

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.

```
2.
3. public class Persona
4. {
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.
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.     public void AccionDoctor()
22.     {
23.         Console.WriteLine($"{Nombre} {Apellidos} está tratando a un
paciente.");
24.     }
25.
26.     public void AccionIngeniero()
27.     {
28.         Console.WriteLine($"{Nombre} {Apellidos} está diseñando un
puente.");
29.     }
30.
31.     public void AccionProfesor()
32.     {
33.         Console.WriteLine($"{Nombre} {Apellidos} está dando una
clase.");
34.     }
35.
36.     public void AccionAbogado()
37.     {
38.         Console.WriteLine($"{Nombre} {Apellidos} está defendiendo a un
cliente en el tribunal.");
39.     }
```

```

40.
41.     public void AccionChef()
42.     {
43.         Console.WriteLine($"{Nombre} {Apellidos} está preparando una
comida gourmet.");
44.     }
45.
46.     public void AccionArtista()
47.     {
48.         Console.WriteLine($"{Nombre} {Apellidos} está pintando una obra
maestra.");
49.     }
50.
51.     public void AccionBombero()
52.     {
53.         Console.WriteLine($"{Nombre} {Apellidos} está apagando un
incendio.");
54.     }
55.
56.     public static void Main(string[] args)
57.     {
58.         Persona doctor = new Persona("Ana", "García", 45, true,
"12345678A");
59.         Persona ingeniero = new Persona("Juan", "Pérez", 30, false,
"87654321B");
60.         Persona profesor = new Persona("María", "López", 55, true,
"11223344C");
61.         Persona abogado = new Persona("Carlos", "Martínez", 40, false,
"44332211D");
62.         Persona chef = new Persona("Sofía", "Hernández", 28, false,
"55667788E");
63.         Persona artista = new Persona("Luis", "Fernández", 35, true,
"66778899F");
64.         Persona bombero = new Persona("David", "Rodríguez", 32, true,
"77889900G");
65.
66.         doctor.AccionDoctor();
67.         ingeniero.AccionIngeniero();
68.         profesor.AccionProfesor();
69.         abogado.AccionAbogado();
70.         chef.AccionChef();
71.         artista.AccionArtista();
72.         bombero.AccionBombero();
73.     }
74.}

```

75. 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;

```
public class Cuenta
{
    private double saldo;
    private string titular;

    public Cuenta()
    {
        saldo = 0.0;
        titular = "Desconocido";
    }

    public Cuenta(double saldoInicial, string titular)
    {
        this.saldo = saldoInicial;
        this.titular = titular;
    }

    public double Saldo
    {
        get { return saldo; }
        set { saldo = value; }
    }

    public string Titular
    {
        get { return titular; }
        set { titular = value; }
    }

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

```
        Console.WriteLine($"Ingreso de {cantidad} realizado. Saldo  
actual: {saldo}");  
    }  
    else  
    {  
        Console.WriteLine("Cantidad de ingreso no válida.");  
    }  
}
```

```
public void Reintegro(double cantidad)  
{  
    if (cantidad > 0 && cantidad <= saldo)  
    {  
        saldo -= cantidad;  
        Console.WriteLine($"Reintegro de {cantidad} realizado. Saldo  
actual: {saldo}");  
    }  
    else  
    {  
        Console.WriteLine("Cantidad de reintegro no válida o saldo  
insuficiente.");  
    }  
}
```

```
public void Transferencia(Cuenta otraCuenta, double cantidad)  
{  
    if (cantidad > 0 && cantidad <= saldo)  
    {  
        saldo -= cantidad;  
        otraCuenta.Ingreso(cantidad);  
        Console.WriteLine($"Transferencia de {cantidad} a  
{otraCuenta.Titular} realizada. Saldo actual: {saldo}");  
    }  
}
```

```

        else
        {
            Console.WriteLine("Cantidad de transferencia no válida o saldo
insuficiente.");
        }
    }

    public static void Main(string[] args)
    {
        Cuenta cuenta1 = new Cuenta(500.0, "Alice");
        Cuenta cuenta2 = new Cuenta(300.0, "Bob");

        cuenta1.Ingreso(200.0);
        cuenta1.Reintegro(100.0);
        cuenta1.Transferencia(cuenta2, 150.0);

        Console.WriteLine($"Saldo de {cuenta1.Titular}: {cuenta1.Saldo}");
        Console.WriteLine($"Saldo de {cuenta2.Titular}: {cuenta2.Saldo}");
    }
}

```

76. 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.

