Formulario Práctica JUnit

Debe rellenar este formulario y subirlo al Blackboard de la asignatura para poder evaluar la práctica.

1. Con la información proporcionada para la realización de la práctica, ¿qué tipo de pruebas se llevan a cabo en la práctica? Justifique su respuesta:

En la práctica, hemos llevado a cabo principalmente pruebas de caja blanca, pero también realizamos pruebas de caja negra en la segunda parte. Durante la ejecución de la práctica, nos enfocamos en realizar pruebas que abarcaran tanto el funcionamiento interno del sistema (pruebas de caja blanca) como su comportamiento externo (pruebas de caja negra). Este enfoque integral nos permitió evaluar exhaustivamente el software desde diferentes perspectivas, lo que contribuyó a garantizar su calidad y fiabilidad.

1. En el ejercicio 2 de la práctica ejecutó la batería de pruebas pruebasINSW3.jar, ¿cuántos tests se realizaron con la batería? ¿qué resultados obtuvo? ¿qué deduce a partir de los resultados obtenidos?

Se llevaron a cabo seis pruebas en la calculadora, y también se realizó un "repetition test" cinco veces. Todos los resultados de la calculadora fueron positivos. Tras realizar estas pruebas pertinentes, se puede deducir que la clase de la calculadora está en perfectas condiciones. Durante la evaluación de la calculadora, se realizaron múltiples pruebas, incluyendo un "repetition test" repetido en varias ocasiones para garantizar la fiabilidad de los resultados. Tras un exhaustivo análisis de los resultados de todas las pruebas, se puede concluir que la clase de la calculadora ha demostrado un desempeño consistente y confiable, lo que respalda su calidad y funcionalidad.

1. En el ejercicio 3, ha creado dos tests y uno de ellos produce error. ¿Cuál de ellos? ¿Qué quiere decir el error que produce el test ejecutado?

En el ejercicio tres, de las dos clases, la que presenta fallas es la clase de Gestor. Se identificaron deficiencias en la función de añadir producto y en la función de añadir producto con cantidad excesiva. Tras realizar un análisis exhaustivo, se concluyó que estas funciones no están correctamente implementadas. Durante la evaluación de las clases, se detectaron problemas específicos en la funcionalidad de la clase de Gestor, lo que sugiere la necesidad de revisar y corregir las implementaciones de las funciones mencionadas. Este hallazgo resalta la importancia de realizar pruebas detalladas y exhaustivas para identificar posibles deficiencias en el código y garantizar la calidad del software.

1. ¿En qué consiste el criterio de clases de equivalencia y cómo lo ha aplicado para realizar su batería de pruebas?

Al aplicar el criterio de clases de equivalencia para diseñar mi batería de pruebas, identifiqué distintas categorías de entradas basadas en requisitos y especificaciones. Agrupé estas entradas en clases de equivalencia, cada una representando un conjunto con un comportamiento similar. Luego, seleccioné casos de prueba representativos para cada clase, asegurándome de abarcar valores límite y dentro del rango. Al ejecutar esta batería de pruebas, pude evaluar eficientemente el comportamiento del sistema en diversas situaciones, registrando resultados para análisis posterior. Este enfoque optimizado facilitó la identificación de patrones, tendencias y áreas críticas del sistema, proporcionando una cobertura efectiva con un conjunto de pruebas bien estructurado. Este método de diseño de pruebas basado en clases de equivalencia permitió una evaluación exhaustiva del sistema, lo que resultó en una mayor confianza en la calidad y fiabilidad del software desarrollado

1. Complete la siguiente tabla con los tests que ha realizado en su batería de pruebas, justifique la utilidad del test realizado y los resultados obtenidos en cada uno de los tests. Puede añadir todas las filas que considere necesarias en la tabla:

| Test | Justificación del test | Resultado |
| --- | --- | --- |
| Se añade un producto correctamente | Comprobar funcionalidad básica del módulo | Bien |
| Se lanza excepción al introducir producto = null | Comprobar que al introducir un elemento null en el almacén se lanza excepción | Bien |
| añadir producto con cantidad excesiva | Verificar si es posible agregar más cantidad a un producto de la que debería tener. | Mal |
| Prueba de añadir un producto en específico | Verificar si es posible agregar un producto cualquiera. | Mal |
| Obtener precio del producto | Compruebe si es posible obtener el precio del producto correctamente. | Bien |
| Obtener cantidad del producto | Verifique si es posible obtener correctamente el precio del producto. | Bien |
| Obtener nombre del producto | Compruebe si es posible obtener correctamente el nombre del producto. | Bien |
| Añadir cantidad al producto | Verifique si es posible agregar cantidad a un producto. | Bien |
|  |  |  |