NOMBRE:
APELLIDOS:
NIA:
GRUPO:

Primer parcial

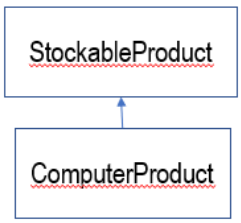
2ª Parte: Problemas (7 puntos sobre 10)

Duración: 70 minutos
Puntuación máxima: 7 puntos
Fecha: 25 de marzo de 2021

Instrucciones para el examen:

- No se permite el uso de libros o apuntes, ni tener teléfonos móviles u otros dispositivos electrónicos encendidos. Incumplir cualquiera de estas normas puede ser motivo de expulsión inmediata del examen.

Problema 1 (3 / 7 puntos)

El proyecto de gestión del almacén con el que has estado trabajando ha decidido crear una nueva línea de productos almacenables llamada ComputerProduct que representa productos informáticos que pueden estar diseñados para diferentes sistemas operativos. Esta nueva clase (ComputerProduct) supone una especialización de los productos anteriores (StockableProduct) según la jerarquía que se muestra en la figura.	

Suponiendo que ya está programada la clase StockableProduct y su constructor
public StockableProduct(String name, String brand, char category, boolean isCountable, String measurementUnit, int numUnits, double costPerUnit, double pricePerUnit, Provider provider)
se pide:

Apartado-1 Declaración de clase y Atributos (0,75 puntos)

- Declara la clase ComputerProduct que contenga un atributo operatingSystem de tipo character que represente el sistema operativo para el que está diseñado y crea (a modo de ejemplo) una de las tres constantes para representar los valores posibles que puede tomar dicho atributo WINDOWS-W, MAC-M y LINUX-L

Apartado-2 Método set y creación de excepciones (1,75 puntos)

- Crea un método public void setOperatingSystem(char operatingSystem) que asigne valor al atributo correspondiente asegurándose que el valor asignado es uno de los permitidos. En caso de que no lo sea debe lanzar la excepción OperatingSystemException.
- Crea la clase OperatingSystemException que herede de la clase Exception y tenga un constructor que reciba como parámetro el mensaje de error correspondiente.

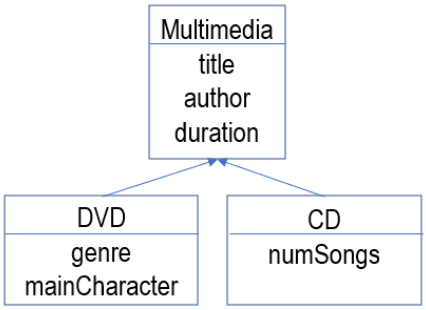
Apartado-3 Constructor (0,5)

- Crea un constructor que reciba todos los parámetros necesarios para asignar valor a todos los atributos de la clase ComputerProduct (incluidos los que hereda de su padre). NOTA: Toma como referencia el constructor de la clase StockableProduct.

Problema 2 (2,5 / 7 puntos)

Se quiere diseñar el sistema de gestión de películas de un videoclub a partir del siguiente diagrama de clases teniendo en cuenta la siguiente especificación

- Puedes asumir que existen los métodos set y get de todos los atributos de las clases representadas en la jerarquía, no es necesario que los programes.
- Puedes asumir que existen los métodos toString de las tres clases que devuelven un String con los resultados de los atributos.



Apartado 1. Declaración de clase atributos y constructores (0,6 puntos)

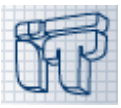
- Escribe la declaración de clase, atributos y constructores de la clase Multimedia y DVD que figuran en la jerarquía teniendo en cuenta que los atributos de cada clase sólo son accesibles desde la propia clase.

Apartado 2. Metodo findMultimediaDuration (0,7 puntos)

Programa el método
public static String findMultimediaDuration(Multimedia[] collection, int duration)
que dada una colección de elementos multimedia (CDs y DVDs) y una duración, devuelve como resultado todos aquellos elementos que tengan la duración indicada.

