PROG ASSIGNMENT

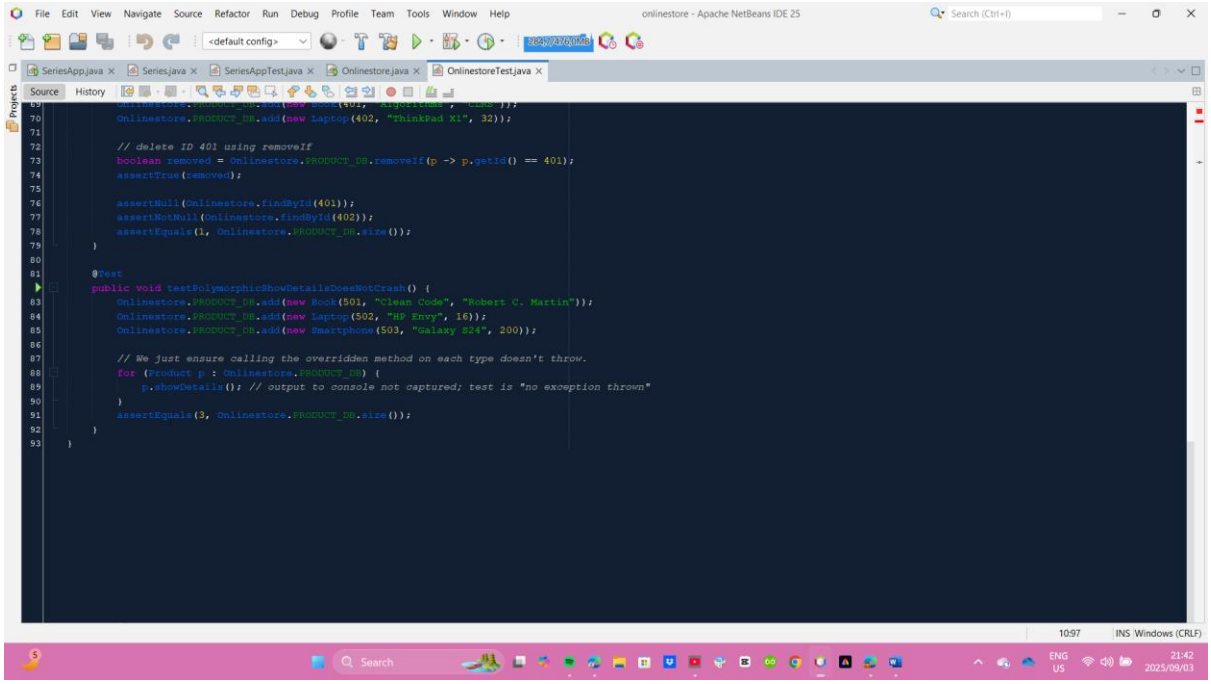
DIVINE .Z.SULUBIKA -ST10485027(GR.3)

SECTION A :

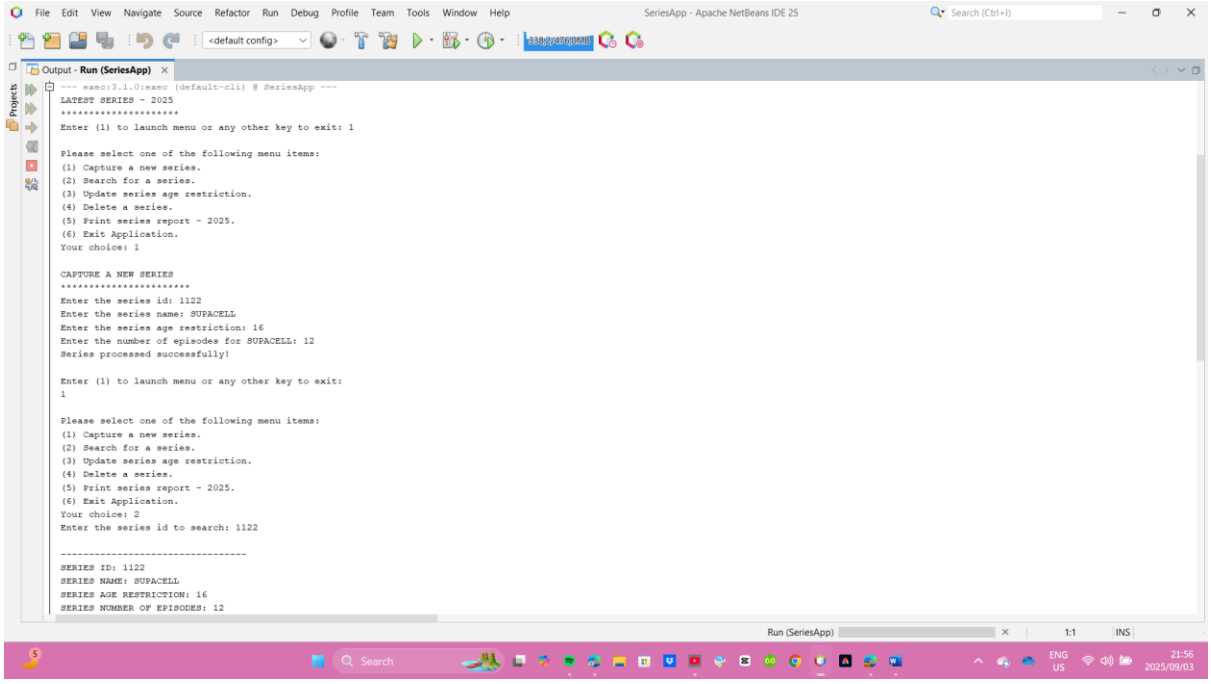
MAIN CLASS: SeriesApp.Java

```java
package tutorial.onlinestore;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import static org.junit.Assert.*;

/*
 ========================Tests for the Online Store Inventory================================
 */
public class OnlinestoreTest {

    @Before
    public void setUp() {
        Onlinestore.PRODUCT_DB.clear();
    }

    @After
    public void tearDown() {
        Onlinestore.PRODUCT_DB.clear();
    }

    @Test
    public void testAddProducts() {
        Book b = new Book(101, "Java Basics", "James Gosling");
        Laptop l = new Laptop(102, "Dell XPS 13", 16);
        Smartphone s = new Smartphone(103, "Pixel 9", 50);

        Onlinestore.PRODUCT_DB.add(b);
        Onlinestore.PRODUCT_DB.add(l);
        Onlinestore.PRODUCT_DB.add(s);

        assertEquals(3, Onlinestore.PRODUCT_DB.size());
        assertEquals("Java Basics", Onlinestore.PRODUCT_DB.get(0).getName());
        assertTrue(Onlinestore.PRODUCT_DB.get(1) instanceof Laptop);
        assertTrue(Onlinestore.PRODUCT_DB.get(2) instanceof Smartphone);
```

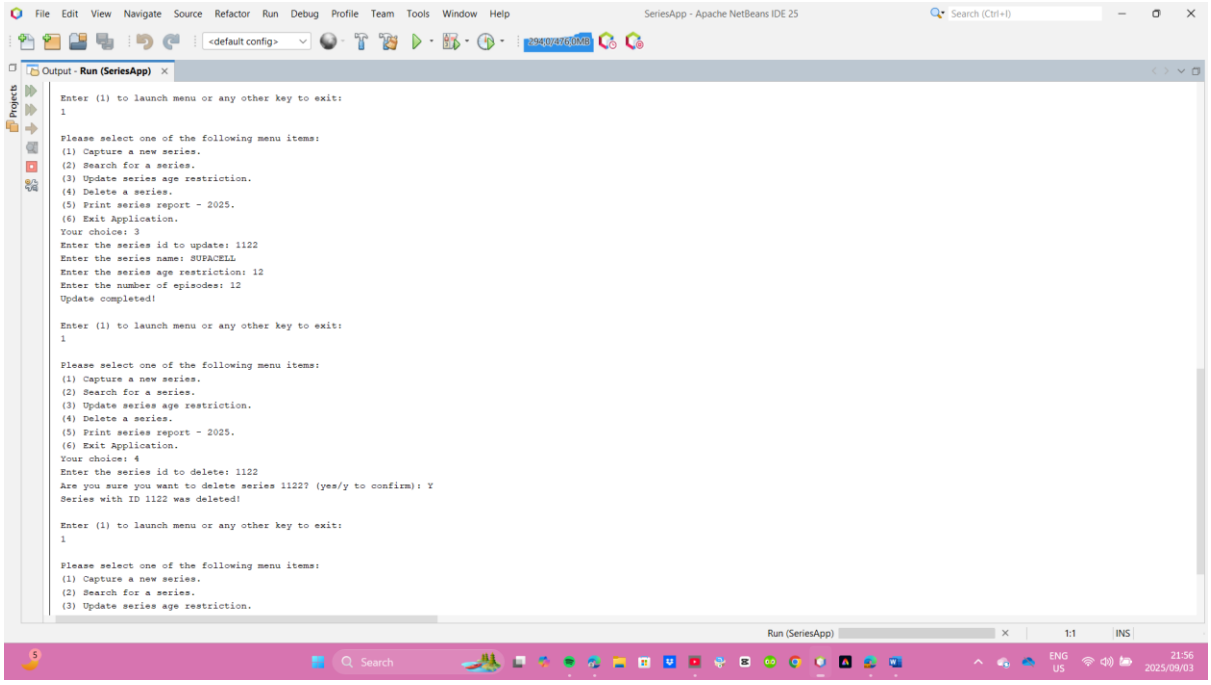
```java
        assertTrue(Onlinestore.PRODUCT_DB.get(2) instanceof Smartphone);
    }

    @Test
    public void testFindById() {
        Onlinestore.PRODUCT_DB.add(new Book(201, "OOP in Java", "Oracle"));
        Onlinestore.PRODUCT_DB.add(new Laptop(202, "MacBook Air", 8));

        Product p1 = Onlinestore.findById(201);
        Product p2 = Onlinestore.findById(202);
        Product p3 = Onlinestore.findById(999); // not present

        assertNotNull(p1);
        assertEquals("OOP in Java", p1.getName());
        assertTrue(p2 instanceof Laptop);
        assertNull(p3);
    }

    @Test
    public void testUpdateName() {
        Smartphone s = new Smartphone(301, "iPhone 15", 48);
        Onlinestore.PRODUCT_DB.add(s);

        Product target = Onlinestore.findById(301);
        assertNotNull(target);

        target.setName("iPhone 16 Pro");
        assertEquals("iPhone 16 Pro", Onlinestore.findById(301).getName());
    }

    @Test
    public void testDeleteById() {
        Onlinestore.PRODUCT_DB.add(new Book(401, "Algorithms", "CLRS"));
        Onlinestore.PRODUCT_DB.add(new Laptop(402, "ThinkPad X1", 32));

        // delete ID 401 using removeIf
        boolean removed = Onlinestore.PRODUCT_DB.removeIf(p -> p.getId() == 401);
```


