UTN AVELLANEDA
Tecnicatura Universitaria en Programación

Materia:	Programación II		
Nivel:	2º Cuatrimestre		
Tipo de Examen:	Primer Parcial		
Apellido(1):		Fecha:	16 oct 2024 8:00 p.m.
Nombre/s(1):		Docente a cargo(2):	Carlos Arroyo Díaz
División(1):		Nota(2):	
DNI(1):		Firma(2):	

(1) Campos a completar solo por el estudiante en caso de imprimir este enunciado en papel.
(2) Campos a completar solo por el docente en caso de imprimir este enunciado en papel.

Enunciado:

Sistema de gestión de una tienda de música

De acuerdo a las descripciones de las siguientes clases, se pide:

1. **Modelar en UML** (adjuntando la imagen en la entrega) y posteriormente crear el código correspondiente en JAVA (en todos los casos, reutilizar código). **[1 pto]**

2. Realizar la respectiva organización del código en **paquetes** apropiados. **[0.5 pto]**

3. Generar un proyecto en JAVA nombrado como: **[3 ptos]**

Apellido_Nombre, que sea de tipo Biblioteca de Clases, el cual tendrá las siguientes clases (todas en el paquete apellido.nombre)

Artista posee todos sus atributos privados y de tipo cadena de caracteres: nombre y generoMusical.

Un único constructor que inicializa dichos atributos y métodos:

• sonIguales (Artista, Artista). Retornará *true*, si los nombres y los géneros musicales son iguales, *false*, caso contrario. Este método será público y estático.

• getNombreGenero(). Este método público y de instancia, retornará una cadena de caracteres que contengan el nombre y género musical del artista, separados por dos puntos. Ej.: Madonna:Pop.

Disco: Esta clase no podrá instanciarse y todos sus atributos son protegidos:

• artista, variable escalar de tipo Artista.

• anioLanzamiento, de tipo entero.

● titulo, de tipo cadena de caracteres.

● precio, de tipo flotante.

● generadorDeAnio, atributo de clase de tipo Random

Posee un bloque estático (que inicializará el atributo generadorDeAnio) y un constructor de instancia para inicializar los siguientes atributos:

● (titulo, precio, artista)

● (titulo, precio, nombre, generoMusical) Reutilizar código.

Como métodos tendrá:

● getAnioLanzamiento, retornará el valor correspondiente del atributo anioLanzamiento, que se inicializará en dicha propiedad, si y sólo si, su valor es cero. Para inicializar dicho atributo, se utilizará el atributo estático generadorDeAnio (valores aleatorios entre 1950 y 2024).

● getPrecio, que retornará el precio del disco.

● El método privado y de clase mostrar(Disco), retorna una cadena detallando todos los atributos del parámetro de tipo Disco que recibe. Reutilizar código.

● sonIguales(Disco, Disco), método de clase que retorna true, si al comparar dos objetos de tipo Disco, los títulos y artistas son iguales, false, caso contrario.

● Sobrescritura del método equals(), que retorna true, si al comparar dos objetos de tipo Disco, los títulos y artistas son iguales, false, caso contrario. Reutilizar código.

● Sobrescritura del método toString(), retornará el detalle completo del disco. Reutilizar código.

4. También tendrá las siguientes **clases derivadas** de Disco: **[3 ptos]**

Vinilo: posee un único atributo de tipo Velocidad (velocidad), que será inicializado por su único constructor.

Velocidad es un enumerado (agregar el archivo correspondiente) que posee las siguientes enumeraciones:

● RPM_33

● RPM_45

● RPM_78

Sobrescritura del método *toString()*. Retornará una cadena de caracteres conteniendo la información completa del objeto. Reutilizar código. Sobrescritura del método *equals()*, que retorna *true*, si el parámetro recibido es igual a la instancia actual (ambos discos son iguales) y las velocidades son iguales, *false*, caso contrario. Reutilizar código.

CD posee un único atributo propio de tipo Formato (formato), que será inicializado por su único constructor.

Formato es un enumerado (agregar el archivo correspondiente) que posee las siguientes enumeraciones:

- ESTANDAR
- DOBLE
- REMASTERIZADO

Sobrescritura del **método toString().** Retornará una cadena de caracteres conteniendo la información completa del objeto. Reutilizar código.

Sobrescritura del **método equals()**, que retorna *true*, si el parámetro recibido es igual a la instancia actual (ambos discos son iguales) y los formatos son iguales, *false*, caso contrario. Reutilizar código.

La última clase que tendrá el proyecto será **TiendaMusica**. Dicha clase posee dos atributos, ambos privados. Uno indicará la capacidad máxima que tendrá la tienda para almacenar discos (capacidad). El otro es una colección de tipo Disco (discos).

El constructor por defecto será el único que inicializará la lista de discos y establecerá la capacidad máxima en 5 discos, mientras que la sobrecarga que recibe un parámetro de tipo entero, inicializará la capacidad de la tienda. Reutilizar código.

