

Conceptos fundamentales de Java

Sección 6: Creación de un proyecto de inventario

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):

- Uso de bucles (secciones 5.2 y 6.1)
- Manejo de excepciones (sección 6.2)
- Uso de sentencias if (sección 5.1)
- Matrices de objetos (sección 6.1)

1. Abra el programa de inventario que se ha actualizado en la **sección 5: Creación de un proyecto de inventario**

2. Pida al usuario que introduzca el número de productos que desea agregar. Acepte un entero positivo para representar el número de productos y maneje el valor de cero.

- a) Cree una variable denominada `maxSize` que pueda almacenar valores enteros.
- b) Cree una solicitud al principio del método `main` que instruya al usuario que introduzca el valor necesario para el número de productos que se quiere almacenar:

```
Enter the number of products you would like to add
Enter 0 (zero) if you do not wish to add products:
```

- c) Utilice un bucle `do while` para que el programa no continúe hasta que se introduzca un valor positivo válido. Si se introduce un valor inferior a cero, debería aparecer un mensaje de error que indique "Valor incorrecto introducido" antes de que se vuelva a solicitar al usuario que introduzca un nuevo valor. No debe salir del bucle hasta que se introduzca un valor igual a cero o superior.

3. Ahora agregará el manejo de algunos errores para tratar los errores de tiempo de ejecución en el código. Actualmente, el programa se encarga de números introducidos fuera del rango determinado, pero no puede manejar entradas de tipos de datos incorrectos.

- a) Agregue un bloque `try` que incluya todo el código dentro del bucle `do while`.
- b) Agregue una sentencia `catch` por encima del `while` que utilizará un parámetro `Exception e`. El programa debe utilizar una sentencia de salida de la consola para mostrar el valor de `e` en la pantalla.
- c) Ya que ahora asigna un valor para `maxSize` dentro de una sentencia `try`, existe la posibilidad de que no se asigne un valor a `maxSize` al llegar a la cláusula `while`. Para asegurarse de que esto no ocurra, asigne un valor inicial de `-1` para `maxSize` cuando se declare. **INDICACIÓN:** Asigne siempre un valor que falle en el bucle para que el código esté obligado a asignar un valor correcto antes de continuar.

- d) Ejecute y pruebe el código introduciendo un carácter en lugar de un número.
 - e) Agregue una línea del código de la sentencia catch que limpie el buffer de entrada para que se muestre la solicitud y el sistema espere la entrada del usuario.
 - f) Anote el tipo específico de excepción producida cuando introduzca un carácter y cree una sentencia catch solo para esa excepción. Con este error se debería mostrar el mensaje **Tipo de datos incorrecto introducido.** en la consola y también limpiar el buffer de entrada.
 - g) Ejecute y pruebe el código introduciendo distintos valores de entrada.
4. Modifique la clase ProductTester para manejar varios productos mediante una matriz unidimensional si se introduce un valor superior a cero.
- a) Cree una sentencia if que muestre el mensaje "No se requieren productos." en la consola si el valor de maxSize es cero.
 - b) Agregue una sentencia Else para tratar con cualquier valor distinto de cero. Cree una única matriz unidimensional denominada productos en función de la clase Product que tendrá el número de elementos especificado por el usuario en la variable maxSize.
5. Ahora rellenará la matriz con los valores del usuario para cada campo en un objeto de producto.
- a) Dentro de la sentencia else debajo de donde ha creado la matriz, escriba un bucle for que iterará en la matriz de cero a 1, inferior a maxSize.
 - b) Como la última entrada que ha recibido del usuario era numérica, tendrá que agregar una sentencia que limpie el buffer de entrada como la primera línea en el bucle for.
 - c) Copie el código que ha utilizado para obtener la entrada del usuario para todos los campos de producto en el bucle for. Esto incluye el nombre, la cantidad, el precio y el número de elemento.
 - d) Agregue un nuevo objeto de producto en la matriz con el valor de índice para la posición y el constructor que tome 4 parámetros.
6. Utilice un bucle for each para visualizar la información para cada producto individual en la matriz de productos.
7. Elimine el código innecesario que no se utilice en este ejercicio.
8. Guarde el proyecto.