Lab 07 - Testing (part 2)

In this lab you will write some tests for some classes in Java.

We have class Product with the following codes:

```java
// Product.java
public class Product {
    private String name;
    private double price;
    private double discount; // in percentage between 0 and 100
    public String getName() {
        return name;
    }
    public void setName(String name) {
        if(!name.isBlank()) {
            this.name = name;
        }
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        if(price>0) {
            this.price = price;
        }
    }
    public double getDiscount() {
        return discount;
    }
    public void setDiscount(double discount) {
        if(discount>=0 && discount<=100) {
            this.discount = discount;
        }
    }
}
```

And class ProductManager with the follwing codes:

```java
// ProductManager.java
public class ProductManager {
    private Product[] products = new Product[100];
    private int count = 0;
    public void addProduct(String name, double price){
        addProduct(name, price, 0);
    }
    public void addProduct(String name, double price, double discount){
        if(findProduct(name) >= 0) {
            System.out.println("Can't add, product already exists.");
            return;
        }
        Product prod = new Product();
        prod.setName(name.trim());
        prod.setPrice(price);
        prod.setDiscount(discount);
        products[count++] = prod;
    }

    public int findProduct(String name) {
        name = name.trim();
        for (int i=0;i<count;i++) {
            if(products[i].getName().equalsIgnoreCase(name)) {
                return i;
            }
        }
        return -1;
    }

    public Product removeProduct(String name) {
        int foundIndex = findProduct(name);
        if(foundIndex == -1) {
            System.out.println("Product not found!");
            return null;
        }
        Product prod = products[foundIndex];
        for(int i=foundIndex;i<count;i++) {
            products[i] = products[i+1];
        }
        count--;
        return prod;
    }

    public boolean updateProduct(String oldName, String newName, double newPrice, double newDiscount) {
        int foundIndex = findProduct(oldName);
        if(foundIndex == -1) {
            System.out.println("Product not found!");
            return false;
        }
        products[foundIndex].setName(newName.trim());
        products[foundIndex].setPrice(newPrice);
        products[foundIndex].setDiscount(newDiscount);
        return true;
    }

    public int getCount() {
        return count;
    }
}
```

Task 1 - Product test class

Write ProductTest class to test setter methods of Product class. Tests should includes:

- Set name correctly (name is not blank)
- Set name incorrectly (name is blank)
- Set price correctly (price is greater than 0)
- Set price incorrectly (price is less than or equals to 0)
- Set discount correctly (discount is between 0 and 100)
- Set discount incorrectly (discount less than 0 or greater than 100)

Task 2 - Add product test

Write AddProductTest class to test ProductManager.addProduct method:

- Add product correctly (product name not exists)
- Add product incorrectly (product name duplicated)
- Check for count is increased by 1 or not

Task 3 - Remove product test

Write RemoveProductTest class to test ProductManager.removeProduct method:

- Remove product correctly (product name exists)
- Remove product incorrectly (product name not exists)
- Check for count is decreased by 1 or not

Task 4 - Update product test

Write UpdateProductTest class to test ProductManager.updateProduct method:

- Update product correctly (product name exists, new name is not blank)
- Update product incorrectly (product name not exists)
- Update product incorrectly (product name exists but new name is blank)
- Update product incorrectly (product name exists but new price is less than 0)
- Update product incorrectly (product name exists but new discount is less than 0 or greater than 100)
- Check for count is stay the same as before the update

Task 5 - Menu and Product management

Write java application to display a menu for user to manage the products.