Project 4.5

Sure, let's walk through the steps to extend the `Product` class to create `DVD` and `CD` subclasses and then update the `ProductTester` class to handle these new types of products.

Step 1: Create `DVD` Subclass

1. **Create `DVD` class**:
  - Add additional instance fields: `length`, `ageRating`, and `filmStudio`.
  - Create a constructor that uses `super()` to call the `Product` constructor.
  - Create getters and setters for the new fields.
  - Override the `getInventoryValue` method to add a 5% restocking fee.
  - Override the `toString` method to include the new fields.

`DVD` Class

```java
public class DVD extends Product {
    private int length;
    private int ageRating;
    private String filmStudio;

    public DVD(int itemNumber, String name, int quantity, double price, int length, int ageRating, String filmStudio) {
        super(itemNumber, name, quantity, price);
        this.length = length;
        this.ageRating = ageRating;
        this.filmStudio = filmStudio;
    }

    public int getLength() {
        return length;
```

```
}

public void setLength(int length) {
    this.length = length;
}

public int getAgeRating() {
    return ageRating;
}

public void setAgeRating(int ageRating) {
    this.ageRating = ageRating;
}

public String getFilmStudio() {
    return filmStudio;
}

public void setFilmStudio(String filmStudio) {
    this.filmStudio = filmStudio;
}

@Override
public double getInventoryValue() {
    return super.getInventoryValue() * 1.05; // Adding 5% restocking fee
}

@Override
public String toString() {
    return super.toString() +
            "\nMovie Length: " + getLength() +
                "\nAge Rating: " + getAgeRating() +
                "\nFilm Studio: " + getFilmStudio();
    }
}
```

Step 2: Create `CD` Subclass

1. **Create `CD` class**:
   - Add additional instance fields: `artist`, `numberOfSongs`, and `label`.
   - Create a constructor that uses `super()` to call the `Product` constructor.
   - Create getters and setters for the new fields.
   - Override the `toString` method to include the new fields.

`CD` Class

```java
public class CD extends Product {
    private String artist;
    private int numberOfSongs;
    private String label;

    public CD(int itemNumber, String name, int quantity, double price, String artist, int numberOfSongs, String label) {
        super(itemNumber, name, quantity, price);
        this.artist = artist;
        this.numberOfSongs = numberOfSongs;
        this.label = label;
    }

    public String getArtist() {
        return artist;
    }

    public void setArtist(String artist) {
        this.artist = artist;
    }

    public int getNumberOfSongs() {
        return numberOfSongs;
    }

    public void setNumberOfSongs(int numberOfSongs) {
        this.numberOfSongs = numberOfSongs;
    }

    public String getLabel() {
        return label;
    }

    public void setLabel(String label) {
        this.label = label;
    }

    @Override
    public String toString() {
        return super.toString() +
                "\nArtist: " + getArtist() +
                "\nSongs on Album: " + getNumberOfSongs() +
                "\nRecord Label: " + getLabel();
    }
}
```

~~~

Step 3: Update `ProductTester` Class

1. **Update `addToInventory` method to handle `CD` and `DVD` objects**.

2. **Create `addCDToInventory` method** to add CDs.

3. **Create `addDVDToInventory` method** to add DVDs.

Modified `ProductTester` Class

```java
import java.util.Scanner;

public class ProductTester {
    public static void main(String[] args) {
        // Create a Scanner object for keyboard input
        Scanner in = new Scanner(System.in);

        int maxSize = getNumProducts(in);

        if (maxSize == 0) {
            System.out.println("No products required!");
        } else {
            Product[] products = new Product[maxSize];
            int productType;

            for (int i = 0; i < products.length; i++) {
                productType = getProductType(in);
                if (productType == 1) {
                    addCDToInventory(products, i, in);
                } else {
~~~

```
                addDVDToInventory(products, i, in);
            }
        }

        int menuChoice;
        do {
            menuChoice = getMenuOption(in);
            executeMenuChoice(menuChoice, products, in);
        } while (menuChoice != 0);
    }

    // Close the Scanner
    in.close();
}

// Method to get the type of product
public static int getProductType(Scanner in) {
    int productType = -1;
    do {
        try {
            System.out.println("Enter product type (1 for CD, 2 for DVD): ");
            productType = in.nextInt();
            if (productType < 1 || productType > 2) {
                System.out.println("Invalid product type. Please enter 1 or 2.");
            }
        } catch (Exception e) {
            System.out.println("Incorrect data type entered! Please enter a valid integer.");
            in.nextLine(); // Clear the input buffer
        }
    } while (productType < 1 || productType > 2);
    return productType;
}

// Method to add CDs to the inventory
public static void addCDToInventory(Product[] products, int index, Scanner in) {
    int tempNumber;
    String tempName;
    int tempQty;
    double tempPrice;
    String tempArtist;
    int tempNumSongs;
    String tempLabel;

    in.nextLine(); // Clear the input buffer
    System.out.println("Enter the details for CD " + (index + 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();
    in.nextLine(); // Consume newline left-over
    System.out.print("Artist: ");
    tempArtist = in.nextLine();
    System.out.print("Number of Songs: ");
    tempNumSongs = in.nextInt();
    in.nextLine(); // Consume newline left-over
    System.out.print("Label: ");
```

```
        tempLabel = in.nextLine();


        products[index] = new CD(tempNumber, tempName, tempQty, tempPrice, tempArtist,
tempNumSongs, tempLabel);
    }


    // Method to add DVDs to the inventory
    public static void addDVDToInventory(Product[] products, int index, Scanner in) {
        int tempNumber;
        String tempName;
        int tempQty;
        double tempPrice;
        int tempLength;
        int tempAgeRating;
        String tempFilmStudio;


        in.nextLine(); // Clear the input buffer
        System.out.println("Enter the details for DVD " + (index + 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();
        System.out.print("Length (minutes): ");
        tempLength = in.nextInt();
        System.out.print("Age Rating: ");
        tempAgeRating = in.nextInt();
        in.nextLine(); // Consume newline left-over
        System.out.print("Film Studio: ");
        tempFilmStudio = in.nextLine();

        products[index] = new DVD(tempNumber, tempName, tempQty, tempPrice, tempLength, tempAgeRating, tempFilmStudio);
    }

    // The rest of the methods from the previous implementation...
}
```

Step 4: Run and Test Your Code

Ensure to run your program to check if it functions correctly with the new `CD` and `DVD` subclasses. This completes the necessary updates for Section 7 Part 2 of the inventory project. If you encounter any issues or need further assistance, feel free to ask!