SOFTWARE QUALITY & METRICS
TEST 1

- ASHISH KUMAR
- 2K18/SE/041

Tasks:

- Exploit JUnit to test the attached program
- Specifications to test (write appropriate tests for the following conditions).

1. When created, the cart has 0 items.
2. When empty, the cart has 0 items.
3. When a new product is added, the number of items must be incremented.
4. When an item is removed, the number of items must be decreased.

Following programs are the input java files:

Product.java

```
package com.simplilearn.mavenproject;

public class Product {
        private String title;
        private double price;

        public Product(String t, double p) {
                this.title = t;
                this.price = p;
        }

        public String getTitle() {
                return title;
        }

        public double getPrice() {
                return price;
        }

        public boolean equals(Object o) {
                if (o instanceof Product) {
                        Product p = (Product) o;
                        return p.getTitle().equals(title);
        }
        return false;
    }
}
```

ShoppingCart.java

```
package com.simplilearn.mavenproject;

import java.util.*;

public class ShoppingCart {

    private ArrayList items;

    public ShoppingCart() {

        items = new ArrayList();

    }

    public double getBalance() {

        double balance = 0.00;

        for (Iterator i = items.iterator(); i.hasNext();) {

            Product item = (Product) i.next();

            balance += item.getPrice();

        }

        return balance;

    }

    public void addItem(Product item) {

        items.add(item);

    }

    public void removeItem(Product item) throws ProductNotFoundException {

        if (!items.remove(item)) {

            throw new ProductNotFoundException();
```

```
		}
	}
	public int getItemCount() {
		return items.size();
	}
	public void empty() {
		items.clear();
	}
}
```

ProductNotFoundException.java

```
package com.simplilearn.mavenproject;
public class ProductNotFoundException extends Exception {
	public ProductNotFoundException() {
		super();
	}
}
```

THEORY: JUnit is a popular unit-testing framework in the Java ecosystem. JUnit is a Java library to help you perform unit testing. Unit testing is the process of examining a small "unit" of software (usually a single class) to verify that it meets its expectations or specification. A unit test generally consists of various testing methods that each interacts with the class under test in some specific way to make sure it works as expected.

My Testing Code:

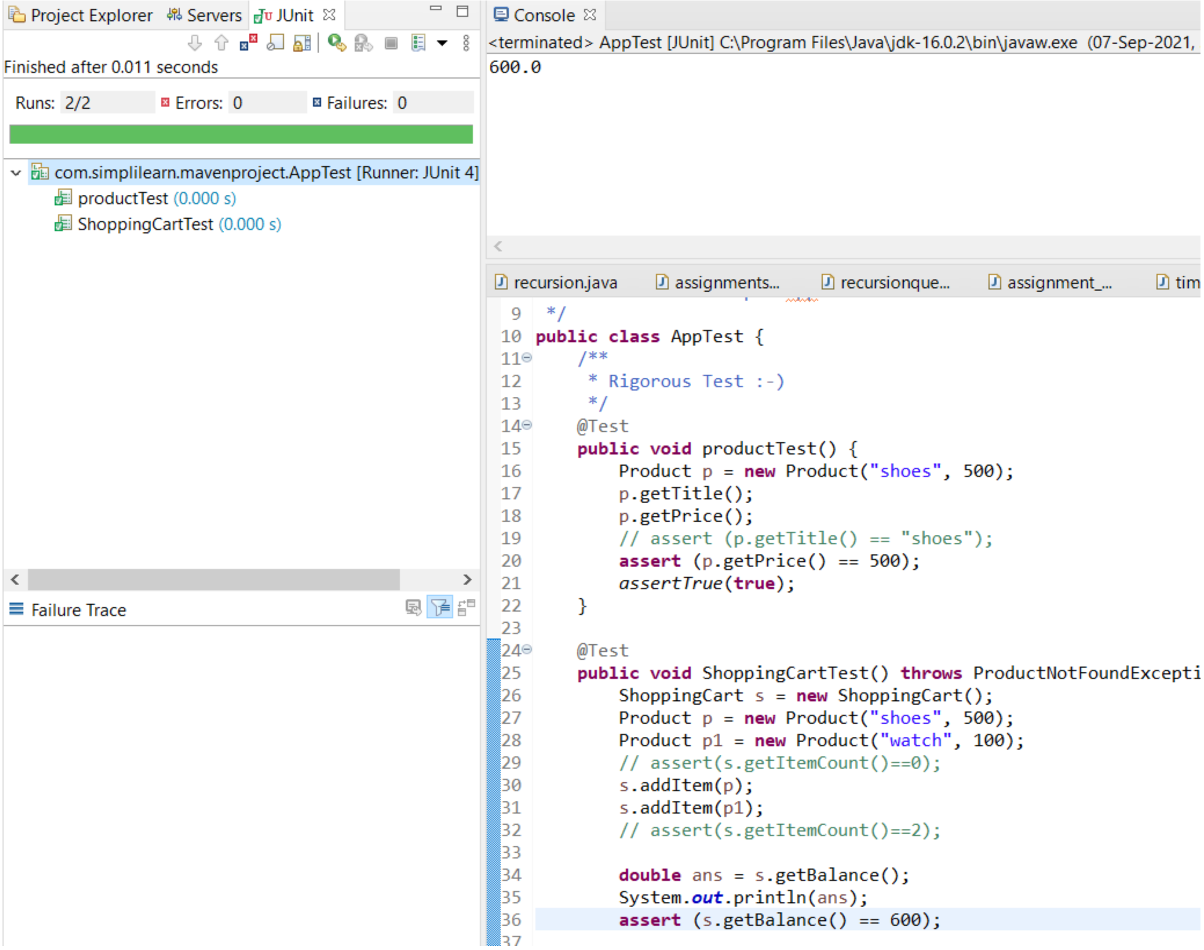
I have written testing code in a java file named as "**AppTest.java**" and below is the code of it:

```
package com.simplilearn.mavenproject;

import static org.junit.Assert.assertTrue;

import org.junit.Test;

/**

 * Unit test for simple App.

 */

public class AppTest {

    /**

     * Rigorous Test :-)

     */

    @Test

    public void productTest() {   // for testing Product.java file

        Product p = new Product("shoes", 500);    //created a object p

        p.getTitle();

        p.getPrice();

      // assert (p.getTitle() = = "shoes");  // to check whether title of object is shoes or not

        assert (p.getPrice() = = 500);    // to check whether the price of object is 500 or not

        assertTrue(true);

    }
```

```
// for testing ShoppingCart.java file
@Test
        public void ShoppingCartTest() throws ProductNotFoundException {
                ShoppingCart s = new ShoppingCart();     //created a object s of ShoppingCart
                Product p = new Product("shoes", 500);    //created a object p of class Product
                Product p1 = new Product("watch", 100);  //created a object p1 of class Product
                // assert(s.getItemCount()= =0);

                s.addItem(p);                                    //adding items to ShoppingCart
                s.addItem(p1);
                // assert(s.getItemCount()= =2);    //checking whether the count of cart is 2 or not

        // double ans=s.getBalance();    //Getting the value of total balance of added items in cart
                // System.out.println(ans);
                // assert(s.getBalance()= = 600);

                // s.removeItem(p1);                    //removing p1 item from cart
                // assert(s.getItemCount()= = 1);   //checking whether the count of cart is 1 or not

                s.empty();                              // emptying the cart means to remove all added items
                assert (s.getItemCount() = = 0); //checking whether the cart is empty or not
                assertTrue(true);
        }
}
```

OUTPUT:

There is **no error and failure**. Also Total Balance is displayed in console section. ProductTest() and ShoppingCartTest() is successfully executed.



