# CMPE 313 - Homework #3

Deniz Melih Öz & Muhammed Abdullah Yusuf

## Introduction

JUnit, which is the standard testing tool for development in Java, is a part of the XUnit tools that are used for Test-Driven Development in progrogramming languages. When a code is given, ad hoc testing can be carried out which involves tests which occur to the programmer in the moment or, a test suite can be built which is a method of thorough testing which can be run at any time. The disadvantages might include extra programming and a feeling that it is unnecessary but a good testing framework is extremely effective and reduces debugging time more than the amount of time spent in building the test suite.

## Implementation

The main class which we have selected to implement JUnit on is “BrowserPage” represented by MainWin which implements and makes use of other classes such as Manager, ShoppingCart, Stock and User. Basically, our main class is a page which displays multiple books, their availability, and the option to purchase them via adding them to the user’s shopping cart. The “Driver” class is also present which makes use of various methods such as addStock(), which is used to add books into stock, addUser(), addManager() etc.

Below are images of the main class and its various subsequent classes.

### Driver Class

```
1
2  import java.awt.EventQueue;
3
4
5  public class Driver {
6      //Count goes up each time an object is created
7      static int UserID=0;
8      static int ManagerID=0;
9      static int StockID=0;
10     static int CartID=0;
11
12     //Stores some objects
13 static User[] Users= new User[99];
14 static Manager[] Managers= new Manager[99];
15 static Stock[] Inventory= new Stock[99];
16 static ShoppingCart[] Carts= new ShoppingCart[99];
17     @SuppressWarnings("unused")
18     public static void main(String[] args) {
19     //Manually Populating Users, Managers, Stock, and a few Carts
20         addUser("Adam","passAdam123"); //1
21         addUser("Bob","passBob123"); //2
22         addUser("Charlie","passCharlie123"); //3
23         addUser("David","passDavid123"); //4
24         addUser("Eddie","passEddie123"); //5
25         //
26         addManager("Rep. Adam","passRepAdam123"); //1
27         addManager("Rep. Bob","passRepBob123"); //2
28         //
29         addStock("Da Vinci Code for Dummies","Bad",11.11,11); //1
30         addStock("Kite Runner","Good",32.21,21); //2
31         addStock("Fifty Shades","Bad",69.0,53); //3
32         addStock("Harry Potter 1","Bad",42.0,42); //4
33         addStock("Mockingbird","Good",31.13,47); //5
34         addStock("Hannibal","Good",89.25,90); //6
35         addStock("How to Steal Books","Bad",0.0,0); //7
36         addStock("Hunger Games","Good",18.85,24); //8
37         //
38         addCart(Users[2]); //1
39         addCart(Users[4]); //2
40         addCart(Users[5]); //3
41     // UI Stuff beginnings
42     EventQueue.invokeLater(new Runnable() {
43     public void run() {
44         try {
45             MainWin main= new MainWin(Users,Managers,Inventory,Carts);
46             } catch (Exception e) {
47                 e.printStackTrace();
48                 }}});
49
50     }
```

```
53     static void addUser(String a,String b) {
54         User temp= new User(a,b,UserID);
55         Users[UserID]=temp;
56         UserID++;
57     }
58     static void addManager(String a,String b) {
59         Manager temp= new Manager(a,b,ManagerID);
60         Managers[ManagerID]=temp;
61         ManagerID++;
62     }
63     static void addStock(String a,String b, Double c,int d) {
64         Stock temp= new Stock(a,b,c,d,StockID);
65         Inventory[StockID]=temp;
66         StockID++;
67     }
68     static void addCart(User a) {
69         ShoppingCart temp= new ShoppingCart(a,CartID);
70         Carts[CartID]=temp;
71         CartID++;
72     }
73
```

**A Snippet of the MainWIn Class (The Entire Code is too long so it has not been uploaded)**

```
1
2  import java.awt.CardLayout;
23
24 public class MainWin {
25
26
27
28
29
30
31
32
33 //  Double sub=0.0,prod=0.0,kval=0.0;
34     //Arrays from Driver
35     User[] UsersFinger;
36     Manager[] ManagersFinger;
37     Stock[] InventoryFinger;
38     ShoppingCart[] CartsFinger;
39     //
40     JFrame fr = new JFrame();
41     JPanel cards= new JPanel(new CardLayout()); //CardLayoutContainer
42     JPanel startpan= new JPanel();
43     JPanel pan1=new JPanel();
44     JPanel pan2=new JPanel();
45     JPanel pan3=new JPanel();
46     //Test stuff below
47     //
48     //Card Layout Set
49     CardLayout cl = (CardLayout) (cards.getLayout());
50
51     /// Test Buttons on start panel (Comment out later)
52     JButton but1= new JButton("User View");
53     JButton but2= new JButton("Manager View");
54     JButton but3= new JButton("Shopping Cart");
55     //
56
57
58     //Constructor
59     public MainWin(User[] a,Manager[] b,Stock[] c,ShoppingCart[] d) {
60
61         //Get "Pointers" of drivers arrays
62         UsersFinger=a;
63         ManagersFinger=b;
64         InventoryFinger=c;
65         CartsFinger=d;
66         //Frame Stuff
67         fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
68         fr.setTitle("TEDU Books");
69         fr.setSize(950,650);
70         fr.setVisible(true);
71         fr.setResizable(false);
72         fr.getContentPane().add(cards);
73         //Setup inner Panels
74         setupstartpan();
```

This class requires the arrays present in the Driver class to be fed into it. It creates a Swing Window and several panels for customers to interact with. These panels are not separate classes and do not extend JPanel, therefore every “page” of the program is processed by a single JUnit Test Class, **TestMainWin**.

