

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.

```
10 referencias
public class Persona
{
    private string nombre;
    private string apellidos;
    private int edad;
    private bool casado;
    private 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;
    }

    0 referencias
    public string Nombre { get => nombre; set => nombre = value; }
    0 referencias
    public string Apellidos { get => apellidos; set => apellidos = value; }
    0 referencias
    public int Edad { get => edad; set => edad = value; }
    0 referencias
    public bool Casado { get => casado; set => casado = value; }
    0 referencias
    public string NumeroDocumentoIdentidad { get => numeroDocumentoIdentidad; set => numeroDocumentoIdentidad = value; }

    3 referencias
    public void Trabajar() => Console.WriteLine($"{nombre} está trabajando.");
    2 referencias
    public void Estudiar() => Console.WriteLine($"{nombre} está estudiando.");
    2 referencias
    public void Conducir() => Console.WriteLine($"{nombre} está conduciendo.");
}
```

```

0 referencias
class Program
{
    0 referencias
    static void Main(string[] args)
    {
        // Personas con acciones
        List<Persona> personas = new List<Persona>
        {
            new Persona("Ana", "Pérez", 28, false, "ID001"),
            new Persona("Luis", "Ramírez", 34, true, "ID002"),
            new Persona("Carlos", "Vega", 22, false, "ID003"),
            new Persona("Laura", "Martínez", 40, true, "ID004"),
            new Persona("Diego", "López", 29, false, "ID005"),
            new Persona("Sofía", "Gómez", 31, true, "ID006"),
            new Persona("Miguel", "Torres", 36, true, "ID007"),
        };

        personas[0].Estudiar();
        personas[1].Trabajar();
        personas[2].Conducir();
        personas[3].Trabajar();
        personas[4].Estudiar();
        personas[5].Conducir();
        personas[6].Trabajar();
    }
}

```

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.

```

public class Cuenta
{
    private double saldo;

    1 referencia
    public Cuenta() { saldo = 0; }
    1 referencia
    public Cuenta(double saldoInicial) { saldo = saldoInicial; }

    2 referencias
    public double Saldo { get => saldo; set => saldo = value; }

    1 referencia
    public void Ingreso(double cantidad)
    {
        if (cantidad > 0) saldo += cantidad;
    }

    0 referencias
    public void Reintegro(double cantidad)
    {
        if (cantidad > 0 && cantidad <= saldo) saldo -= cantidad;
    }

    1 referencia
    public void Transferencia(Cuenta destino, double cantidad)
    {
        if (cantidad > 0 && cantidad <= saldo)
        {
            saldo -= cantidad;
            destino.Ingreso(cantidad);
        }
    }
}

```

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.

```

public class Contador
{
    private int valor;

    1 referencia
    public Contador() { valor = 0; }
    0 referencias
    public Contador(int valorInicial) { valor = valorInicial; }

    1 referencia
    public int Valor { get => valor; set => valor = value; }

    2 referencias
    public void Incrementar() => valor++;
    1 referencia
    public void Decrementar() => valor--;
}

```

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.

```
public class Libro
{
    private string titulo;
    private string autor;
    private bool prestado;

    0 referencias
    public Libro() { titulo = ""; autor = ""; prestado = false; }
    1 referencia
    public Libro(string titulo, string autor)
    {
        this.titulo = titulo;
        this.autor = autor;
        this.prestado = false;
    }

    0 referencias
    public string Titulo { get => titulo; set => titulo = value; }
    0 referencias
    public string Autor { get => autor; set => autor = value; }
    0 referencias
    public bool Prestado { get => prestado; set => prestado = value; }

    1 referencia
    public void Prestamo() => prestado = true;
    1 referencia
    public void Devolucion() => prestado = false;

    0 referencias
    public override string ToString() => $"Titulo: {titulo}, Autor: {autor}, Prestado: {prestado}";
}
```

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

```

public class Fraccion
{
    private int numerador;
    private int denominador;

    6 referencias
    public Fraccion(int numerador, int denominador)
    {
        this.numerador = numerador;
        this.denominador = denominador != 0 ? denominador : 1;
    }

    0 referencias
    public int Numerador { get => numerador; set => numerador = value; }
    0 referencias
    public int Denominador { get => denominador; set => denominador = value != 0 ? value : 1; }

    1 referencia
    public Fraccion Sumar(Fraccion f)
    {
        int num = numerador * f.denominador + f.numerador * denominador;
        int den = denominador * f.denominador;
        return new Fraccion(num, den);
    }

    1 referencia
    public Fraccion Restar(Fraccion f)
    {
        int num = numerador * f.denominador - f.numerador * denominador;
        int den = denominador * f.denominador;
        return new Fraccion(num, den);
    }

    1 referencia
    public Fraccion Multiplicar(Fraccion f) => new Fraccion(numerador * f.numerador, denominador * f.denominador);
    1 referencia
    public Fraccion Dividir(Fraccion f) => new Fraccion(numerador * f.denominador, denominador * f.numerador);
    0 referencias
    public override string ToString() => $"{numerador}/{denominador}";
}

```