Finished after 0.011 seconds

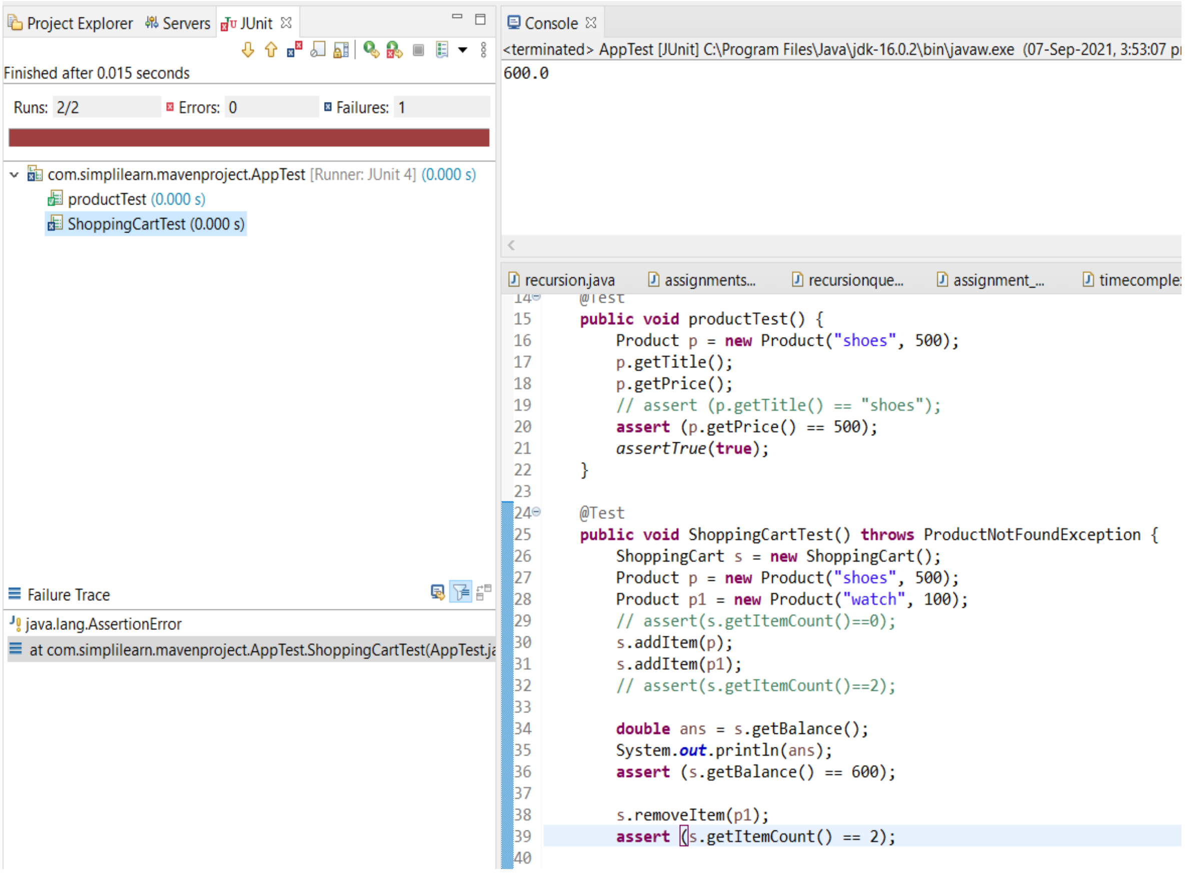
Runs: 2/2 Errors: 0 Failures: 0

com.simplilearn.mavenproject.AppTest [Runner: JUnit 4]
productTest (0.000 s)
ShoppingCartTest (0.000 s)

```
<terminated> AppTest [JUnit] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe  (07-Sep-2021,
600.0
```

```
 */
public class AppTest {
    /**
     * Rigorous Test :-)
     */
    @Test
    public void productTest() {
        Product p = new Product("shoes", 500);
        p.getTitle();
        p.getPrice();
        // assert (p.getTitle() == "shoes");
        assert (p.getPrice() == 500);
        assertTrue(true);
    }

    @Test
    public void ShoppingCartTest() throws ProductNotFoundExcepti
        ShoppingCart s = new ShoppingCart();
        Product p = new Product("shoes", 500);
        Product p1 = new Product("watch", 100);
        // assert(s.getItemCount()==0);
        s.addItem(p);
        s.addItem(p1);
        // assert(s.getItemCount()==2);

        double ans = s.getBalance();
        System.out.println(ans);
        assert (s.getBalance() == 600);
```

Now there is **no error** because syntax is correct. But there is **1 failure** and it occurs in Line No. 39 as I have already remove p1 from cart. Now count of cart becomes 1, and I am checking for count of 2. That's why it is giving failure.



Finished after 0.015 seconds

Runs: 2/2 Errors: 0 Failures: 1

com.simplilearn.mavenproject.AppTest [Runner: JUnit 4] (0.000 s)
productTest (0.000 s)
ShoppingCartTest (0.000 s)

java.lang.AssertionError
at com.simplilearn.mavenproject.AppTest.ShoppingCartTest(AppTest.ja

```
<terminated> AppTest [JUnit] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (07-Sep-2021, 3:53:07 p
600.0
```

```
14      @Test
15      public void productTest() {
16          Product p = new Product("shoes", 500);
17          p.getTitle();
18          p.getPrice();
19          // assert (p.getTitle() == "shoes");
20          assert (p.getPrice() == 500);
21          assertTrue(true);
22      }
23
24      @Test
25      public void ShoppingCartTest() throws ProductNotFoundException {
26          ShoppingCart s = new ShoppingCart();
27          Product p = new Product("shoes", 500);
28          Product p1 = new Product("watch", 100);
29          // assert(s.getItemCount()==0);
30          s.addItem(p);
31          s.addItem(p1);
32          // assert(s.getItemCount()==2);
33
34          double ans = s.getBalance();
35          System.out.println(ans);
36          assert (s.getBalance() == 600);
37
38          s.removeItem(p1);
39          assert (s.getItemCount() == 2);
40
```

Rest of the explanation is given in the video (attached in zip folder).