

## Conceptos fundamentales de Java

### Sección 7, parte 2: Creación de un proyecto de inventario

### Solución del proyecto

#### Visión general

Este proyecto avanzará al ritmo del usuario en las secciones 4, 5, 6 y 7 del curso. Después de cada sección, se podrán realizar más aportaciones hasta que se cree una aplicación Java completa para mantener el inventario. En cada parte, tome como base la última parte de modo que se cumplan tanto los requisitos anteriores como los nuevos. Incluya todas las partes en un paquete denominado inventario.

Cree un programa de inventario que se pueda utilizar para una serie de productos distintos (CD, DVD, software, etc.).

#### Tema(s):

- Trabajar con subclases (sección 7.4)
  - uso de extensiones
  - uso de super()
  - Sustitución de métodos de una superclase
- Actualización de la interfaz de usuario (sección 4.2)

#### Escenario

Se ha corrido la voz sobre su software de inventario y una empresa que vende exclusivamente CD o DVD ha recurrido a usted. Les gustaría personalizar el software para que pueda almacenar la duración, la clasificación de edad y el estudio cinematográfico para los DVD, así como el artista, el número de canciones y la discográfica de sus CD.

1. Abra el programa de inventario que se ha actualizado en la **sección 7, parte 1: Creación de un proyecto de inventario**
  - a. Antes de empezar con la programación, puede ser útil actualizar el diseño de las tablas. Las siguientes tablas muestran los campos que pertenecen a la clase Product (□) y los campos son específicos de un DVD (■).

- i. Complete una tabla de datos de ejemplo para realizar una lista de películas en DVD que prefiera.

DVD	
Atributo	Datos de ejemplo
Nombre del producto	Daredevil
Precio	8,99
Cantidad de unidades en existencias	50
Número de elemento	1
Duración (minutos)	99
Clasificación de edad	15
Estudio cinematográfico	20 <sup>th</sup> Century Fox

- ii. Complete una tabla de datos de ejemplo para realizar una lista de CD que prefiera.

CD	
Atributo	Datos de ejemplo
Nombre del producto	Dreams we never lost
Precio	7,99
Cantidad de unidades en existencias	50
Número de elemento	2
Artista	Tidelines
Número de canciones	14
Discográfica	Tide Lines Music

2. Ahora implementará la herencia mediante la creación de subclases de la clase Product.
  - a. Cree una subclase de la clase Product denominada **DVD** que tenga campos de instancia adicionales para almacenar la duración de las películas en minutos, la clasificación de edad y el estudio cinematográfico que ha sacado la película.
  - b. Cree un único constructor que acepte valores para cada campo de instancia para las clases DVD y Product. Utilice la llamada super() en el constructor en Product y transfiera los parámetros necesarios.
  - c. Cree getters y setters para los campos de instancia de DVD.
  - d. Siga exactamente el mismo proceso para crear una subclase de Product denominada CD. Cree los campos de instancia, el constructor, y getters y setters.
3. En la subclase DVD, sustituya el método para calcular el valor del inventario de un DVD con el mismo nombre que el método creado anteriormente en la clase de producto. El método de subclase DVD debe sumar también una tasa de devolución al almacén del 5 % al valor del inventario de ese producto. Tendrá que obtener los valores de precio y cantidad en existencias de la superclase.
4. Ahora definirá el resultado para los objetos DVD y CD.
  - a. Sustituya el método toString() de la clase Product de modo que toda la información sobre los objetos de la nueva subclase (DVD) pueda mostrarse en el resultado de la consola. Utilice los métodos getter para los valores en lugar de hacer referencia a los campos directamente, esto obliga a usar la versión local de los métodos que puedan existir en lugar de los métodos en la superclase. El resultado se debe parecer a lo siguiente:

```

Item Number      : 1
Name             : Daredevil
Movie Length     : 99
Age Rating       : 15
Film Studio      : 20th Century Fox
Quantity in stock: 50
Price            : 8.99
Stock Value      : 471.975
Product Status   : Active

```

- b. Haga lo mismo en la clase CD de forma que tenga el siguiente resultado:

```

Item Number      : 2
Name             : Dreams we never lost
Artist           : Tidelines
Songs on Album   : 14
Record label     : Tide Lines Music
Quantity in stock: 50
Price            : 7.99
Stock Value      : 399.5
Product Status   : Active

```

5. Modifique la clase ProductTester para rellenar la matriz de productos con objetos CD o DVD.
- Copie el método addToInventory y péguelo directamente debajo del original. Cambie el nombre del método a addCDToInventory.
  - Agregue variables temporales adicionales para incluir los campos de instancia que ha agregado a la clase CD.
  - Actualice las solicitudes para pedir al usuario la información en el siguiente orden:

```

Please enter the CD name:
Please enter the artist name:
Please enter the record label name:
Please enter the number of songs:
Please enter the quantity of stock for this product:
Please enter the price for this product:
Please enter the item number:

```

Puede que tenga que limpiar el buffer de entrada antes de solicitar valores.

- La última línea en el método crea un objeto de producto en la matriz de productos; la tendrá que actualizar.
  - Actualmente, la posición en la matriz se identifica con la variable i. Como este método ya no está en el bucle for, no reconoce i como una variable. Para resolver este problema, agregue i como parámetro al final de la firma del método.
  - Ahora debe utilizar el polimorfismo para crear un objeto CD en lugar de un objeto de producto genérico. Actualice el código para identificar CD como el tipo de objeto después de la sentencia = new.
  - Aparece un error al hacer esto puesto que la lista de argumentos no coincide con la lista de parámetros. Actualice los parámetros para que coincidan con la lista de parámetros para el constructor CD.
- Siga el mismo proceso para crear un objeto DVD. Las solicitudes se deben mostrar en el siguiente orden:

```

Please enter the DVD name:
Please enter the film studio name:
Please enter the age rating:
Please enter the length in minutes:
Please enter the quantity of stock for this product:
Please enter the price for this product:
Please enter the item number:

```

6. Ahora, es necesario que actualice el código en el método `addInventory` para permitir que el usuario pueda agregar un CD o DVD.

- a. Elimine todas las variables locales existentes y sustitúyalas por una única variable denominada `stockChoice` que almacene un valor entero y se pueda inicializar en menos 1.
- b. Solicite al usuario un valor para `stockChoice` permitiéndole elegir entre objetos CD o DVD con la visualización del siguiente menú:

```
1: CD
2: DVD
Please enter the product type:
```

- c. Si el usuario introduce un valor entero inferior a 1 o superior a 2, aparecerá el siguiente mensaje de error:

```
Only numbers 1 or 2 allowed!
```

- d. Se debe volver a solicitar al usuario que introduzca una entrada válida y se deben gestionar todas las entradas incorrectas.
- e. Después de la sentencia `while`, agregue una sentencia de condición adecuada que permita al usuario agregar un CD si introduce 1 o un DVD si introduce 2. Utilice el método adecuado para agregar que ha creado anteriormente.

7. Ejecute y pruebe el código.

8. No permita al usuario que agregue existencias a una línea de productos descatalogados.

Actualice el método `addToInventory` en la clase **Product** para que deje de agregar existencias a una línea descatalogada.

9. Ejecute y pruebe el código.

10. Guarde el proyecto.