccion.Denominador)
        {
            return new Fraccion((this.Numerador + fraccion.Numerador),
this.Denominador);

        }
        else
        {
            return new Fraccion(((this.Numerador *
fraccion.Denominador) + (this.Denominador * fraccion.Numerador)),
(this.Denominador * fraccion.Denominador));
        }
    }
    public Fraccion Restar(Fraccion fraccion)
    {
        if (this.Denominador == fraccion.Denominador)
        {
            return new Fraccion((this.Numerador - fraccion.Numerador),
this.Denominador);

        }
        else
        {

```

```

        return new Fraccion(((this.Numerador *
fraccion.Denominador) - (this.Denominador * fraccion.Numerador)),
(this.Denominador * fraccion.Denominador));
    }
}

    public Fraccion Multiplicar(Fraccion fraccion)
    {
        return new Fraccion((this.Numerador * fraccion.Numerador),
(this.Denominador * fraccion.Denominador));
    }

    public Fraccion Dividir(Fraccion fraccion)
    {
        return new Fraccion((this.Numerador * fraccion.Denominador),
(this.Denominador * fraccion.Numerador));
    }
}

```