Name: Dikshyanta Uprety

Student ID: 20045364

Task: 1.2

This document contains the test strategies for the files TestGadget.java, TestDrown.java, and testPolymorphism.java.

**Test Strategy for TestGadget.java**

|  |  |  |
| --- | --- | --- |
| **Test case** | **Purpose** | **Results Expected** |
| Create a Gadget object with (quantityInStock (50), price (100), gadgetCode (“0001”)) as parameters | To test the parameterized constructor | A Gadget object is created with:  quantityInStock=50 price=100  gadgetCode= “0001” |
| Add 10 gadgets to the stock | To test the addGadget (int x) method | quantityInStock is increased by 10. (quantityInStock=60) |
| Add 0 gadgets to the stock | To test the addGadget (int x) method, when x<0 | An error message is displayed, informing at least a unit must be added to stock |
| Add 110 gadgets to the stock | To test the addGadget (int x) method, when x+ quantity>100 | An error message is displayed, informing that the stock cannot exceed 100 after addition |
| Sell 5 gadgets from the stock | To test the sellGadget (int x) method | quantityInStock is decreased by 5. (quantityInStock=55) |
| Sell 0 gadgets from the stock | To test the sellGadget (int x) method, when x<0 | An error message is displayed, informing at least a unit must be sold from the stock |
| Sell 105 gadgets from the stock | To test the sellGadget (int x) method, when x> quantity | An error message is displayed, informing that the number of gadgets sold cannot exceed the quantity |
| Set the price without VAT to 200 | To test the setPrice (double x) method | The price is set to 200. (price= 200) |

**Test Strategy for TestDrown.java**

|  |  |  |
| --- | --- | --- |
| **Test case** | **Purpose** | **Results Expected** |
| Create a Drown object with (quantityInStock (45), price (150), gadgetCode (“1001”), brand (“Dell”)) as parameters | To test the parameterized constructor | A Drown object is created with:  quantityInStock=45 price=150  gadgetCode= “0001”  brand= “Dell” |
| Add 8 drowns to the stock | To test the addGadget (int x) method | quantityInStock in stock is increased by 8. (quantityInStock=53) |
| Add 0 drowns to the stock | To test the addGadget (int x) method, when x<0 | An error message is displayed, informing at least a unit must be added to stock |
| Add 115 drowns to the stock | To test the addGadget (int x) method, when x+ quantity>100 | An error message is displayed, informing that the stock cannot exceed 100 after addition |
| Sell 5 drowns from the stock | To test the sellGadget (int x) method | quantityInStock in stock is decreased by 5. (quantityInStock=48) |
| Sell 0 drowns from the stock | To test the sellGadget (int x) method, when x<0 | An error message is displayed, informing at least a unit must be sold from the stock |
| Sell 125 drowns from the stock | To test the sellGadget (int x) method, when x> quantity | An error message is displayed, informing that the number of gadgets sold cannot exceed the quantity |
| Set the price without VAT of the drowns to 220 | To test the setPrice (double x) method | price is set to 220.  (price= 220) |
| Set the brand of the drowns to “Razor” | To test the setBrand (String x) method | The brand is set to Razor. (brand= “Razor”) |

**Test Strategy for TestPolymorphism.java**

TestPolymorphism class contains two methods, gadgetInstance (gadget x) and the main method. The test strategies for the two methods are presented using two tables below.

These inputs have been tried and results verified. However, due to it being user input that is stored only till the program runs, the evidence of these tests cannot be found in the program like TestGadget and TestDrown.

**For gadgetInstance (gadget x)**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Purpose** | **Expected Result** |
| Increase stock by stockToAdd | To test the addGadget (int x) method in different child classes of the Gadget class. | quantityInStock in increased by stockToAdd.  (quantityInStock+=stockToAdd) |
| Decrease stock by stockToSell | To test the sellGadget (int x) method in different child classes of the Gadget class. | quantityInStock is decreased by stockToSell.  (quantityInStock-=stockToSell) |
| Change price to priceToSet | To test the setPrice (double x) method in different child classes of the Gadget class. | Price of the gadget is set to priceToSet.  (price= priceToSet) |

**For the main method**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Purpose** | **Expected Result** |
| Input stockToAdd= 0 | Test if input is accepted when (!>0) | An error message is printed and the user is asked to re-enter values in range |
| Input stockToAdd=100 | Test if input is accepted when (!<=20) | An error message is printed and the user is asked to re-enter values in range |
| Input stockToAdd=15 | Test if input is accepted when (>0 && <=20) | The input is accepted and the program moves to its next task |
| Input stockToSell=-1 | Test if input is accepted when (!>0) | An error message is printed and the user is asked to re-enter values in range |
| Input stockToSell=250 | Test if input is accepted when (!<=20) | An error message is printed and the user is asked to re-enter values in range |
| Input stockToAdd=20 | Test if input is accepted when (>0 && <=20) | The input is accepted and the program moves to its next task |
| Input priceToSet=-10 | Test if input is accepted when (!>0) | An error message is printed and the user is asked to re-enter values in range |
| Input priceToSet=450 | Test if input is accepted when (!<200) | An error message is printed and the user is asked to re-enter values in range |
| Input priceToSet=100 | Test if input is accepted when (>0 && <=200) | The input is accepted and the program moves to its next task |