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
onlinestore - Apache NetBeans IDE 25
SeriesApp.java
Series.java
SeriesAppTest.java
Onlinestore.java
OnlinestoreTest.java
Source
History
Onlinestore.PRODUCT_DB.add(new Laptop(402, "ThinkPad X1", 32));
// delete ID 401 using removeIf
boolean removed = Onlinestore.PRODUCT_DB.removeIf(p -> p.getId() == 401);
assertTrue(removed);
assertNull(Onlinestore.findById(401));
assertNotNull(Onlinestore.findById(402));
assertEquals(1, Onlinestore.PRODUCT_DB.size());
}
@Test
public void testPolymorphicShowDetailsDoesNotCrash() {
Onlinestore.PRODUCT_DB.add(new Book(501, "Clean Code", "Robert C. Martin"));
Onlinestore.PRODUCT_DB.add(new Laptop(502, "HP Envy", 16));
Onlinestore.PRODUCT_DB.add(new Smartphone(503, "Galaxy S24", 200));
// We just ensure calling the overridden method on each type doesn't throw.
for (Product p : Onlinestore.PRODUCT_DB) {
p.showDetails(); // output to console not captured; test is "no exception thrown"
}
assertEquals(3, Onlinestore.PRODUCT_DB.size());
}
}
10:97
INS Windows (CRLF)

OUTPUT:


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
SeriesApp - Apache NetBeans IDE 25
Output - Run (SeriesApp)
--- exec:3.1.0:exec (default-cli) @ SeriesApp ---
LATEST SERIES - 2025
****
Enter (1) to launch menu or any other key to exit: 1
Please select one of the following menu items:
(1) Capture a new series.
(2) Search for a series.
(3) Update series age restriction.
(4) Delete a series.
(5) Print series report - 2025.
(6) Exit Application.
Your choice: 1
CAPTURE A NEW SERIES
****
Enter the series id: 1122
Enter the series name: SUPACELL
Enter the series age restriction: 16
Enter the number of episodes for SUPACELL: 12
Series processed successfully!
Enter (1) to launch menu or any other key to exit:
1
Please select one of the following menu items:
(1) Capture a new series.
(2) Search for a series.
(3) Update series age restriction.
(4) Delete a series.
(5) Print series report - 2025.
(6) Exit Application.
Your choice: 2
Enter the series id to search: 1122

SERIES ID: 1122
SERIES NAME: SUPACELL
SERIES AGE RESTRICTION: 16
SERIES NUMBER OF EPISODES: 12
Run (SeriesApp)
1:1
INS




```
Please select one of the following menu items:
(1) Capture a new series.
(2) Search for a series.
(3) Update series age restriction.
(4) Delete a series.
(5) Print series report - 2025.
(6) Exit Application.
Your choice: 5
No series captured yet.

Enter (1) to launch menu or any other key to exit:
```

CODE:

MAIN CLASS

//SECTION A :

```
package seriesapp;


import java.util.ArrayList;

import java.util.Iterator;
```

```
import java.util.List;
import java.util.Scanner;

public class SeriesApp {
   public static final Scanner input = new Scanner(System.in);
  public static final List<Series> SERIES_DB = new ArrayList<>();

   public static void main(String[] args) {
      System.out.println("LATEST SERIES - 2025");
      System.out.println("*********************");
      System.out.print("Enter (1) to launch menu or any other key to exit: ");
      String launch = input.nextLine();

      if (!"1".equals(launch)) {
         System.out.println("Goodbye!");
         return;
      }

      boolean running = true;
      while (running) {
         printMenu();
         String choice = input.nextLine();

         switch (choice) {
            case "1" -> captureNewSeries();
            case "2" -> searchSeriesById();
            case "3" -> updateSeries();
            case "4" -> deleteSeries();
```

```
            case "5" -> printReport();
            case "6" -> {
                System.out.println("Exiting application...");
                running = false;
            }
            default -> System.out.println("Invalid option! Please choose 1-6.");
        }

        if (running) {
            System.out.println("\nEnter (1) to launch menu or any other key to exit:");
            String again = input.nextLine();
            if (!"1".equals(again)) {
                System.out.println("Goodbye!");
                break;
            }
        }
    }
}

// 1.1) Menu
public static void printMenu() {
    System.out.println("\nPlease select one of the following menu items:");
    System.out.println("(1) Capture a new series.");
    System.out.println("(2) Search for a series.");
    System.out.println("(3) Update series age restriction.");
    System.out.println("(4) Delete a series.");
    System.out.println("(5) Print series report - 2025.");
    System.out.println("(6) Exit Application.");
```

```
    System.out.print("Your choice: ");
  }

  // 1.2 & 1.3) Capture new series
  public static void captureNewSeries() {
    System.out.println("\nCAPTURE A NEW SERIES");
    System.out.println("************************");

    int id = readInt("Enter the series id: ");
    System.out.print("Enter the series name: ");
    String name = input.nextLine();
    int age = readAgeRestrictionValidated();
    int episodes = readInt("Enter the number of episodes for " + name + ": ");

    SERIES_DB.add(new Series(id, name, age, episodes));
    System.out.println("Series processed successfully!");
  }

public static int readAgeRestrictionValidated() {
    while (true) {
      System.out.print("Enter the series age restriction: ");
      String ageStr = input.nextLine();
      if (!isInteger(ageStr)) {
        System.out.println("You have entered an incorrect series age! Please re-enter.");
        continue;
      }
      int age = Integer.parseInt(ageStr);
      if (age < 2 || age > 18) {
```

```
            System.out.println("You have entered an incorrect series age! Please re-enter.");
            continue;
        }
        return age;
    }
}

// 1.5) Search by ID
public static void searchSeriesById() {
    int id = readInt("Enter the series id to search: ");
    Series s = findById(id);
    if (s == null) {
        System.out.println("\n---------------------------------");
        System.out.println("Series with ID " + id + " was not found!");
        System.out.println("---------------------------------");
    } else {
        System.out.println("\n---------------------------------");
        System.out.println("SERIES ID: " + s.getId());
        System.out.println("SERIES NAME: " + s.getName());
        System.out.println("SERIES AGE RESTRICTION: " + s.getAgeRestriction());
        System.out.println("SERIES NUMBER OF EPISODES: " + s.getEpisodes());
        System.out.println("---------------------------------");
    }
}

// 1.6) Update
public static void updateSeries() {
    int id = readInt("Enter the series id to update: ");
```