Métodos de instancia:

sonIguales(Disco), retornará *true*, si es que el disco ya se encuentra en la tienda, *false*, caso contrario. Reutilizar código.

agregar(Disco), si la tienda posee capacidad de almacenar al menos un disco más y dicho disco no se encuentra en la tienda, lo agrega a la colección, caso contrario, informará lo acontecido. Reutilizar código.

Método privado y de instancia **getPrecio(TipoDisco)**, retornará el valor de la tienda de acuerdo con el enumerado que recibe como parámetro.

TipoDisco es un enumerado (agregar el archivo correspondiente) que posee las siguientes enumeraciones:

- VINILOS
- CDS
- TODOS

Agregar los métodos públicos y de instancia:

- getPrecioDeVinilos(), retornará un valor doble precisión que representa el precio de todos los vinilos de la tienda.
- getPrecioDeCDs(), retornará un valor doble precisión que representa el precio de todos los CDs de la tienda.
- getPrecioTotal() , retornará un valor doble precisión que representa el total de los discos de la tienda.

En todos los casos, reutilizar código.

El método público de clase **mostrar(TiendaMusica)**, retorna una cadena de caracteres con toda la información de la tienda que recibe cómo parámetro, incluyendo el detalle de cada uno de sus discos. Reutilizar código.

5. Aplique validaciones en donde crea necesario y por los menos dos excepciones personalizadas.

[2 ptos]

6. Copiar las siguientes líneas de código en el método main (de la clase Principal), debe hacer las modificaciones para que funcione el programa (try-catch) :

[0.5 pto]

```
TiendaMusica miTienda = new TiendaMusica(7);

Artista a = new Artista("The Beatles", "Rock");

Artista b = new Artista("Michael Jackson", "Pop");

Vinilo v1 = new Vinilo("Abbey Road", 50f, "The Beatles", "Rock", Velocidad.RPM_33);

CD c1 = new CD("Thriller", 30f, b, Formato.ESTANDAR);

Vinilo v2 = new Vinilo("Let It Be", 45f, "The Beatles", "Rock", Velocidad.RPM_45);

CD c2 = new CD("Bad", 28f, b, Formato.REMASTERIZADO);

CD c3 = new CD("Dangerous", 32f, b, Formato.DOBLE);
```

```
CD c4 = new CD("Off The Wall", 25f, "Michael Jackson", "Pop", Formato.ESTANDAR);

miTienda.agregar(v1);

//YA INGRESADO

miTienda.agregar(v1);

miTienda.agregar(c1);

miTienda.agregar(v2);

miTienda.agregar(c2);

miTienda.agregar(c3);

//SIN LUGAR

miTienda.agregar(c4);

System.out.println("");

//TRUE

System.out.println(v1.equals(v1));

//FALSE

System.out.println(v1.equals("The Beatles"));

//FALSE

System.out.println(v1.equals(v2));

//TRUE

System.out.println(c1.equals(c1));

//FALSE

System.out.println(c1.equals(c2));

//FALSE

System.out.println(c1.equals(c4));

System.out.println("");

System.out.println(TiendaMusica.mostrar(miTienda));
```

SALIDA DE CONSOLA

```
El disco ya existe en la tienda

true
false
false
true
false
false

Tienda de Musica:
Capacidad: 7
```

```
Discos:
Vinilo{Disco{artista=Artista{nombre='The Beatles', generoMusical='Rock'}, anioLanzamiento=1956, titulo='Abb
CD{Disco{artista=Artista{nombre='Michael Jackson', generoMusical='Pop'}, anioLanzamiento=2002, titulo='Thri
Vinilo{Disco{artista=Artista{nombre='The Beatles', generoMusical='Rock'}, anioLanzamiento=1998, titulo='Let
CD{Disco{artista=Artista{nombre='Michael Jackson', generoMusical='Pop'}, anioLanzamiento=1982, titulo='Bad'
CD{Disco{artista=Artista{nombre='Michael Jackson', generoMusical='Pop'}, anioLanzamiento=2004, titulo='Danc
CD{Disco{artista=Artista{nombre='Michael Jackson', generoMusical='Pop'}, anioLanzamiento=2000, titulo='Off
```

```
Precio total de vinilos: 95.0
Precio total de CDs: 115.0
Precio total de la tienda: 210.0
------------------------------------------------------------------------
BUILD SUCCESS
```

NOTA: Los únicos valores que podrán cambiar son los que indican el año de lanzamiento de cada disco, ya que son valores aleatorios de entre 1950 y 2024.

Importante:

1) Dos (2) errores en el mismo tema anulan su puntaje.
2) NO se corregirán exámenes que NO compilen.
3) NO se corregirán exámenes que NO contengan la imagen del modelado en UML.
4) No sé corregirán proyectos que no sea identificable su autor.
5) Reutilizar tanto código como sea posible.
6) Colocar nombre de la clase (en estáticos), this o super en todos los casos que corresponda.
7) Subir el proyecto y la imagen UML en un único archivo .7z, .zip, .rar o similar. Nombrarlo con su Apellido.Nombre

Duración:

120 minutos