

Conceptos fundamentales de Java

Sección 5: 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):

- Entrada del teclado/escáner (sección 5.1)
- Campos booleanos y operadores ternarios (sección 5.1)

1. Abra el programa de inventario que se ha creado en la **sección 4: Creación de un proyecto de inventario**
2. Modifique la clase ProductTester.
 - a) Agregue un escáner denominado **in** al principio de su método main.
 - b) Cree las variables locales que almacenarán los valores de cada uno de los atributos de la clase Product. Asigne a las variables el nombre tempNumber, tempName, tempQty y tempPrice.
 - c) Pida al usuario que introduzca valores para cada uno de los atributos de la clase Product. Solicite el nombre, la cantidad, el precio y el número de elemento, y almacene los valores en las variables locales temporales que acaba de crear.
 - d) Utilice los valores que ha introducido el usuario para crear el objeto p1. Esto significa que utilizará el constructor que toma 4 parámetros en lugar del constructor predeterminado.
3. Pida al usuario que le proporcione valores para p2.
 - a) Utilice las mismas variables locales que antes para obtener la entrada del usuario para crear el objeto p2. Copie y pegue el código después de la línea que crea el objeto p1.
 - b) Ejecute el programa e identifique dónde se ha producido un error.
 - c) El programa no aparece para solicitarle un valor para el nombre. Esto se debe a que el último valor introducido ha sido un valor numérico y ha dejado algunos caracteres especiales en el buffer de entrada. Para limpiar el buffer de entrada, agregue la siguiente sentencia antes de solicitar valores para p2:

```
in.nextLine();
```

Esto obtiene valores almacenados en el buffer y los descarta dejando un buffer vacío.
 - d) Ejecute el programa ahora, no debería presentar errores y debería mostrar todos los valores, incluidos los que ha introducido el usuario en la consola.
 - e) Cierre el objeto Scanner cuando termine con él.

4. Querrá poder marcar los productos como activos o descatalogados. Si se descataloga un producto, significa que las existencias restantes de dicho producto serán las últimas y no se realizarán más pedidos.
 - a) Agregue una instancia boolean a la clase Product denominada Activo que tenga un valor true predeterminado.
 - b) Cree métodos getter/setter para este nuevo campo.
 - c) Agregue el valor de este nuevo campo al método toString() para que la salida coincida con lo siguiente:

```

Item Number      : 1
Name             : Greatest Hits
Quantity in stock: 25
Price            : 9.99
Product Status   : true

```

5. Al ejecutar el código, se visualiza el valor true o false para Activo. Esto no es fácil de usar y lo mejor sería que la salida indicara Activo (true) o Descatalogado (false). Para ello, agregue un operador ternario en el método toString().
6. Llame al setter de la clase de controlador y defina el valor activo en false para el objeto p6 antes de visualizar los valores en la pantalla. Ejecute y pruebe el código.
7. Cree un método en la clase Product que devolverá el valor de inventario para cada elemento. Utilice el precio del producto multiplicado por la cantidad de existencias para calcular el valor de inventario. No utilice ninguna variable local en este método; solo tiene que devolver el valor en una única línea de código.
8. Actualice el método toString() en la clase Product para incluir una llamada al método en el método getInventoryValue() que acaba de crear para que la salida sea de la siguiente forma:

```

Item Number      : 1
Name             : Greatest Hits
Quantity in stock: 25
Price            : 9.99
Stock Value      : 249.75
Product Status   : true

```

9. Guarde el proyecto.