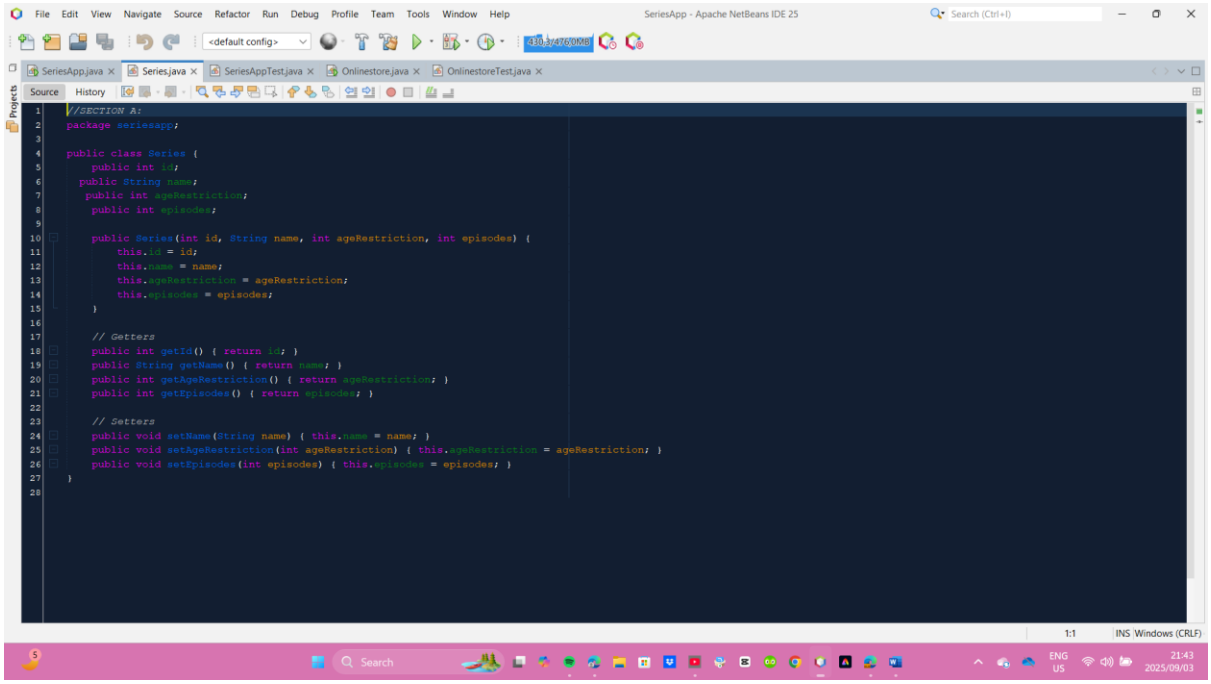
```
    Series s = findById(id);
    if (s == null) {
        System.out.println("Series with ID " + id + " was not found!");
        return;
    }

    System.out.print("Enter the series name: ");
    String newName = input.nextLine();
    int newAge = readAgeRestrictionValidated();
    int newEpisodes = readInt("Enter the number of episodes: ");

    s.setName(newName);
    s.setAgeRestriction(newAge);
    s.setEpisodes(newEpisodes);

    System.out.println("Update completed!");
}

// 1.7) Delete
public static void deleteSeries() {
    int id = readInt("Enter the series id to delete: ");
    Series s = findById(id);
    if (s == null) {
        System.out.println("Series with ID " + id + " was not found!");
        return;
    }
```

```
        System.out.print("Are you sure you want to delete series " + id + "? (yes/y to confirm): ");
        String confirm = input.nextLine().toLowerCase();
        if (confirm.equals("y") || confirm.equals("yes")) {
            Iterator<Series> it = SERIES_DB.iterator();
            while (it.hasNext()) {
                if (it.next().getId() == id) {
                    it.remove();
                    break;
                }
            }
            System.out.println("Series with ID " + id + " was deleted!");
        } else {
            System.out.println("Delete cancelled.");
        }
    }

    // 1.8) Report
    public  static void printReport() {
        if (SERIES_DB.isEmpty()) {
            System.out.println("No series captured yet.");
            return;
        }

        for (int i = 0; i < SERIES_DB.size(); i++) {
            Series s = SERIES_DB.get(i);
            System.out.println("\nSeries " + (i + 1));
            System.out.println("--------------------------------");
```

```
        System.out.println("SERIES ID: " + s.getId());
        System.out.println("SERIES NAME: " + s.getName());
        System.out.println("SERIES AGE RESTRICTION: " + s.getAgeRestriction());
        System.out.println("NUMBER OF EPISODES: " + s.getEpisodes());
        System.out.println("---------------------------------");
    }
}

// Helpers
public static Series findById(int id) {
    for (Series s : SERIES_DB) {
        if (s.getId() == id) return s;
    }
    return null;
}

public static int readInt(String prompt) {
    while (true) {
        System.out.print(prompt);
        String str = input.nextLine();
        if (isInteger(str)) {
            return Integer.parseInt(str);
        }
        System.out.println("Please enter a valid number.");
    }
}

public static boolean isInteger(String s) {
```

```
        if (s == null || s.isEmpty()) return false;
        try {
            Integer.parseInt(s);
            return true;
        } catch (NumberFormatException e) {
            return false;
        }
    }
}
```

