

Adonis Sena 20231935

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.

```
public class Persona

    private String nombre;

    private String apellidos;

    private int edad;

    private boolean casado;

    private String numeroDocumentoidentida

    public Persona(String nombre, String apellidos, int edad, boolean casado,
String numeroDocumentoidentidad) {

        this.nombre = nombre;

        this.apellidos = apellidos;

        this.edad = edad;

        this.casado = casado;

        this.numeroDocumentoidentidad = numeroDocumentoidentidad;

    }

    public String getNombre() {
```

```
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getApellidos() {
        return apellidos; }
    public void setApellidos(String apellidos) {
        this.apellidos = apellidos; }
    public int getEdad() {
        return edad; }
    public void setEdad(int edad) {    this.edad = edad;
    }
    public boolean isCasado() {
        return casado;
    }
    public void setCasado(boolean casado) {
        this.casado = casado;
    }
    public String getNumeroDocumentoldidentidad() {
        return numeroDocumentoldidentidad;
    }
    public void setNumeroDocumentoldidentidad(String
numeroDocumentoldidentidad) {
        this.numeroDocumentoldidentidad = numeroDocumentoldidentidad;
    }
```

```
public void enseñar() {  
    System.out.println(nombre + " está enseñando en una escuela.");  
}  
public void operar() {  
    System.out.println(nombre + " está realizando una cirugía.");  
}  
public void programar() {  
    System.out.println(nombre + " está escribiendo código en Java.");  
}  
public void pintar() {  
    System.out.println(nombre + " está pintando un paisaje.");  
}  
public void jugarFutbol() {  
    System.out.println(nombre + " está jugando fútbol.");  
}  
public void cantar() {  
    System.out.println(nombre + " está cantando en un concierto.");  
}  
public void escribir() {  
    System.out.println(nombre + " está escribiendo un libro.");  
}  
public static void main(String[] args) {  
  
    Persona persona1 = new Persona("Luis", "Martínez", 35, true,  
"123456789");
```

```

    persona1.enseñar();

    Persona persona2 = new Persona("Ana", "Gómez", 29, false,
"987654321");

    persona2.operar();

    Persona persona3 = new Persona("Carlos", "Sánchez", 40, true,
"456789123");

    persona3.programar();

    Persona persona4 = new Persona("María", "Lopez", 33, false,
"789123456");

    persona4.pintar();

    Persona persona5 = new Persona("Pedro", "Ramírez", 25, true,
"321654987");

    persona5.jugarFutbol();

    Persona persona6 = new Persona("Sofía", "Hernández", 30, false,
"654987321");

    persona6.cantar();

    Persona persona7 = new Persona("Juan", "Pérez", 50, true,
"159753486");

    persona7.escribir();
}
}

```

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 {

```

```
// Atributos de la clase
private String titular;
private String numeroCuenta;
private double saldo;
public Cuenta() {
    this.titular = "Desconocido";
    this.numeroCuenta = "0000000000";
    this.saldo = 0.0;
}
public Cuenta(String titular, String numeroCuenta, double saldo) {
    this.titular = titular;
    this.numeroCuenta = numeroCuenta;
    this.saldo = saldo;
}

// Getters y setters
public String getTitular() {
    return titular;
}
public void setTitular(String titular) {
    this.titular = titular;
}
public String getNumeroCuenta() {
    return numeroCuenta;
}
```

```

public void setNumeroCuenta(String numeroCuenta) {
    this.numeroCuenta = numeroCuenta;
}

public double getSaldo() {
    return saldo;
}

public void setSaldo(double saldo) {
    this.saldo = saldo;
}

```

Método para realizar ingresos

```

public void ingreso(double cantidad) {
    if (cantidad > 0) {
        saldo += cantidad;

        System.out.println("Ingreso de " + cantidad + " realizado con éxito.
Nuevo saldo: " + saldo);
    } else {
        System.out.println("La cantidad debe ser mayor a 0.");
    }
}

```

realizar reintegros

```

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

        System.out.println("Reintegro de " + cantidad + " realizado con éxito.
Nuevo saldo: " + saldo);
    }
}

```

```
    } else {  
        System.out.println("Fondos insuficientes o cantidad inválida.");  
    }  
}
```

transferencias a otra cuenta

```
public void transferencia(Cuenta cuentaDestino, double cantidad) {  
    if (cantidad > 0 && cantidad <= saldo) {  
        saldo -= cantidad;  
        cuentaDestino.ingreso(cantidad);  
        System.out.println("Transferencia de " + cantidad + " realizada con  
éxito a la cuenta: " + cuentaDestino.getNumeroCuenta());  
    } else {  
        System.out.println("Fondos insuficientes o cantidad inválida.");  
    }  
}
```

Método main para probar la clase