NOTA. Observa el código de la clase MultimediaApp que contiene un array de objetos multimedia con 4 DVDs y 3 CDs. La llamada al método que figura en la línea 14 para encontrar todos los objetos multimedia de duración 60 minutos devolvería dos objetos multimedia. El formato de los Strings viene dado por los métodos toString de las diferentes clases.

```
Title: Film-3 Duration: 60 director: Director-1 mainCharacter: Actor-3
Title: Soundtrack-1 Duration: 60 numSongs: 8
```



```
3  public class MultimediaApp {
4      public static void main(String[] args) {
5          Multimedia[] myCollection = {
6              new DVD("Film-1", 150, "Director-1", "Actor-1"),
7              new DVD("Film-2", 90, "Director-2", "Actor-2"),
8              new DVD("Film-3", 60, "Director-1", "Actor-3"),
9              new DVD("Film-4", 190, "Director-2", "Actor-2"),
10             new CD("Soundtrack-1", 60, 8),
11             new CD("Soundtrack-2", 90, 12),
12             new CD("Soundtrack-3", 80, 12)};
13
14         System.out.println(findMultimediaDuration(myCollection, 60));
15         System.out.println(findMaxDuration(myCollection));
16     }
17     public static String findMultimediaDuration(Multimedia[] collection,
18                                                 int duration) {
19        //your code here
20     }
21     public static String findMaxDuration(Multimedia[] collection) {
22        //your code here
23     }
24 }
```

Apartado 3. Método findMaxDuration (0,7 puntos)

Programa el método

public static String findMaxDuration(Multimedia[] collection)

que dada una colección de objetos multimedia devuelve el objeto de mayor duración.

NOTA. Observa el código de la clase MultimediaApp que contiene un array de objetos multimedia con 4 DVDs y 3 CDs. La llamada al método que figura en la línea 15 para encontrar el objeto multimedia de mayor duración devolvería el siguiente resultado de acuerdo a los métodos toString de las diferentes clases

```
Title: Film-4 Duration: 190 director: Director-2 mainCharacter: Actor-2
```

Apartado 4. Testing (0,5 puntos)

Dado el método String getMainCharacter() de la clase DVD. Escribe un Test unitario como los vistos en clase que cree un DVD interpretado por un actor llamado “Actor-1” y compruebe el funcionamiento correcto de su método get.

Problema 3 (1,5 / 7 puntos)

Programa el método recursivo public static int sumDigits(int num) que dado un número entero positivo (num) devuelve como resultado la suma de todos sus dígitos. Por ejemplo la llamada en la línea 6 con sumDigits (565) imprimiría como resultado 16 y con sumDigits(1234) daría como resultado 10.

NOTAS:

- No se permite la conversión de entero a String para resolver el problema.
- Puedes utilizar el hecho de que el número está escrito en base decimal.

```
3 public class SumDigits {
4     public static void main(String[] args) {
5         int number = Integer.parseInt(args[0]);
6         System.out.println(sumDigits(number));
7     }
8
9     public static int sumDigits(int num) {
10        //todo
11    }
12 }
```

SOLUCIÓN DE REFERENCIA (Varias soluciones son posibles)

Ejercicio 1. Solución (3 / 7 puntos)

Apartado-1 Declaración de clase y Atributos (0,75 puntos)

- 0,25 Declaración de la clase
- 0,25 Declaración del atributo category
- 0,25 Declaración de la constante

```
public class ComputerProduct extends StockableProduct {
    private char operatingSystem;
    public static char WINDOWS = 'W';
    public static char LINUX = 'L';
    public static char MAC = 'M';
}
```

Apartado-2 Método set y creación de excepciones (1,75 puntos)