SERIES CLASS: Series.java

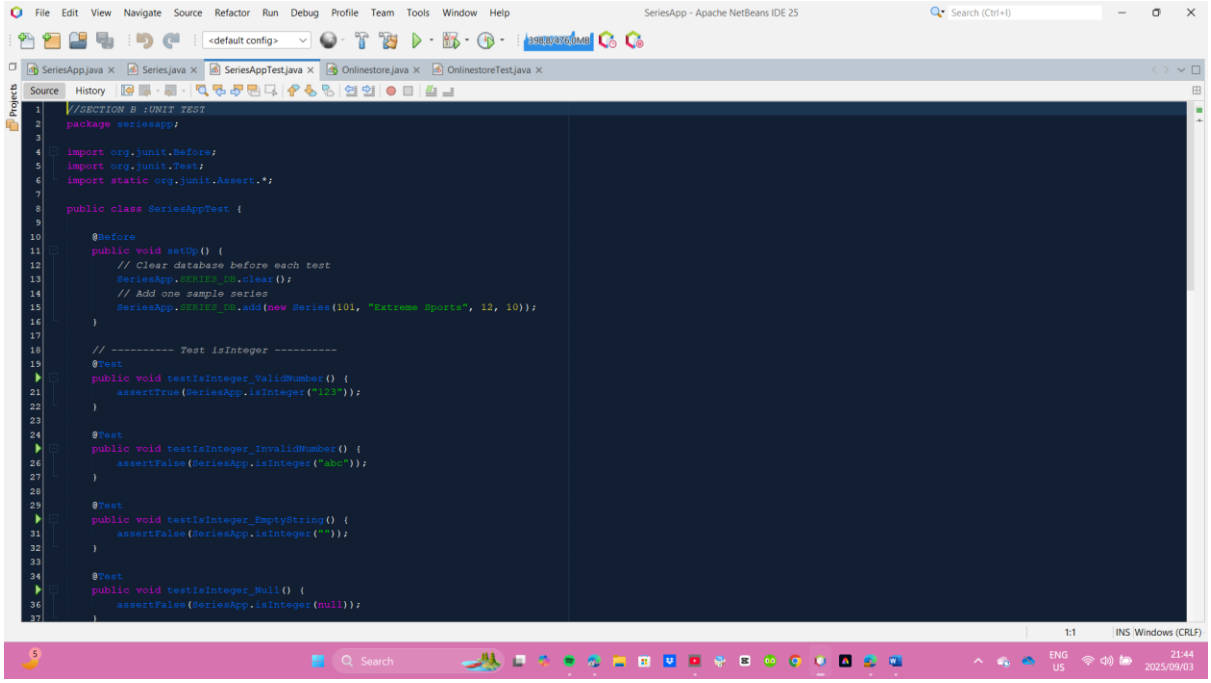


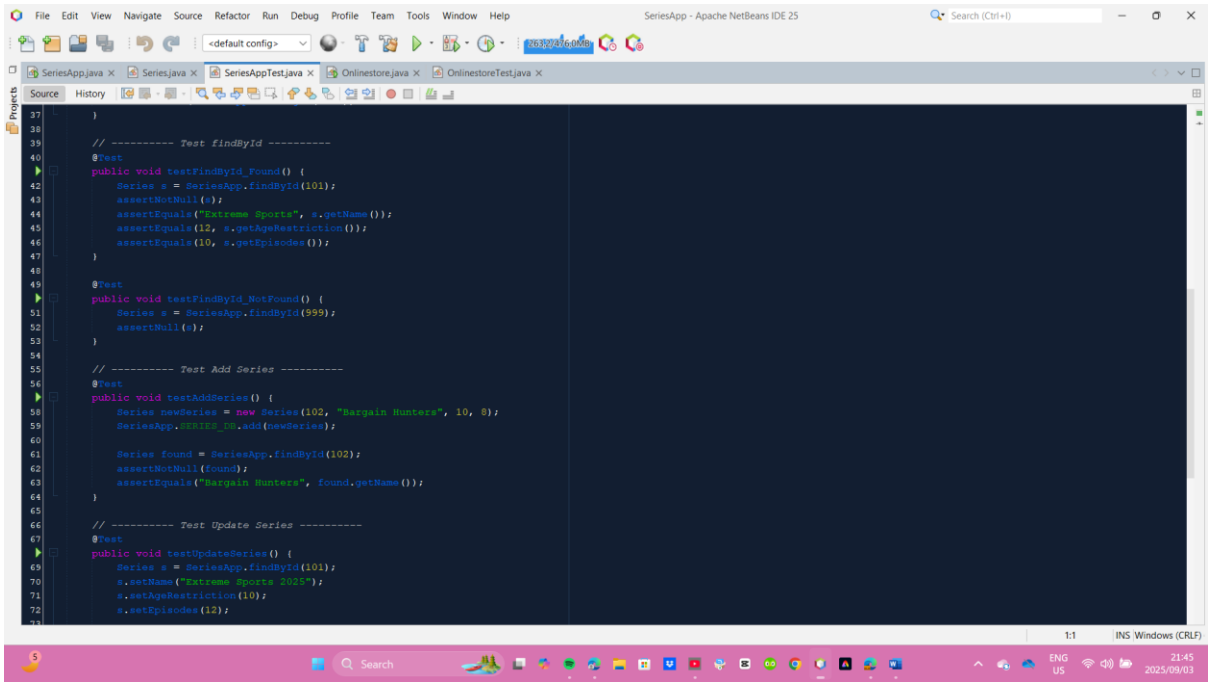
CODE: SERIES CLASS

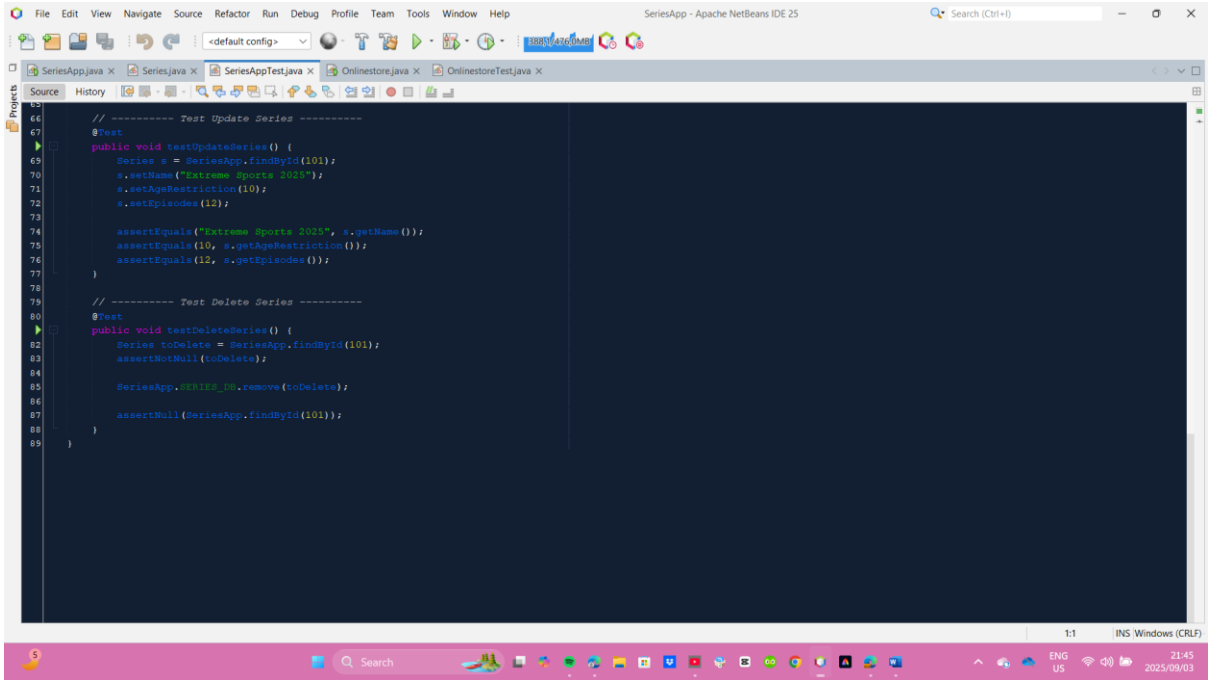
//SECTION A:

```
package seriesapp;

public class Series {
   public int id;
  public String name;
  public int ageRestriction;
   public int episodes;

   public Series(int id, String name, int ageRestriction, int episodes) {
      this.id = id;
      this.name = name;
      this.ageRestriction = ageRestriction;
      this.episodes = episodes;
   }

   // Getters
   public int getId() { return id; }
   public String getName() { return name; }
   public int getAgeRestriction() { return ageRestriction; }
   public int getEpisodes() { return episodes; }

   // Setters
   public void setName(String name) { this.name = name; }
   public void setAgeRestriction(int ageRestriction) { this.ageRestriction =
ageRestriction; }
   public void setEpisodes(int episodes) { this.episodes = episodes; }
}
```

SECTION B: UNIT TESTING





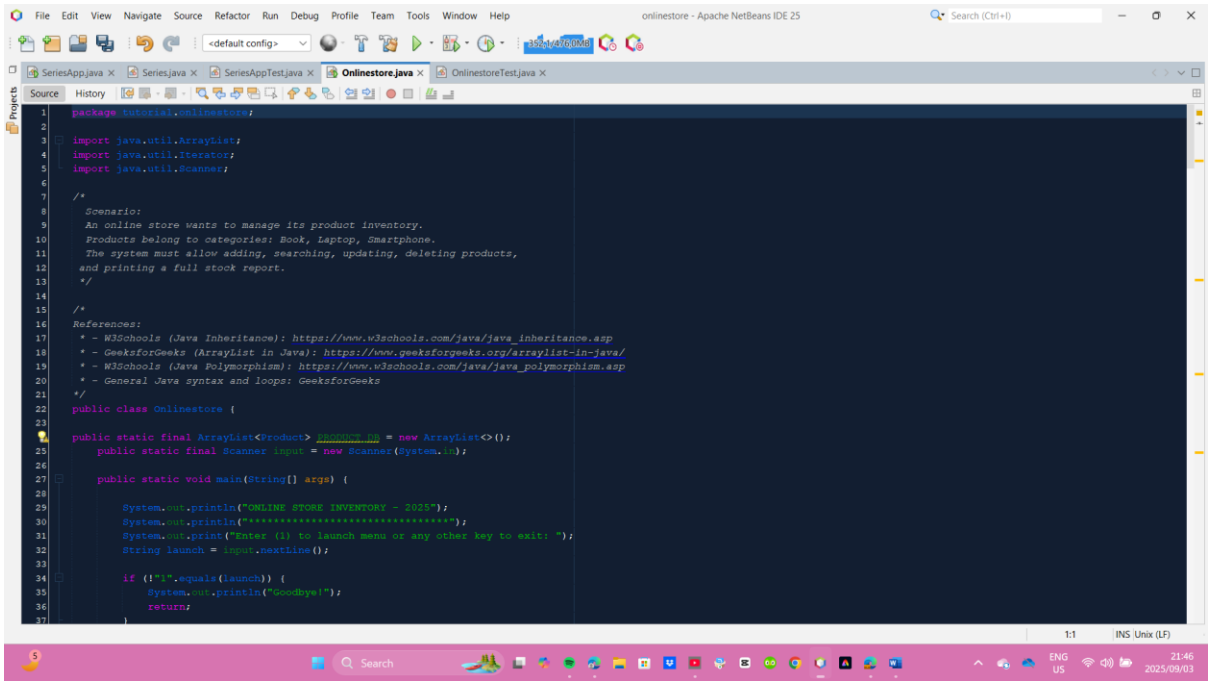


```java
    // ---------- Test Update Series ----------
    @Test
    public void testUpdateSeries() {
        Series s = SeriesApp.findById(101);
        s.setName("Extreme Sports 2025");
        s.setAgeRestriction(10);
        s.setEpisodes(12);

        assertEquals("Extreme Sports 2025", s.getName());
        assertEquals(10, s.getAgeRestriction());
        assertEquals(12, s.getEpisodes());
    }

    // ---------- Test Delete Series ----------
    @Test
    public void testDeleteSeries() {
        Series toDelete = SeriesApp.findById(101);
        assertNotNull(toDelete);

        SeriesApp.SERIES_DB.remove(toDelete);

        assertNull(SeriesApp.findById(101));
    }
}
```

SECTION B : OWN APP

Scenario:

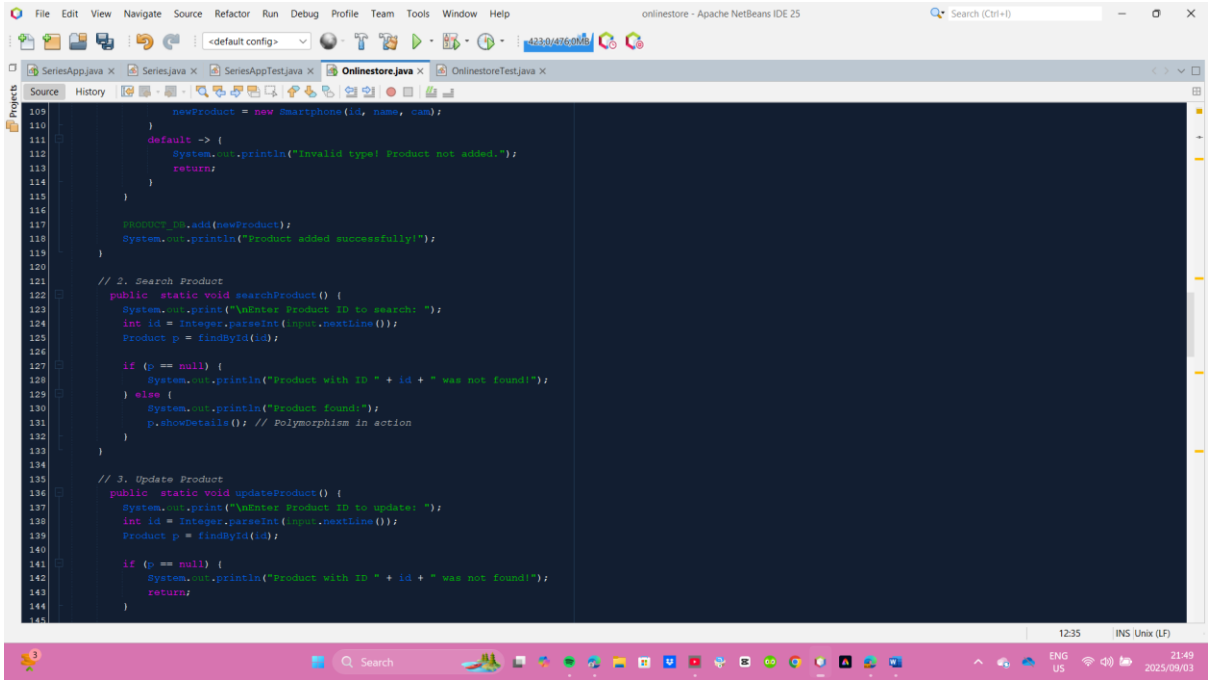
An online store wants to manage its product inventory. Products belong to categories: Book, Laptop, Smartphone. The system must allow adding, searching, updating, deleting products, and printing a full stock report.MAIN CLASS: Onlinestore.java



