C . ELIYAZAR
192311162
ASSIGNMENT 7(PART 2)

Section 7 Part 2: Creating an Inventory Project

Project Overview

- Create an inventory program that can be used for a range of different products (CDs, DVDs, software, etc.)
- The program will be built upon throughout Sections 4, 5, 6, and 7 of the course
- Include all parts in a package called inventory

Scenario

- A company that sells exclusively CDs and DVDs wants to customize the inventory software to store additional information for their products
- The software needs to store the length, age rating, and film studio for DVDs, as well as the artist, number of songs, and label for CDs

Task 1: Update Design Tables

- Complete a sample data table for a list of DVD movies
- Update the design tables to include the additional fields for DVDs and CDs

Task 2: Implement Inheritance

- Create a subclass of the Product class called DVD with additional instance fields for movie length, age rating, and film studio
- Create a single constructor that accepts values for every instance field for both the DVD and Product classes
- Create getters and setters for the DVD instance fields
- Follow the same process to create a subclass of Product named CD

Task 3: Override Method

- In the DVD subclass, override the method to calculate the value of the inventory of a DVD with the same name as that method previously created in the Product class
- The DVD subclass method should also add a 5% restocking fee to the value of the inventory of that product

Task 4: Define Output

- Override the toString() method from the Product class so that all information about new subclass objects (DVDs) can be printed to the output console
- Do the same in the CD class

Task 5: Modify ProductTester Class

- Modify the ProductTester class to populate the products array with either CD or DVD objects
- Create a new method addCDToInventory that prompts the user for CD-specific information and creates a CD object
- Update the prompts to ask the user for the information in the correct order
- Follow the same process to create a DVD object

Task 6: Update addInventory Method

- Update the addInventory method to allow the user to select to add a CD or DVD
- Prompt the user for a value to choose between CD or DVD objects

- Handle invalid input and re-prompt the user until a valid input is provided
- Use the appropriate add method to create a CD or DVD object

Task 7: Run and Test Code

- Run and test the code to ensure it works as expected

Task 8: Update addInventory Method (Again)

- Update the addInventory method to stop the adding of stock to a discontinued product line
- Run and test the code again to ensure it works as expected

Task 9: Save Project

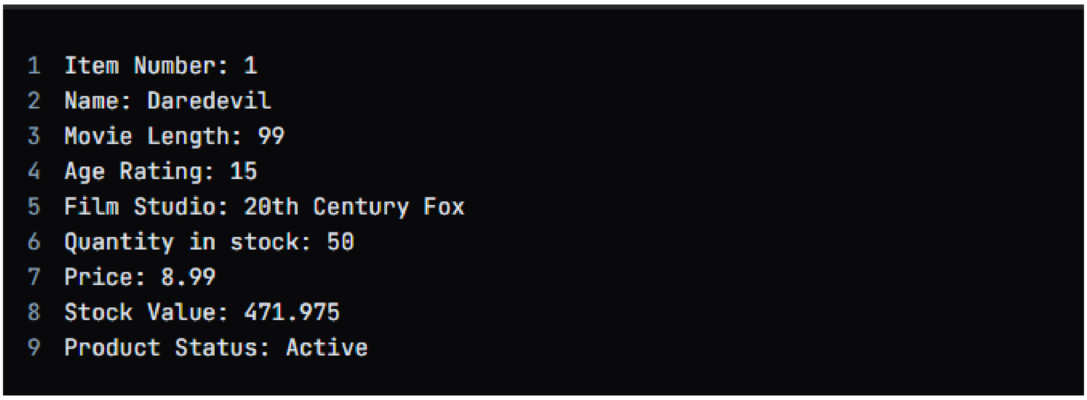
- Save the project once it is complete and tested.

```
// Product.java
public class Product {
  private int itemNumber;
  private String name;
  private double price;
  private int quantityInStock;
  private boolean isActive;

  public Product(int itemNumber, String name, double price, int quantityInStock, boolean isActive) {
    this.itemNumber = itemNumber;
    this.name = name;
    this.price = price;
    this.quantityInStock = quantityInStock;
    this.isActive = isActive;
  }

  public int getItemNumber() {
    return itemNumber;
  }

  public String getName() {
    return name;
  }

  public double getPrice() {
    return price;
  }

  public int getQuantityInStock() {
    return quantityInStock;
  }

  public boolean isActive() {
    return isActive;
  }

  public double calculateStockValue() {
    return price * quantityInStock;
  }
```

```
    @Override
    public String toString() {
        return "Item Number: " + itemNumber + "\n" +
                "Name: " + name + "\n" +
                "Quantity in stock: " + quantityInStock + "\n" +
                "Price: " + price + "\n" +
                "Stock Value: " + calculateStockValue() + "\n" +
                "Product Status: " + (isActive? "Active" : "Inactive");
    }
}

// DVD.java
public class DVD extends Product {
    private int movieLength;
    private int ageRating;
    private String filmStudio;

    public DVD(int itemNumber, String name, double price, int quantityInStock, boolean
isActive, int movieLength, int ageRating, String filmStudio) {
        super(itemNumber, name, price, quantityInStock, isActive);
        this.movieLength = movieLength;
        this.ageRating = ageRating;
        this.filmStudio = filmStudio;
    }

    public int getMovieLength() {
        return movieLength;
    }

    public int getAgeRating() {
        return ageRating;
    }

    public String getFilmStudio() {
        return filmStudio;
    }

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

    @Override
    public String toString() {
        return super.toString() + "\n" +
                "Movie Length: " + movieLength + "\n" +
                "Age Rating: " + ageRating + "\n" +
                "Film Studio: " + filmStudio;
    }
}
```