```
public static void main(String[] args) {  
    // Crear cuentas de ejemplo  
    Cuenta cuenta1 = new Cuenta("Juan Pérez", "1234567890", 5000.0);  
    Cuenta cuenta2 = new Cuenta("Ana López", "0987654321", 3000.0);  
  
    // Probar métodos  
    cuenta1.ingreso(1000);  
    cuenta1.reintegro(200);  
    cuenta1.transferencia(cuenta2, 1500);  
}
```

```
// Mostrar saldos finales  
System.out.println("Saldo final de la cuenta 1: " + cuenta1.getSaldo());  
System.out.println("Saldo final de la cuenta 2: " + cuenta2.getSaldo());  
}  
}
```

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 {  
    // Atributo de la clase  
    private int contador;  
  
    // Constructor por defecto  
    public Contador() {  
        this.contador = 0; // Valor inicial del contador  
    }  
  
    // Constructor con parámetros  
    public Contador(int contador) {  
        this.contador = contador;  
    }  
}
```



```
// Getter para obtener el valor del contador
```

```
public int getContador() {  
    return contador;  
}
```

```
// Setter para establecer el valor del contador
```

```
public void setContador(int contador) {  
    this.contador = contador;  
}
```

```
// Método para incrementar el contador
```

```
public void incrementar() {  
    contador++;  
    System.out.println("Contador incrementado. Valor actual: " + contador);  
}
```

```
// Método para decrementar el contador
```

```
public void decrementar() {  
    if (contador > 0) {  
        contador--;  
        System.out.println("Contador decrementado. Valor actual: " +  
contador);  
    } else {  
        System.out.println("El contador no puede ser menor que 0.");  
    }  
}
```

```
}
```

```
// Método main para probar la clase
```

```
public static void main(String[] args) {
```

```
    // Crear un objeto de la clase Contador con el constructor por defecto
```

```
    Contador contador1 = new Contador();
```

```
    contador1.incrementar(); // Incrementar el contador
```

```
    contador1.incrementar();
```

```
    contador1.decrementar(); // Decrementar el contador
```

```
    contador1.decrementar();
```

```
    contador1.decrementar(); // Intentar decrementar más allá de 0
```

```
    // Crear un objeto de la clase Contador con el constructor con  
    parámetros
```

```
    Contador contador2 = new Contador(10);
```

```
    contador2.incrementar();
```

```
    contador2.decrementar();
```

```
    System.out.println("Valor final del contador2: " +  
    contador2.getContador());
```

```
}
```

```
}
```

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 {
```

```
// Atributos de la clase
```

```
private String titulo;
```

```
private String autor;
```

```
private String isbn;
```

```
private boolean prestado;
```

```
// Constructor por defecto
```

```
public Libro() {
```

```
    this.titulo = "Desconocido";
```

```
    this.autor = "Desconocido";
```

```
    this.isbn = "0000000000";
```

```
    this.prestado = false;
```

```
}
```

```
// Constructor con parámetros
```

```
public Libro(String titulo, String autor, String isbn, boolean prestado) {
```

```
    this.titulo = titulo;
```

```
    this.autor = autor;
```

```
    this.isbn = isbn;
```

```
    this.prestado = prestado;
```

```
}
```

```
// Getters y setters
```

```
public String getTitulo() {
```

```
        return titulo;  
    }
```

```
public void setTitulo(String titulo) {  
    this.titulo = titulo;  
}
```

```
public String getAutor() {  
    return autor;  
}
```

```
public void setAutor(String autor) {  
    this.autor = autor;  
}
```

```
public String getIsbn() {  
    return isbn;  
}
```

```
public void setIsbn(String isbn) {  
    this.isbn = isbn;  
}
```

```
public boolean isPrestado() {
```

```
    return prestado;
}
```

```
public void setPrestado(boolean prestado) {
    this.prestado = prestado;
}
```

```
// Método para realizar préstamo
```

```
public void prestar() {
    if (!prestado) {
        prestado = true;
        System.out.println("El libro \"" + titulo + "\" ha sido prestado.");
    } else {
        System.out.println("El libro \"" + titulo + "\" ya está prestado.");
    }
}
```

```
// Método para realizar devolución
```

```
public void devolver() {
    if (prestado) {
        prestado = false;
        System.out.println("El libro \"" + titulo + "\" ha sido devuelto.");
    } else {
        System.out.println("El libro \"" + titulo + "\" no está prestado actualmente.");
    }
}
```

```

    }
}

// Método toString para mostrar información del libro
@Override
public String toString() {
    return "Libro [Título=" + titulo + ", Autor=" + autor + ", ISBN=" + isbn + ",
    Prestado=" + prestado + "];"
}