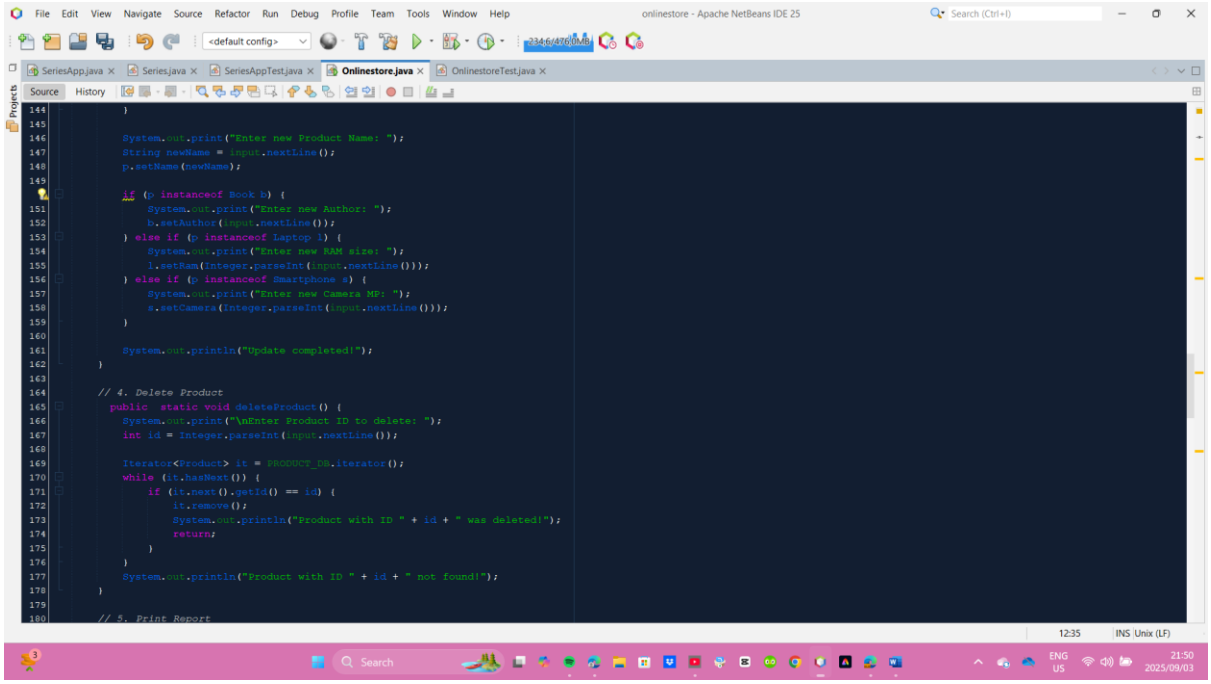
```java
package tutorial.onlinestore;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

/*
  Scenario:
  An online store wants to manage its product inventory.
  Products belong to categories: Book, Laptop, Smartphone.
  The system must allow adding, searching, updating, deleting products,
 and printing a full stock report.
 */

/*
References:
 * - W3Schools (Java Inheritance): https://www.w3schools.com/java/java_inheritance.asp
 * - GeeksforGeeks (ArrayList in Java): https://www.geeksforgeeks.org/arraylist-in-java/
 * - W3Schools (Java Polymorphism): https://www.w3schools.com/java/java_polymorphism.asp
 * - General Java syntax and loops: GeeksforGeeks
*/
public class Onlinestore {

public static final ArrayList<Product> PRODUCT_DB = new ArrayList<>();
    public static final Scanner input = new Scanner(System.in);

    public static void main(String[] args) {

        System.out.println("ONLINE STORE INVENTORY - 2025");
        System.out.println("******************************");
        System.out.print("Enter (1) to launch menu or any other key to exit: ");
        String launch = input.nextLine();

        if (!"1".equals(launch)) {
            System.out.println("Goodbye!");
            return;
        }
```

```java
        }

        boolean running = true;
        while (running) {
            printMenu();
            String choice = input.nextLine();

            switch (choice) {
                case "1" -> addProduct();
                case "2" -> searchProduct();
                case "3" -> updateProduct();
                case "4" -> deleteProduct();
                case "5" -> printReport();
                case "6" -> {
                    System.out.println("Exiting application...");
                    running = false;
                }
                default -> System.out.println("Invalid option! Please choose 1-6.");
            }

            if (running) {
                System.out.print("\nEnter (1) to return to menu or any other key to exit: ");
                String again = input.nextLine();
                if (!"1".equals(again)) {
                    System.out.println("Goodbye!");
                    break;
                }
            }
        }
    }

    // Menu
      public  static void printMenu() {
       System.out.println("\nPlease select one of the following menu items:");
       System.out.println("(1) Add a new product");
       System.out.println("(2) Search for a product");
```

```java
       System.out.println("(3) Update product details");
       System.out.println("(4) Delete a product");
       System.out.println("(5) Print inventory report");
       System.out.println("(6) Exit application");
       System.out.print("Your choice: ");
    }

    // 1. Add Product
     public  static void addProduct() {
       System.out.println("\nADD A NEW PRODUCT");
       System.out.println("*********************************");

       System.out.print("Enter Product ID: ");
       int id = Integer.parseInt(input.nextLine());

       System.out.print("Enter Product Name: ");
       String name = input.nextLine();

       System.out.println("Select Product Type: (1) Book, (2) Laptop, (3) Smartphone");
       String type = input.nextLine();

       Product newProduct = null;
       switch (type) {
           case "1" -> {
               System.out.print("Enter Author Name: ");
               String author = input.nextLine();
               newProduct = new Book(id, name, author);
           }
           case "2" -> {
               System.out.print("Enter RAM size (GB): ");
               int ram = Integer.parseInt(input.nextLine());
               newProduct = new Laptop(id, name, ram);
           }
           case "3" -> {
               System.out.print("Enter Camera (MP): ");
               int cam = Integer.parseInt(input.nextLine());
```



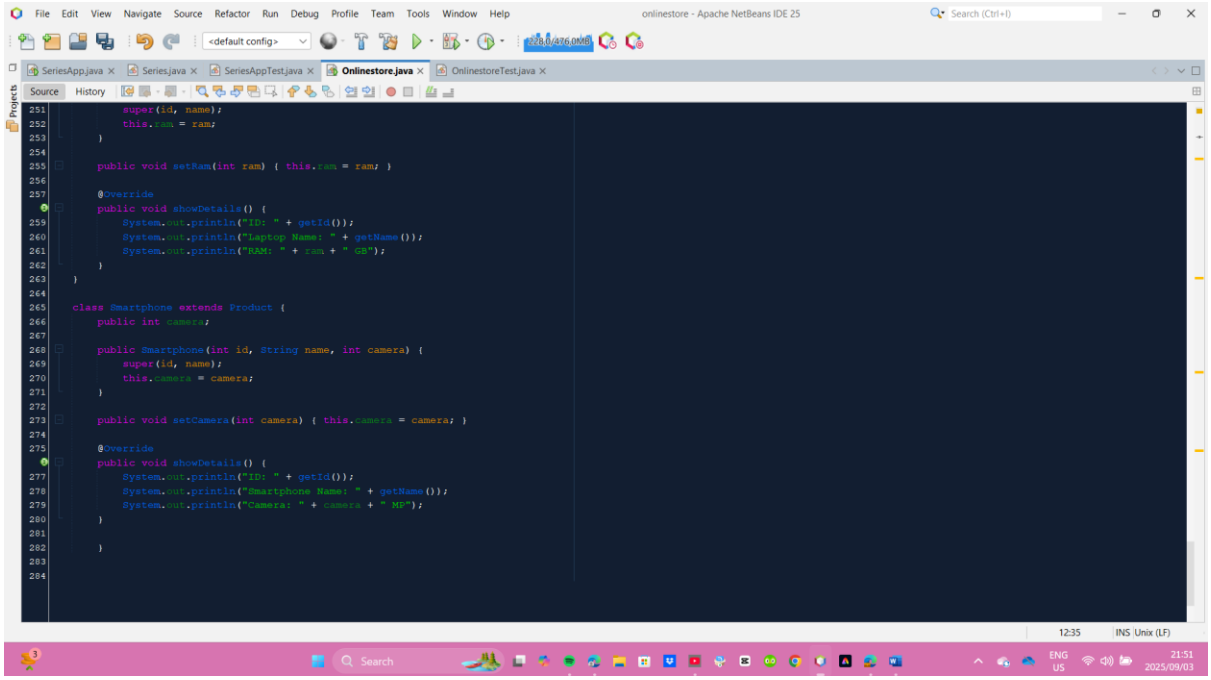
```java
                newProduct = new Smartphone(id, name, cam);
            }
            default -> {
                System.out.println("Invalid type! Product not added.");
                return;
            }
        }

        PRODUCT_DB.add(newProduct);
        System.out.println("Product added successfully!");
    }

    // 2. Search Product
     public  static void searchProduct() {
        System.out.print("\nEnter Product ID to search: ");
        int id = Integer.parseInt(input.nextLine());
        Product p = findById(id);

        if (p == null) {
            System.out.println("Product with ID " + id + " was not found!");
        } else {
            System.out.println("Product found:");
            p.showDetails(); // Polymorphism in action
        }
    }

    // 3. Update Product
     public  static void updateProduct() {
        System.out.print("\nEnter Product ID to update: ");
        int id = Integer.parseInt(input.nextLine());
        Product p = findById(id);

        if (p == null) {
            System.out.println("Product with ID " + id + " was not found!");
            return;
        }
```



