Project 2

Let's extend the project to incorporate the new requirements as outlined in Sections 5.1 and the additional features:

Step 1: Open the Inventory Program

Assume the previous program is saved and now we will modify it according to the new instructions.

Step 2: Modify `ProductTester` Class

We'll start by modifying the `ProductTester` class to include user input using a `Scanner`.

```java
import java.util.Scanner;

public class ProductTester {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        // Variables to store user input
        int tempNumber;
        String tempName;
        int tempQty;
        double tempPrice;

        // Getting user input for first product
        System.out.println("Enter the details for Product 1:");
        System.out.print("Item Number: ");
        tempNumber = in.nextInt();
        in.nextLine();  // Consume newline left-over
        System.out.print("Name: ");
        tempName = in.nextLine();
```

```
System.out.print("Quantity: ");
tempQty = in.nextInt();
System.out.print("Price: ");
tempPrice = in.nextDouble();

// Create product p1 using parameterized constructor
Product p1 = new Product(tempNumber, tempName, tempQty, tempPrice);

// Getting user input for second product
System.out.println("\nEnter the details for Product 2:");
System.out.print("Item Number: ");
tempNumber = in.nextInt();
in.nextLine();  // Consume newline left-over
System.out.print("Name: ");
tempName = in.nextLine();
System.out.print("Quantity: ");
tempQty = in.nextInt();
System.out.print("Price: ");
tempPrice = in.nextDouble();

// Create product p2 using parameterized constructor
Product p2 = new Product(tempNumber, tempName, tempQty, tempPrice);

// Closing the Scanner
in.close();

// Display details of each product
System.out.println("\nProduct 1 details:");
System.out.println(p1.toString());

System.out.println("\nProduct 2 details:");
```

```
        System.out.println(p2.toString());
    }
}
```

Step 3: Modify `Product` Class

Now, modify the `Product` class to include a boolean field `active`, its getter and setter, a method to calculate the inventory value, and update the `toString()` method.

```java
public class Product {
    // Instance field declarations
    private int itemNumber;
    private String name;
    private int unitsInStock;
    private double price;
    private boolean active;

    // Default constructor
    public Product() {
        this.itemNumber = 0;
        this.name = "";
        this.unitsInStock = 0;
        this.price = 0.0;
        this.active = true;
    }

    // Parameterized constructor
    public Product(int number, String name, int qty, double price) {
        this.itemNumber = number;
        this.name = name;
```

```
    this.unitsInStock = qty;
    this.price = price;
    this.active = true;
}

// Getter and Setter methods
public int getItemNumber() {
    return itemNumber;
}

public void setItemNumber(int itemNumber) {
    this.itemNumber = itemNumber;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getUnitsInStock() {
    return unitsInStock;
}

public void setUnitsInStock(int unitsInStock) {
    this.unitsInStock = unitsInStock;
}

public double getPrice() {
```

```
    return price;
}

public void setPrice(double price) {
    this.price = price;
}

public boolean isActive() {
    return active;
}

public void setActive(boolean active) {
    this.active = active;
}

// Method to calculate inventory value
public double getInventoryValue() {
    return price * unitsInStock;
}

// Override toString() method
@Override
public String toString() {
    String status = active ? "Active" : "Discontinued";
    return "Item Number: " + itemNumber +
           "\nName: " + name +
           "\nQuantity in stock: " + unitsInStock +
           "\nPrice: " + price +
           "\nStock Value: " + getInventoryValue() +
           "\nProduct Status: " + status;
}
```

}
```

### Step 4: Modify `ProductTester` to Use `setActive` and Display Products
Add code to mark `p1` or `p2` as discontinued and display the updated details.

```java
import java.util.Scanner;

public class ProductTester {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        // Variables to store user input
        int tempNumber;
        String tempName;
        int tempQty;
        double tempPrice;

        // Getting user input for first product
        System.out.println("Enter the details for Product 1:");
        System.out.print("Item Number: ");
        tempNumber = in.nextInt();
        in.nextLine();  // Consume newline left-over
        System.out.print("Name: ");
        tempName = in.nextLine();
        System.out.print("Quantity: ");
        tempQty = in.nextInt();
        System.out.print("Price: ");
        tempPrice = in.nextDouble();

```
    // Create product p1 using parameterized constructor
    Product p1 = new Product(tempNumber, tempName, tempQty, tempPrice);

    // Getting user input for second product
    System.out.println("\nEnter the details for Product 2:");
    System.out.print("Item Number: ");
    tempNumber = in.nextInt();
    in.nextLine();  // Consume newline left-over
    System.out.print("Name: ");
    tempName = in.nextLine();
    System.out.print("Quantity: ");
    tempQty = in.nextInt();
    System.out.print("Price: ");
    tempPrice = in.nextDouble();

    // Create product p2 using parameterized constructor
    Product p2 = new Product(tempNumber, tempName, tempQty, tempPrice);

    // Mark p2 as discontinued
    p2.setActive(false);

    // Closing the Scanner
    in.close();

    // Display details of each product
    System.out.println("\nProduct 1 details:");
    System.out.println(p1.toString());

    System.out.println("\nProduct 2 details:");
    System.out.println(p2.toString());
}
```

}
```

### Step 9: Save Your Project

Ensure all changes are saved in your IDE or text editor.

This completes the additional features specified for Sections 5.1. Run and test the program to ensure it works as expected.
```