// Método main para probar la clase
public static void main(String[] args) {
    // Crear libros de ejemplo
    Libro libro1 = new Libro("El principito", "Antoine de Saint-Exupéry",
    "1234567890", false);

    Libro libro2 = new Libro("Cien años de soledad", "Gabriel García
    Márquez", "0987654321", true);

    // Probar métodos
    System.out.println(libro1.toString());
    libro1.prestar();
    libro1.devolver();

    System.out.println(libro2.toString());
    libro2.prestar(); // Intentar prestar un libro ya prestado

```

```
        libro2.devolver();  
    }  
}
```

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

```
public class Fraccion {  
    // Atributos de la clase  
    private int numerador;  
    private int denominador;  
  
    // Constructor por defecto  
    public Fraccion() {  
        this.numerador = 0;  
        this.denominador = 1; // No se permite denominador igual a 0  
    }  
  
    // Constructor con parámetros  
    public Fraccion(int numerador, int denominador) {  
        if (denominador == 0) {  
            throw new IllegalArgumentException("El denominador no puede ser  
cero.");  
        }  
        this.numerador = numerador;  
        this.denominador = denominador;  
    }  
}
```

```
}
```

```
// Getters y setters
```

```
public int getNumerador() {  
    return numerador;  
}
```

```
public void setNumerador(int numerador) {  
    this.numerador = numerador;  
}
```

```
public int getDenominador() {  
    return denominador;  
}
```

```
public void setDenominador(int denominador) {  
    if (denominador == 0) {  
        throw new IllegalArgumentException("El denominador no puede ser  
cero.");  
    }  
    this.denominador = denominador;  
}
```

```
// Método para sumar fracciones
```

```
public Fraccion sumar(Fraccion otra) {
```



```
        int nuevoNumerador = this.numerador * otra.denominador +  
otra.numerador * this.denominador;  
  
        int nuevoDenominador = this.denominador * otra.denominador;  
  
        return simplificar(new Fraccion(nuevoNumerador, nuevoDenominador));  
    }
```

// Método para restar fracciones

```
public Fraccion restar(Fraccion otra) {  
  
    int nuevoNumerador = this.numerador * otra.denominador -  
otra.numerador * this.denominador;  
  
    int nuevoDenominador = this.denominador * otra.denominador;  
  
    return simplificar(new Fraccion(nuevoNumerador, nuevoDenominador));  
}
```

// Método para multiplicar fracciones

```
public Fraccion multiplicar(Fraccion otra) {  
  
    int nuevoNumerador = this.numerador * otra.numerador;  
  
    int nuevoDenominador = this.denominador * otra.denominador;  
  
    return simplificar(new Fraccion(nuevoNumerador, nuevoDenominador));  
}
```

// Método para dividir fracciones

```
public Fraccion dividir(Fraccion otra) {  
  
    if (otra.numerador == 0) {
```

```
        throw new IllegalArgumentException("No se puede dividir entre  
cero.");  
    }  
  
    int nuevoNumerador = this.numerador * otra.denominador;  
    int nuevoDenominador = this.denominador * otra.numerador;  
    return simplificar(new Fraccion(nuevoNumerador, nuevoDenominador));  
}
```

// Método para simplificar la fracción

```
private Fraccion simplificar(Fraccion fraccion) {  
    int mcd = calcularMCD(fraccion.numerador, fraccion.denominador);  
    fraccion.numerador /= mcd;  
    fraccion.denominador /= mcd;  
    return fraccion;  
}
```

// Método para calcular el máximo común divisor (MCD)

```
private int calcularMCD(int a, int b) {  
    if (b == 0) {  
        return a;  
    }  
    return calcularMCD(b, a % b);  
}
```

// Método toString para mostrar la fracción

@Override

```
public String toString() {  
    return numerador + "/" + denominador;  
}
```

// Método main para probar la clase

```
public static void main(String[] args) {  
    Fraccion fraccion1 = new Fraccion(3, 4);  
    Fraccion fraccion2 = new Fraccion(2, 5);  
  
    System.out.println("Fracción 1: " + fraccion1);  
    System.out.println("Fracción 2: " + fraccion2);  
  
    Fraccion suma = fraccion1.sumar(fraccion2);  
    System.out.println("Suma: " + suma);  
  
    Fraccion resta = fraccion1.restar(fraccion2);  
    System.out.println("Resta: " + resta);  
  
    Fraccion multiplicacion = fraccion1.multiplicar(fraccion2);  
    System.out.println("Multiplicación: " + multiplicacion);  
  
    Fraccion division = fraccion1.dividir(fraccion2);  
    System.out.println("División: " + division);  
}
```

}

}