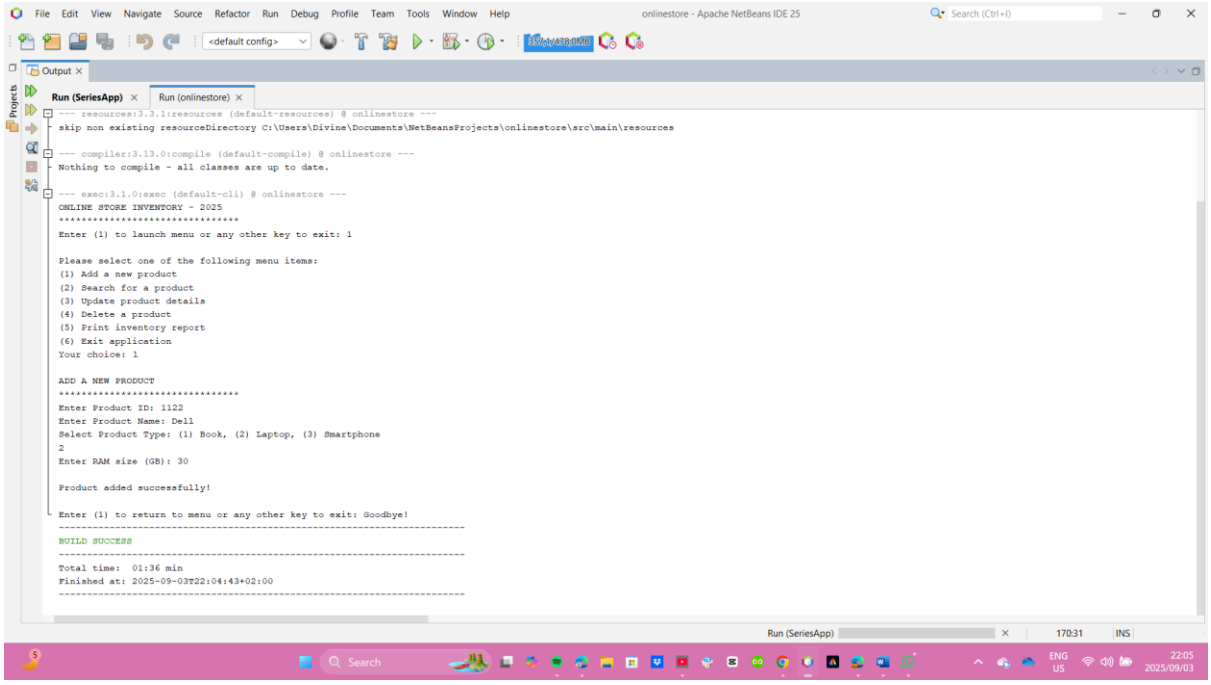
```java
        }

        System.out.print("Enter new Product Name: ");
        String newName = input.nextLine();
        p.setName(newName);

        if (p instanceof Book b) {
            System.out.print("Enter new Author: ");
            b.setAuthor(input.nextLine());
        } else if (p instanceof Laptop l) {
            System.out.print("Enter new RAM size: ");
            l.setRam(Integer.parseInt(input.nextLine()));
        } else if (p instanceof Smartphone s) {
            System.out.print("Enter new Camera MP: ");
            s.setCamera(Integer.parseIns(input.nextLine()));
        }

        System.out.println("Update completed!");
    }

    // 4. Delete Product
     public  static void deleteProduct() {
        System.out.print("\nEnter Product ID to delete: ");
        int id = Integer.parseInt(input.nextLine());

        Iterator<Product> it = PRODUCT_DB.iterator();
        while (it.hasNext()) {
            if (it.next().getId() == id) {
                it.remove();
                System.out.println("Product with ID " + id + " was deleted!");
                return;
            }
        }
        System.out.println("Product with ID " + id + " not found!");
    }

    // 5. Print Report
```

```java
        // 5. Print Report
        public  static void printReport() {
            if (PRODUCT_DB.isEmpty()) {
                System.out.println("No products in inventory.");
                return;
            }
            System.out.println("\nINVENTORY REPORT");
            System.out.println("********************************");
            for (int i = 0; i < PRODUCT_DB.size(); i++) {
                System.out.println("Product " + (i + 1));
                PRODUCT_DB.get(i).showDetails(); // Polymorphism
                System.out.println("--------------------------------");
            }
        }

        // Helper: Find product by ID
    public static Product findById(int id) {
        for (Product p : PRODUCT_DB) {
            if (p.getId() == id) return p;
        }
        return null;
    }

    public static void addProductDirect(java.awt.print.Book b) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
    public ArrayList<Object> products;
}

/* ==============================
   Base Class + Subclasses
   ============================== */

abstract class Product {
        public int id;
        public  String name;
```

```java
        public  String name;

    public Product(int id, String name) {
        this.id = id;
        this.name = name;
    }

    public int getId() { return id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public abstract void showDetails(); // Overridden by subclasses
}

class Book extends Product {
        public String author;

    public Book(int id, String name, String author) {
        super(id, name);
        this.author = author;
    }

    public void setAuthor(String author) { this.author = author; }

    @Override
    public void showDetails() {
        System.out.println("ID: " + getId());
        System.out.println("Book Name: " + getName());
        System.out.println("Author: " + author);
    }
}

class Laptop extends Product {
    public  int ram;

    public Laptop(int id, String name, int ram) {
        super(id, name);
```


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
onlinestore - Apache NetBeans IDE 25
SeriesApp.java
Series.java
SeriesAppTest.java
Onlinestore.java
OnlinestoreTest.java
Source
History
super(id, name);
this.ram = ram;
}
public void setRam(int ram) { this.ram = ram; }
@Override
public void showDetails() {
System.out.println("ID: " + getId());
System.out.println("Laptop Name: " + getName());
System.out.println("RAM: " + ram + " GB");
}
}
class Smartphone extends Product {
public int camera;
public Smartphone(int id, String name, int camera) {
super(id, name);
this.camera = camera;
}
public void setCamera(int camera) { this.camera = camera; }
@Override
public void showDetails() {
System.out.println("ID: " + getId());
System.out.println("Smartphone Name: " + getName());
System.out.println("Camera: " + camera + " MP");
}
}
12:35
INS
Unix (LF)

OUTPUT:


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
onlinestore - Apache NetBeans IDE 25
Output
Run (SeriesApp)
Run (onlinestore)
--- resources:3.3.1:resources (default-resources) @ onlinestore ---
skip non existing resourceDirectory C:\Users\Divine\Documents\NetBeansProjects\onlinestore\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ onlinestore ---
Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ onlinestore ---
ONLINE STORE INVENTORY - 2025
****
Enter (1) to launch menu or any other key to exit: 1
Please select one of the following menu items:
(1) Add a new product
(2) Search for a product
(3) Update product details
(4) Delete a product
(5) Print inventory report
(6) Exit application
Your choice: 1
ADD A NEW PRODUCT
****
Enter Product ID: 1122
Enter Product Name: Dell
Select Product Type: (1) Book, (2) Laptop, (3) Smartphone
2
Enter RAM size (GB): 30
Product added successfully!
Enter (1) to return to menu or any other key to exit: Goodbye!
BUILD SUCCESS
Total time: 01:36 min
Finished at: 2025-09-03T22:04:43+02:00
Run (SeriesApp)
170:31
INS

CODE:

MAIN CLASS

```
package tutorial.onlinestore;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

/*
 Scenario:
 An online store wants to manage its product inventory.
 Products belong to categories: Book, Laptop, Smartphone.
 The system must allow adding, searching, updating, deleting products,
 and printing a full stock report.
 */

/*
References:
 * - W3Schools (Java Inheritance): https://www.w3schools.com/java/java_inheritance.asp
 * - GeeksforGeeks (ArrayList in Java): https://www.geeksforgeeks.org/arraylist-in-java/
 * - W3Schools (Java Polymorphism): https://www.w3schools.com/java/java_polymorphism.asp
 * - General Java syntax and loops: GeeksforGeeks
*/
public class Onlinestore {
```

```
public static final ArrayList<Product> PRODUCT_DB = new ArrayList<>();
  public static final Scanner input = new Scanner(System.in);

  public static void main(String[] args) {

    System.out.println("ONLINE STORE INVENTORY - 2025");
    System.out.println("********************************");
    System.out.print("Enter (1) to launch menu or any other key to exit: ");
    String launch = input.nextLine();

    if (!"1".equals(launch)) {
      System.out.println("Goodbye!");
      return;
    }

    boolean running = true;
    while (running) {
      printMenu();
      String choice = input.nextLine();

      switch (choice) {
        case "1" -> addProduct();
        case "2" -> searchProduct();
        case "3" -> updateProduct();
        case "4" -> deleteProduct();
        case "5" -> printReport();
        case "6" -> {
          System.out.println("Exiting application...");
```

```
            running = false;
          }
          default -> System.out.println("Invalid option! Please choose 1-6.");
        }

        if (running) {
          System.out.print("\nEnter (1) to return to menu or any other key to exit: ");
          String again = input.nextLine();
          if (!"1".equals(again)) {
            System.out.println("Goodbye!");
            break;
          }
        }
      }
    }

    // Menu
      public  static void printMenu() {
       System.out.println("\nPlease select one of the following menu items:");
       System.out.println("(1) Add a new product");
       System.out.println("(2) Search for a product");
       System.out.println("(3) Update product details");
       System.out.println("(4) Delete a product");
       System.out.println("(5) Print inventory report");
       System.out.println("(6) Exit application");
       System.out.print("Your choice: ");
    }
```

```
// 1. Add Product
  public  static void addProduct() {
    System.out.println("\nADD A NEW PRODUCT");
    System.out.println("********************************");

    System.out.print("Enter Product ID: ");
    int id = Integer.parseInt(input.nextLine());

    System.out.print("Enter Product Name: ");
    String name = input.nextLine();

    System.out.println("Select Product Type: (1) Book, (2) Laptop, (3) Smartphone");
    String type = input.nextLine();

    Product newProduct = null;
    switch (type) {
      case "1" -> {
        System.out.print("Enter Author Name: ");
        String author = input.nextLine();
        newProduct = new Book(id, name, author);
      }
      case "2" -> {
        System.out.print("Enter RAM size (GB): ");
        int ram = Integer.parseInt(input.nextLine());
        newProduct = new Laptop(id, name, ram);
      }
      case "3" -> {
        System.out.print("Enter Camera (MP): ");
```

```
            int cam = Integer.parseInt(input.nextLine());
            newProduct = new Smartphone(id, name, cam);
        }
        default -> {
            System.out.println("Invalid type! Product not added.");
            return;
        }
    }

    PRODUCT_DB.add(newProduct);
    System.out.println("Product added successfully!");
}

// 2. Search Product
  public  static void searchProduct() {
    System.out.print("\nEnter Product ID to search: ");
    int id = Integer.parseInt(input.nextLine());
    Product p = findById(id);

    if (p == null) {
        System.out.println("Product with ID " + id + " was not found!");
    } else {
        System.out.println("Product found:");
        p.showDetails(); // Polymorphism in action
    }
}

// 3. Update Product
```

```
public  static void updateProduct() {
  System.out.print("\nEnter Product ID to update: ");
  int id = Integer.parseInt(input.nextLine());
  Product p = findById(id);

  if (p == null) {
    System.out.println("Product with ID " + id + " was not found!");
    return;
  }

  System.out.print("Enter new Product Name: ");
  String newName = input.nextLine();
  p.setName(newName);

  if (p instanceof Book b) {
    System.out.print("Enter new Author: ");
    b.setAuthor(input.nextLine());
  } else if (p instanceof Laptop l) {
    System.out.print("Enter new RAM size: ");
    l.setRam(Integer.parseInt(input.nextLine()));
  } else if (p instanceof Smartphone s) {
    System.out.print("Enter new Camera MP: ");
    s.setCamera(Integer.parseInt(input.nextLine()));
  }

  System.out.println("Update completed!");
}
```

```
// 4. Delete Product
  public  static void deleteProduct() {
    System.out.print("\nEnter Product ID to delete: ");
    int id = Integer.parseInt(input.nextLine());

    Iterator<Product> it = PRODUCT_DB.iterator();
    while (it.hasNext()) {
      if (it.next().getId() == id) {
        it.remove();
        System.out.println("Product with ID " + id + " was deleted!");
        return;
      }
    }
    System.out.println("Product with ID " + id + " not found!");
}

// 5. Print Report
   public  static void printReport() {
    if (PRODUCT_DB.isEmpty()) {
      System.out.println("No products in inventory.");
      return;
    }
    System.out.println("\nINVENTORY REPORT");
    System.out.println("********************************");
    for (int i = 0; i < PRODUCT_DB.size(); i++) {
      System.out.println("Product " + (i + 1));
      PRODUCT_DB.get(i).showDetails(); // Polymorphism
      System.out.println("--------------------------------");
```

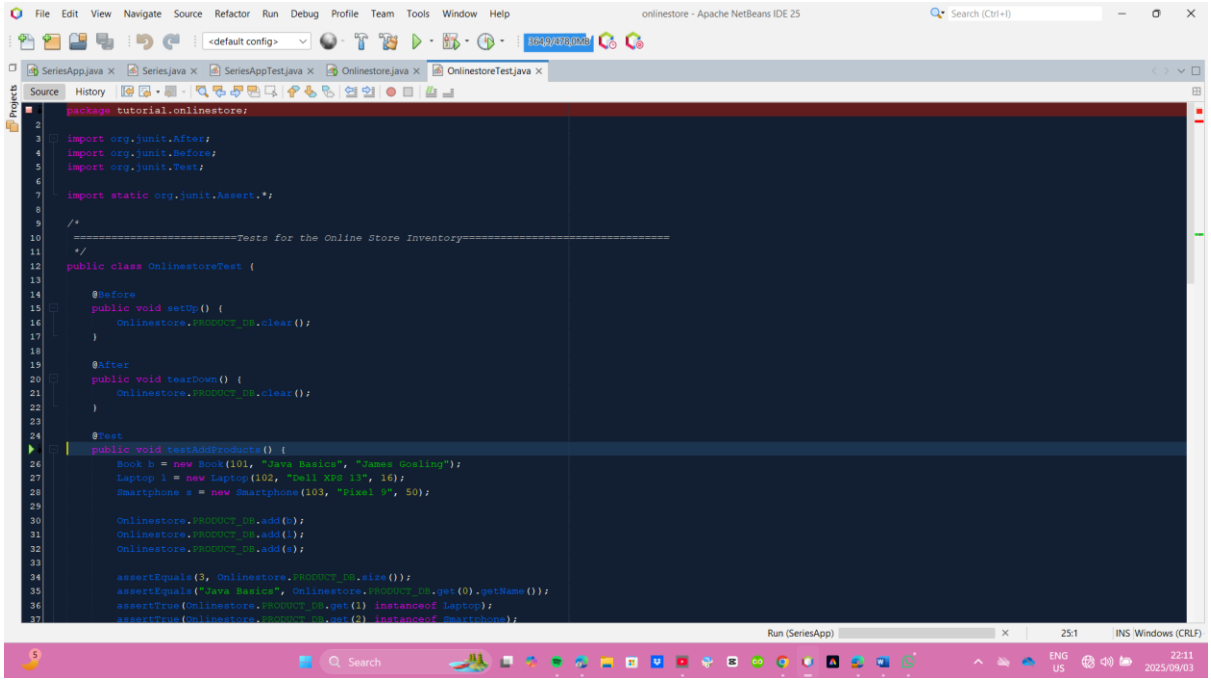
```
        }
    }

    // Helper: Find product by ID
    public static Product findById(int id) {
        for (Product p : PRODUCT_DB) {
            if (p.getId() == id) return p;
        }
        return null;
    }

    public static void addProductDirect(java.awt.print.Book b) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated
from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
    public ArrayList<Object> products;
}

/* ================================
   Base Class + Subclasses
   ================================ */

abstract class Product {
        public int id;
        public  String name;

    public Product(int id, String name) {
        this.id = id;
```

```
        this.name = name;
    }

    public int getId() { return id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public abstract void showDetails(); // Overridden by subclasses
}

class Book extends Product {
      public String author;

    public Book(int id, String name, String author) {
        super(id, name);
        this.author = author;
    }

    public void setAuthor(String author) { this.author = author; }

    @Override
    public void showDetails() {
        System.out.println("ID: " + getId());
        System.out.println("Book Name: " + getName());
        System.out.println("Author: " + author);
    }
}
```

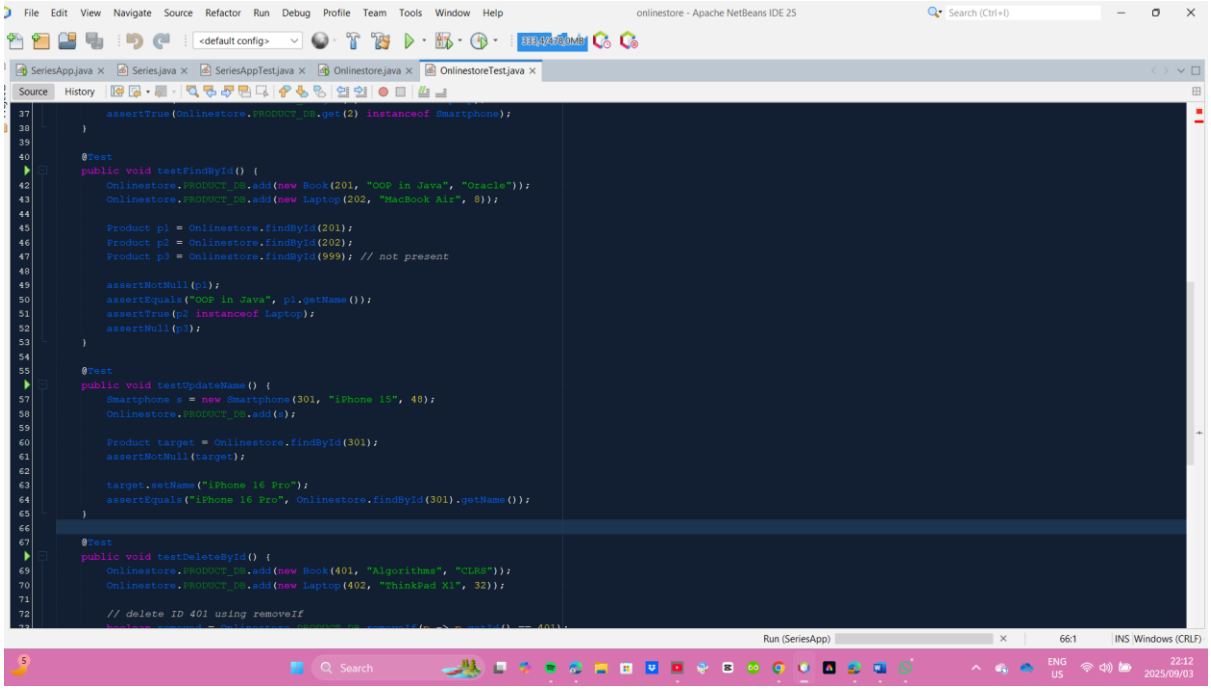
```
class Laptop extends Product {
  public  int ram;

   public Laptop(int id, String name, int ram) {
      super(id, name);
      this.ram = ram;
   }

   public void setRam(int ram) { this.ram = ram; }

   @Override
   public void showDetails() {
      System.out.println("ID: " + getId());
      System.out.println("Laptop Name: " + getName());
      System.out.println("RAM: " + ram + " GB");
   }
}

class Smartphone extends Product {
   public int camera;

   public Smartphone(int id, String name, int camera) {
      super(id, name);
      this.camera = camera;
   }

   public void setCamera(int camera) { this.camera = camera; }
```

```
@Override
public void showDetails() {
   System.out.println("ID: " + getId());
   System.out.println("Smartphone Name: " + getName());
   System.out.println("Camera: " + camera + " MP");
}


}
```