- 1 Crea un método public void setOperatingSystem(char operatingSystem) que asigne valor al atributo correspondiente asegurándose que el valor asignado es uno de los permitidos. En caso de que no lo sea debe lanzar la excepción OperatingSystemException
- 0,75 Crea la clase OperatingSystemException

```
public void setOperatingSystem(char operatingSystem) throws OperatingSystemException{
    if (operatingSystem == WINDOWS || operatingSystem == MAC || operatingSystem == LINUX) {
        this.operatingSystem = operatingSystem;

}else {
    throw new OperatingSystemException("operating system non supported");
}
```

```
public class OperatingSystemException extends Exception {
    public OperatingSystemException(String message) {
        super(message);
    }
}
```

Apartado-3 Constructor (0,5)

- 0,2 declarar correctamente el constructor
- 0,2 por invocar correctamente a super()
- 0,1 por asignar correctamente el atributo

```
public ComputerProduct(String name, String brand, char category,
                       boolean isCountable, String measurementUnit,
                       int numUnits, double costPerUnit, double pricePerUnit,
                       Provider provider, char operatingSystem)
        throws OperatingSystemException {
    super(name, brand, category, isCountable, measurementUnit,
          numUnits, costPerUnit, pricePerUnit, provider);
    setOperatingSystem(operatingSystem);
}
```

Ejercicio 2. Solución (2,5 / 7 puntos)

Apartado 1.Declaración de clases atributos y constructores (0,6 puntos)

- 0,2 Multimedia (cabecera, atributos, constructores) . Se considera correcto si está hecho con this o con set
- 0,4 DVD (cabecera, extends, atributos, constructores, super)

```
public class Multimedia {
    private String title;
    private int duration;
    public Multimedia(String title, int duration) {
        setTitle(title);
        setDuration(duration);
    }
}
public class DVD extends Multimedia{
    private String director;
    private String mainCharacter;

    public DVD(String title, int duration,
               String director, String mainCharacter) {
        super(title, duration);
        this.director = director;
        this.mainCharacter = mainCharacter;
    }
}
```

Apartado 2. FindMultimediaDuration (0,7 puntos)

Apartado 2: (0,7 puntos)

- 0,2 Bucle FOR correcto
- 0.2 Condición correcta
- 0,3 Devolución del resultado y llamada correcta a toString()

```
public static String findMultimediaDuration(Multimedia[] collection,
                                            int duration) {
    String result = "";
    for (int i = 0; i < collection.length; i++) {
        if(collection[i].getDuration() == duration) {
            result = result + collection[i].toString() + "\n";
        }
    }
    return result;
}
```

Apartado 3: (0,7 puntos)

- 0,2 Definición correcta de la variables para sacar el mayor
- 0,2 bucle for
- 0,2 condición correcta
- 0,1 devolver el valor correcto

```
public static String findMaxDuration(Multimedia[] collection) {
    String result = "";
    Multimedia maxDuration = collection[0];
    for (int i = 1; i < collection.length; i++) {
        if(collection[i].getDuration() > maxDuration.getDuration()) {
            maxDuration = collection[i];
        }
    }

    return maxDuration.toString();
}
```

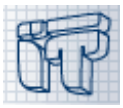
Apartado 4. Testing (0,5 puntos)

```
class DVDTest {
    @Test
    void testGetMainCharacter() {
        DVD myDVD = new DVD("Film-1", 150, "Director-1", "Actor-1");
        assertEquals("Actor-1", myDVD.getMainCharacter());
    }
}
```

Ejercicio 3. Solución (1,5 / 7 puntos)

Rúbrica

- (0,25) Caso trivial correcto (if num<10).
    - Penalizar con un 0 si no se pone el caso trivial correcto.
- (1,25) Caso Recursivo correcto.
    - 0,25 Si se retorna el valor correctamente haciendo el resto entre el número y 10
    - 1 Si se llama a la función de modo recursivo “SumDigits” correctamente



```java
public class SumDigits {
    public static void main(String[] args) {
        int number = Integer.parseInt(args[0]);
        System.out.println(sumDigits(number));
    }

    public static int sumDigits(int num) {
        if (num < 10) {
            return num;
        } else {
            return (num % 10) + sumDigits(num / 10);
        }
    }
}
```