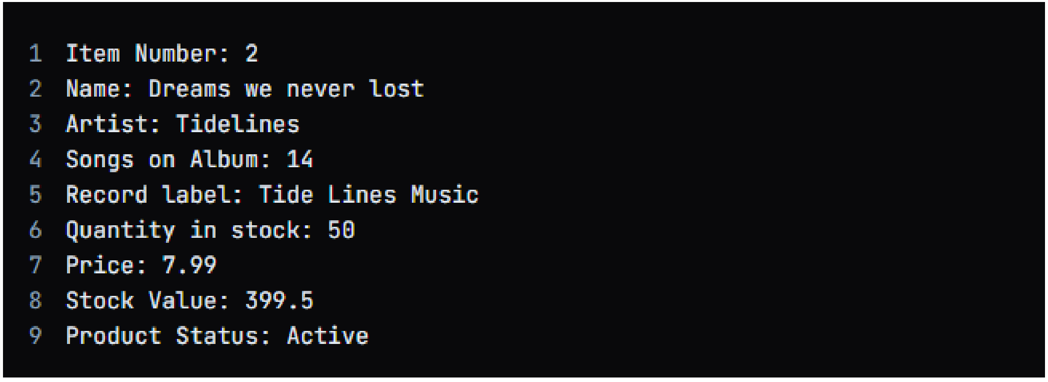
```
// CD.java
public class CD extends Product {
  private String artist;
  private int songsOnAlbum;
  private String recordLabel;

  public CD(int itemNumber, String name, double price, int quantityInStock, boolean
isActive, String artist, int songsOnAlbum, String recordLabel) {
    super(itemNumber, name, price, quantityInStock, isActive);
    this.artist = artist;
    this.songsOnAlbum = songsOnAlbum;
    this.recordLabel = recordLabel;
  }

  public String getArtist() {
    return artist;
  }

  public int getSongsOnAlbum() {
    return songsOnAlbum;
  }

  public String getRecordLabel() {
    return recordLabel;
  }

  @Override
  public String toString() {
    return super.toString() + "\n" +
        "Artist: " + artist + "\n" +
        "Songs on Album: " + songsOnAlbum + "\n" +
        "Record Label: " + recordLabel;
  }
}

// ProductTester.java
import java.util.Scanner;

public class ProductTester {
  private Product[] products;
  private Scanner scanner;

  public ProductTester(int size) {
    products = new Product[size];
    scanner = new Scanner(System.in);
  }

  public void addProductToInventory(int i, Product product) {
    products[i] = product;
  }
  public void addCDToInventory(int i) {
    System.out.print("Please enter the CD name: ");
    String name = scanner.nextLine();
    System.out.print("Please enter the artist: ");
    String artist = scanner.nextLine();
    System.out.print("Please enter the number of songs on the album: ");
    int songsOnAlbum = scanner.nextInt();
    scanner.nextLine(); // consume newline left-over
    System.out.print("Please enter the record label: ");
    String recordLabel = scanner.nextLine();
    System.out.print("Please enter the price: ");
    double price = scanner.nextDouble();
    scanner.nextLine(); // consume newline left-over
    System.out.print("Please enter the quantity in stock: ");
    int quantityInStock = scanner.nextInt();
    scanner.nextLine(); // consume newline left-over
    boolean isActive = true;
    CD cd = new CD(i, name, price, quantityInStock, isActive, artist, songsOnAlbum, recordLabel);
    addProductToInventory
```

DVD Output:



```
Item Number: 1
Name: Daredevil
Movie Length: 99
Age Rating: 15
Film Studio: 20th Century Fox
Quantity in stock: 50
Price: 8.99
Stock Value: 471.975
Product Status: Active
```

CD Output:



```
Item Number: 2
Name: Dreams we never lost
Artist: Tidelines
Songs on Album: 14
Record label: Tide Lines Music
Quantity in stock: 50
Price: 7.99
Stock Value: 399.5
Product Status: Active
```