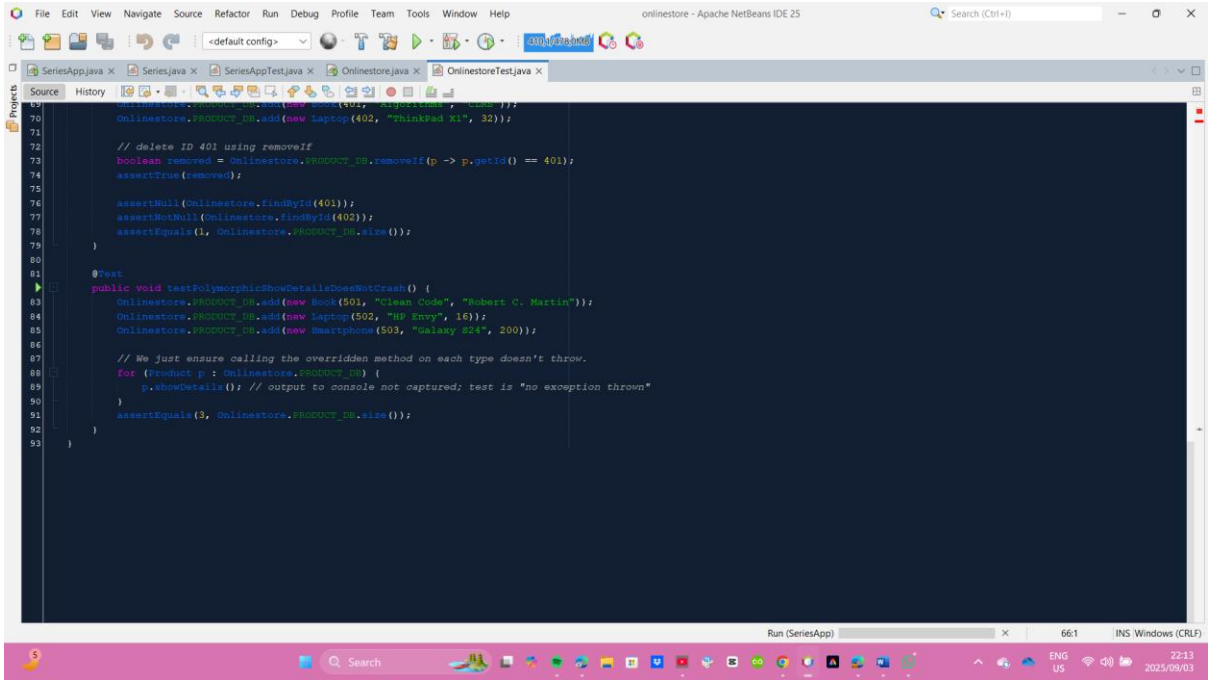
UNIT TESTING:



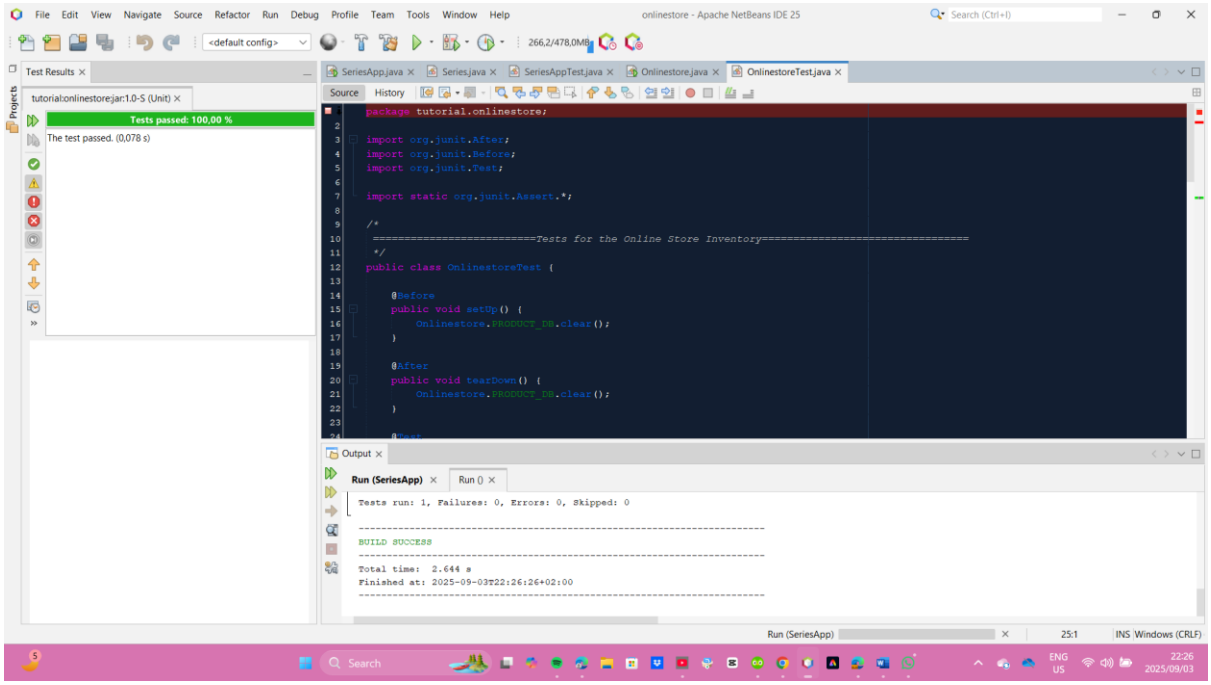
```java
package tutorial.onlinestore;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import static org.junit.Assert.*;

/*
 ========================Tests for the Online Store Inventory================================
 */
public class OnlinestoreTest {

    @Before
    public void setUp() {
        Onlinestore.PRODUCT_DB.clear();
    }

    @After
    public void tearDown() {
        Onlinestore.PRODUCT_DB.clear();
    }

    @Test
    public void testAddProducts() {
        Book b = new Book(101, "Java Basics", "James Gosling");
        Laptop l = new Laptop(102, "Dell XPS 13", 16);
        Smartphone s = new Smartphone(103, "Pixel 9", 50);

        Onlinestore.PRODUCT_DB.add(b);
        Onlinestore.PRODUCT_DB.add(l);
        Onlinestore.PRODUCT_DB.add(s);

        assertEquals(3, Onlinestore.PRODUCT_DB.size());
        assertEquals("Java Basics", Onlinestore.PRODUCT_DB.get(0).getName());
        assertTrue(Onlinestore.PRODUCT_DB.get(1) instanceof Laptop);
        assertTrue(Onlinestore.PRODUCT_DB.get(2) instanceof Smartphone);
```



```java
        assertTrue(Onlinestore.PRODUCT_DB.get(2) instanceof Smartphone);
    }

    @Test
    public void testFindById() {
        Onlinestore.PRODUCT_DB.add(new Book(201, "OOP in Java", "Oracle"));
        Onlinestore.PRODUCT_DB.add(new Laptop(202, "MacBook Air", 8));

        Product p1 = Onlinestore.findById(201);
        Product p2 = Onlinestore.findById(202);
        Product p3 = Onlinestore.findById(999); // not present

        assertNotNull(p1);
        assertEquals("OOP in Java", p1.getName());
        assertTrue(p2 instanceof Laptop);
        assertNull(p3);
    }

    @Test
    public void testUpdateName() {
        Smartphone s = new Smartphone(301, "iPhone 15", 48);
        Onlinestore.PRODUCT_DB.add(s);

        Product target = Onlinestore.findById(301);
        assertNotNull(target);

        target.setName("iPhone 16 Pro");
        assertEquals("iPhone 16 Pro", Onlinestore.findById(301).getName());
    }

    @Test
    public void testDeleteById() {
        Onlinestore.PRODUCT_DB.add(new Book(401, "Algorithms", "CLRS"));
        Onlinestore.PRODUCT_DB.add(new Laptop(402, "ThinkPad X1", 32));

        // delete ID 401 using removeIf
```


onlinestore - Apache NetBeans IDE 25
SeriesApp.java
Series.java
SeriesAppTest.java
Onlinestore.java
OnlinestoreTest.java
Source
History
Run (SeriesApp)
66:1
INS Windows (CRLF)

OUTPUT:


onlinestore - Apache NetBeans IDE 25
Test Results
tutorial.onlinestore.jar:1.0-S (Unit)
Tests passed: 100,00 %
The test passed. (0,078 s)
SeriesApp.java
Series.java
SeriesAppTest.java
Onlinestore.java
OnlinestoreTest.java
Source
History
package tutorial.onlinestore;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;
/*
==========Tests for the Online Store Inventory==========
*/
public class OnlinestoreTest {
@Before
public void setUp() {
Onlinestore.PRODUCT_DB.clear();
}
@After
public void tearDown() {
Onlinestore.PRODUCT_DB.clear();
}
Output
Run (SeriesApp)
Run ()
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
Total time: 2.644 s
Finished at: 2025-09-03T22:26:26+02:00
Run (SeriesApp)
25:1
INS Windows (CRLF)