Examen Entornos de Desarrollo

1º DDAM/DDAW

Miércoles 2 febrero 2022 - 15:00

Contenido

[Ejercicio 1 (3p) 1](#_Toc94692564)

[Ejercicio 2 (7p) 1](#_Toc94692565)

[Especificaciones del código 2](#_Toc94692566)

ENTREGA: Consistirá en un fichero comprimido que debe incluir:

* Documento PDF con las explicaciones y capturas necesarias para responder las preguntas del examen
* Archivo ZIP correspondiente a la exportación del proyecto de Netbeans. En él deben estar presentes las modificaciones de código debidamente comentadas y los puntos de interrupción usados para la depuración

# Ejercicio 1 (3p)

1. Personaliza el entorno de desarrollo para que el proyecto se guarde en AlumnodDDAM/Documentos/examenEDTuNombre\_Apellidos.
2. Haz que los errores aparezcan en color Magenta. Realiza capturas de pantalla del proceso y Resultado
3. Instala el plugin que se te proporciona de forma offline. Realiza capturas de pantalla del proceso.
4. Crea un proyecto donde vas a probar una clase **Tienda.java** de la que vas a realizar pruebas unitarias de funcionamiento mediante la clase **TiendaTestStock.java**. Los ficheros .java deben ubicarse correctamente dentro del proyecto.

Puntuación Ejercicio 1:

1. 0,5p
2. 0,5p
3. 1p
4. 1p

# Ejercicio 2 (7p)

1. Crear un grafo que represente el código a probar del método **pedirStock**.
2. Depura el código del **método pedirStock** y, si es necesario, modifícalo para que funcione de acuerdo con lo expresado en el enunciado. (ver Especificaciones del código) **(Se debe incluir una captura de la depuración detenida en cada uno de los 5 casos que se asigne valor a la variable iCodErr2)**
3. Realiza un análisis de caja negra, incluyendo valores límite y conjetura de errores del **método** **vender**. Debes considerar que este método recibe como parámetro la cantidad a vender, que no podrá ser menor a 0 ni mayor al stock disponible. Al tratarse de pruebas funcionales no es necesario conocer los detalles del código, pero te lo pasamos para que lo tengas.
4. Crea o utiliza la clase **TiendaTestStock** del tipo Caso de prueba **JUnit 4 en netbeans** que nos permita pasar las pruebas unitarias de caja blanca del método **pedirStock**. Esta clase debes corregirla de modo que funcione adecuadamente, para ello debes definir 4 casos de prueba para los casos para que **iCodErr2 valga: 0,2,3 y 4** y aplicarlos en el código de la prueba. Copia el código fuente correspondiente que te proporcionamos si así lo deseas. **(Se debe incluir una captura con todas las pruebas que se piden superadas)**

Puntuación Ejercicio 2:

1. 1 p
2. 2 p
3. 2 p
4. 2 p

# Especificaciones del código

/\*\*

\*

\* @author prof entornos desarrollo

\* La clase tienda se dedica únicamente a la venta de camisetas

\* Tiene el método vender donde se indica el número de camisetas vendidas y el

\* método pedirStock que solicita más stock de camisetas a los proveedores

\* cuando el stock de producto sea menor a 3

\*

\* Tenemos también el método getBeneficio que indica cuánto ganamos con

\* la venta de cada camiseta. El beneficio no puede ser menor que 0. Pero no se va

\* a utilizar en esta prueba

\*

\* Se comienza con un saldo inicial de 1000€ para poder realizar compras y ventas

\* No podemos dejar la caja con menos de 100€ ni tener saldos negativos en la caja

\* No podemos vender camisetas si no las tenemos en stock

\* No podemos pedir un número negativo de camisetas

\* No podemos vender un número negativo de camisetas

\*/

/\*\*

\* Indica el número de camisetas vendidas. No se puede vender un número negativo de

\* camisetas

\* ni más camisetas de las disponibles en stock

\* @param numCamisetas indica el número de camisetas que se venden en esa transacción

\* @return iCodErr código de error que indica porqué ha terminado el programa

\* 0 indica que no ha habido error

\*/

public int vender(int numCamisetas){

/\*\*

\* método pedirStock que solicita más stock de camisetas a los proveedores

\* cuando el stock de producto sea menor a 3

\* No podemos pedir un número negativo de camisetas

\* No podemos dejar la caja con menos de 100€ ni tener saldos negativos en la

\* caja

\*

\* @param num indica el número de camisetas que se compran en esa transacción

\* @return iCodErr2 código de error que indica porqué ha terminado el programa

\* 0 indica que no ha habido error

\*

\*/

public int pedirStock(int num){