## Manager Class

```java

public class Manager{
        int ManagerID;
        String Username="";
        String Pass="";
        //Constructors
        Manager(String a,String b,int c){this.Pass=a;this.Username=b;this.ManagerID=c;}
        Manager(){this.Username="";}
        Manager(String a){this.Username=a;}
        //ToString
        public String toString() {
            String temp= "User: "+this.Username+" Pass: "+this.Pass+" ID:"+this.ManagerID;
            return temp;}
        //Setter Methods
        void setPass(String a) {this.Pass=a;}
        void setUname(String a){this.Username=a;}
        //Getter Methods
        String getUname() {return this.Username;}

String getPass() {return this.Pass;}

int getID() {return this.ManagerID;}
}

```

## ShoppingCart Class

```java
import java.util.Stack;

public class ShoppingCart {
int CartID;
User Owner;
//Stock[] CartItems=new Stock[99];
Stack<Stock> CartItems= new Stack<Stock>();
int CartLastIndex=0;
ShoppingCart(User a,int b){this.Owner=a;this.CartID=b;}
//ToString
public String toString() {
    String temp= "Cart Owner: "+this.Owner.getUname()+" Owner ID: "+this.Owner.getID()+"Items in Cart: "+(this.CartLastIndex+1)+" CartID:"+this.CartID;
    return temp;}



//Add/Remove Items to from Cart
void AddItem(Stock a){this.CartItems.push(a);this.CartLastIndex++;}

void RemoveItem(Stock a) {this.CartItems.remove(a);}

void EmptyCart() {while(!this.CartItems.empty()) {
    this.CartItems.pop();
}
this.CartLastIndex=0;
}



//Auto Getter/Setters
public int getCartID() {
    return CartID;
}
public void setCartID(int cartID) {
    CartID = cartID;
}
public User getOwner() {
    return Owner;
}
public void setOwner(User owner) {
    Owner = owner;
}
public Stack<Stock> getCartItems() {
    return CartItems;
}
public void setCartItems(Stack<Stock> cartItems) {
    CartItems = cartItems;
}

}

```

## Stock Class

```java

public class Stock {
int StockID;
String BookName;
Double Price;
int StockCount;
String Category;
//Constructors
Stock(String a,String b,Double c,int d,int e){this.BookName=a;this.Category=b;this.Price=c;this.StockCount=d;this.StockID=e;}

//ToString
public String toString() {
    String temp= "BookName: "+this.BookName+" Price: "+this.Price+"Amount: "+this.StockCount+" ID:"+this.StockID;
    return temp;}

//Getter Setter AutoComplete
public int getStockID() {
    return StockID;
}
public void setStockID(int stockID) {
    StockID = stockID;
}
public String getBookName() {
    return BookName;
}
public void setBookName(String bookName) {
    BookName = bookName;
}
public Double getPrice() {
    return Price;
}
public void setPrice(Double price) {
    this.Price = price;
}
public int getStockCount() {
    return StockCount;
}
public void setStockCount(int stockCount) {
    StockCount = stockCount;
}
public void StockUp(){this.StockCount++;}
public void StockDown() {this.StockCount--;}

}

```

## TestMainJava Class

```java
import static org.junit.jupiter.api.Assertions.*;

class TestMainWin {

    @Test
    void testMainWin() {
        MainWin MainWin = new MainWin(null, null, null, null);
        fail("Not yet implemented");
    }

    //Browse Function Assertions
    //Tests that the Array being displayed by JTable has all the books from the selected category, by doing a fresh search.
    void testTable() {
        try {
            MainWin test= new MainWin(Driver.Users, Driver.Managers, Driver.Inventory,Driver.Carts);
            } catch (Exception e) {
                e.printStackTrace();
                }

// 1. Check the Checkbox selected Item
// 2. Create a new array of all books from Stock with selected Category (fresh check)
// 3. Pull array of all books displayed on JTable
// 4. assertArrayEquals(String "Displayed Books are of the right category?", Check[],JTable[]) to check the two arrays against each other.
//  This will tell us if the list of books being displayed is current.

    }

    //Tests that the category of the Selected Item on Checkbox, is the same as the category of the first book shown.
    void testCheckbox() {
        try {
        MainWin test= new MainWin(Driver.Users, Driver.Managers, Driver.Inventory,Driver.Carts);
        } catch (Exception e) {
            e.printStackTrace();
            }

    }

    void test3() {}

    void test4() {}

    void test5() {}


    // Tests whether or not the Test window is able to be opened.
    void testWindowOpened() {
        try {
            MainWin test= new MainWin(Driver.Users, Driver.Managers, Driver.Inventory,Driver.Carts);
            fail("My program did not throw an exception!");
            } catch (Exception e) {
                e.printStackTrace();
                }
    }

```

The above class is the testing class which implements JUnit. As you can see that the assertions are present here. We are testing categories regarding the books present in our Library. It creates a new array of all the books present in our stock with the selected category. Then, it loads all of our books by pulling the array which displays our JTable. The method assertArrayEquals() makes the selected arrays equal. This will return a value which will indicate whether the list of books being displayed is correct.