```

77. using System;
78.
79. public class Contador
80. {
81.     private int valor;
82.
83.     public Contador()
84.     {
85.         valor = 0;

```

```

86.     }
87.
88.     public Contador(int valorInicial)
89.     {
90.         valor = valorInicial;
91.     }
92.
93.     public int Valor
94.     {
95.         get { return valor; }
96.         set { valor = value; }
97.     }
98.
99.     public void Incrementar()
100.    {
101.        valor++;
102.        Console.WriteLine($"El contador ha sido incrementado.
Valor actual: {valor}");
103.    }
104.
105.     public void Decrementar()
106.     {
107.         if (valor > 0)
108.         {
109.             valor--;
110.             Console.WriteLine($"El contador ha sido decrementado.
Valor actual: {valor}");
111.         }
112.         else
113.         {
114.             Console.WriteLine("El valor del contador no puede ser
negativo.");
115.         }
116.     }
117.
118.     public static void Main(string[] args)
119.     {
120.         Contador contador1 = new Contador();
121.         contador1.Incrementar();
122.         contador1.Decrementar();
123.
124.         Contador contador2 = new Contador(10);
125.         contador2.Incrementar();
126.         contador2.Decrementar();
127.     }
128. }

```

129. 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.

```

130.     using System;
131.
132.     public class Libro

```

```

133.     {
134.
135.         private string titulo;
136.         private string autor;
137.         private int anioPublicacion;
138.         private bool prestado;
139.
140.
141.         public Libro()
142.         {
143.             titulo = "Desconocido";
144.             autor = "Desconocido";
145.             anioPublicacion = 0;
146.             prestado = false;
147.         }
148.
149.
150.         public Libro(string titulo, string autor, int anioPublicacion)
151.         {
152.             this.titulo = titulo;
153.             this.autor = autor;
154.             this.anioPublicacion = anioPublicacion;
155.             this.prestado = false;
156.         }
157.         public string Titulo
158.         {
159.             get { return titulo; }
160.             set { titulo = value; }
161.         }
162.
163.         public string Autor
164.         {
165.             get { return autor; }
166.             set { autor = value; }
167.         }
168.
169.         public int AnioPublicacion
170.         {
171.             get { return anioPublicacion; }
172.             set { anioPublicacion = value; }
173.         }
174.
175.         public bool Prestado
176.         {
177.             get { return prestado; }
178.             set { prestado = value; }
179.         }
180.
181.         public void Prestamo()
182.         {
183.             if (!prestado)
184.             {
185.                 prestado = true;
186.                 Console.WriteLine($"El libro '{titulo}' ha sido
prestado.");
187.             }
188.             else
189.             {

```

```

190.             Console.WriteLine($"El libro '{titulo}' ya está
prestado.");
191.         }
192.     }
193.
194.     public void Devolucion()
195.     {
196.         if (prestado)
197.         {
198.             prestado = false;
199.             Console.WriteLine($"El libro '{titulo}' ha sido
devuelto.");
200.         }
201.         else
202.         {
203.             Console.WriteLine($"El libro '{titulo}' no estaba
prestado.");
204.         }
205.     }
206.
207.
208.     public override string ToString()
209.     {
210.         return $"Título: {titulo}, Autor: {autor}, Año de
Publicación: {anioPublicacion}, Prestado: {prestado}";
211.     }
212.
213.     public static void Main(string[] args)
214.     {
215.
216.         Libro libro1 = new Libro();
217.         Console.WriteLine(libro1.ToString());
218.
219.
220.         Libro libro2 = new Libro("Cien Años de Soledad", "Gabriel
García Márquez", 1967);
221.         Console.WriteLine(libro2.ToString());
222.
223.
224.         libro2.Prestamo();
225.         Console.WriteLine(libro2.ToString());
226.         libro2.Devolucion();
227.         Console.WriteLine(libro2.ToString());
228.     }
229. }

```