Nombre: [redacted] Padrón: [redacted] 5 / 10

75.07 / 95.02 - ALGORITMOS Y PROGRAMACIÓN III
PRIMER RECUPERATORIO DEL PRIMER PARCIAL - Jueves 3/11/2022

1) Dada la siguiente clase: 1 / 3

```
package ejercicio1;

public class Duracion {

  long calcularDias(int diaIni, int mesIni, int anioIni, int diaFin, int mesFin, int anioFin) {
    if (!esFechaValida(diaIni, mesIni, anioIni))
      throw new java.time.DateTimeException("Fecha incorrecta.");
    if (!esFechaValida(diaFin, mesFin, anioFin))
      throw new java.time.DateTimeException("Fecha incorrecta.");
    java.time.LocalDate fechaIni = java.time.LocalDate.of(anioIni, mesIni, diaIni);
    java.time.LocalDate fechaFin = java.time.LocalDate.of(anioFin, mesFin, diaFin);
    return fechaIni.until(fechaFin, java.time.temporal.ChronoUnit.DAYS);
  }

  private boolean esFechaValida(int dia, int mes, int anio) {
    try {
      java.time.LocalDate.of(anio, mes, dia);
    } catch (java.time.DateTimeException ex) {
      return false;
    }
    return true;
  }
}
```

Salida:

```
9
```

M a) explique cómo se la utilizaría desde otra clase para obtener la salida mostrada;

B b) indique si viola algún principio de diseño y, en caso afirmativo, indique cuál(es) y explique por qué lo hace;

M c) justifique si la clase se basa en algún patrón de diseño y, en caso afirmativo, indique en cuál(es).

2) Dado el siguiente diagrama de clases UML: 1 / 2

B a) indique qué relación representa el mismo;

M b) explique cómo dicha relación podría aparecer en un programa.

3) Dado el siguiente programa:

```
package ejercicio3;

public class Main {

  public static void main(String[] args) {
    Apellido apellido = new Apellido("peREz");
    Nombre nombre = new Nombre("jUaN");
    mostrar(nombre);
    System.out.print(" ");
    mostrar(apellido);
    System.out.println();
  }

  private static void mostrar(Apellido apellido) {
    System.out.print(apellido.toString());
  }

  private static void mostrar(Nombre nombre) {
    System.out.print(nombre.toString());
  }
}

class Cadena {
  private String valor;

  public String getValor() {
    return valor;
  }

  public void setValor(String valor) {
    this.valor = valor;
  }
}

class Apellido extends Cadena {
  public Apellido(String valor) {
    setValor(valor);
  }

  public String toString() {
    return getValor().toUpperCase();
  }
}

class Nombre extends Cadena {
  public Nombre(String valor) {
    setValor(valor);
  }

  public String toString() {
    return Character.toString(getValor().charAt(0)).toUpperCase() +
           getValor().substring(1).toLowerCase();
  }
}
```

a) describa su funcionamiento (en palabras o mediante un diagrama de secuencia);
b) indique si viola algún principio de diseño y, en caso afirmativo, indique cuál(es) y explique por qué lo hace;
c) justifique si el programa es candidato a ser refactorizado mediante algún patrón de diseño y, en caso afirmativo, indique mediante cuál(es) y explique cómo lo haría.

4) Dada la siguiente clase:

```
public class CajaAhorro {
    private int pesos;
    private int centavos;
    public CajaAhorro(){
        this.pesos = 0;
        this.centavos = 0;
    }
    public int getPesos() {
        return pesos;
    }
    public int getCentavos() {
        return centavos;
    }
    public void depositar(int pesos, int centavos) {
        //codigo
    }
    public void estraer(int pesos, int centavos) {
        //codigo
    }
}
```

Cuyas instancias almacenan el saldo de una caja de ahorro (que por definición jamas pueden quedar en descubierto, es decir, con saldo negativo, y en las cuales los valores de los atributos pesos y centavos siempre pueden pertenecer respectivamente, a los intervalos [0, 99999999] y [0, 99].

a) Escriba la menor cantidad de pruebas unitarias que deben considerarse para verificar el correcto funcionamiento de los métodos depositar y extraer, los cuales lanzan excepciones en caso de recibir, para sus parámetros pesos y centavos:

-argumentos que no pertenezcan a los intervalos

-argumentos simultaneamente 0

-argumentos que, en caso de llevarse a cabo la operacion, dejarian valores inválidos en los atributos

b) Escribir los algoritmos de los métodos depositar y extraer (puede usar pseudo código)