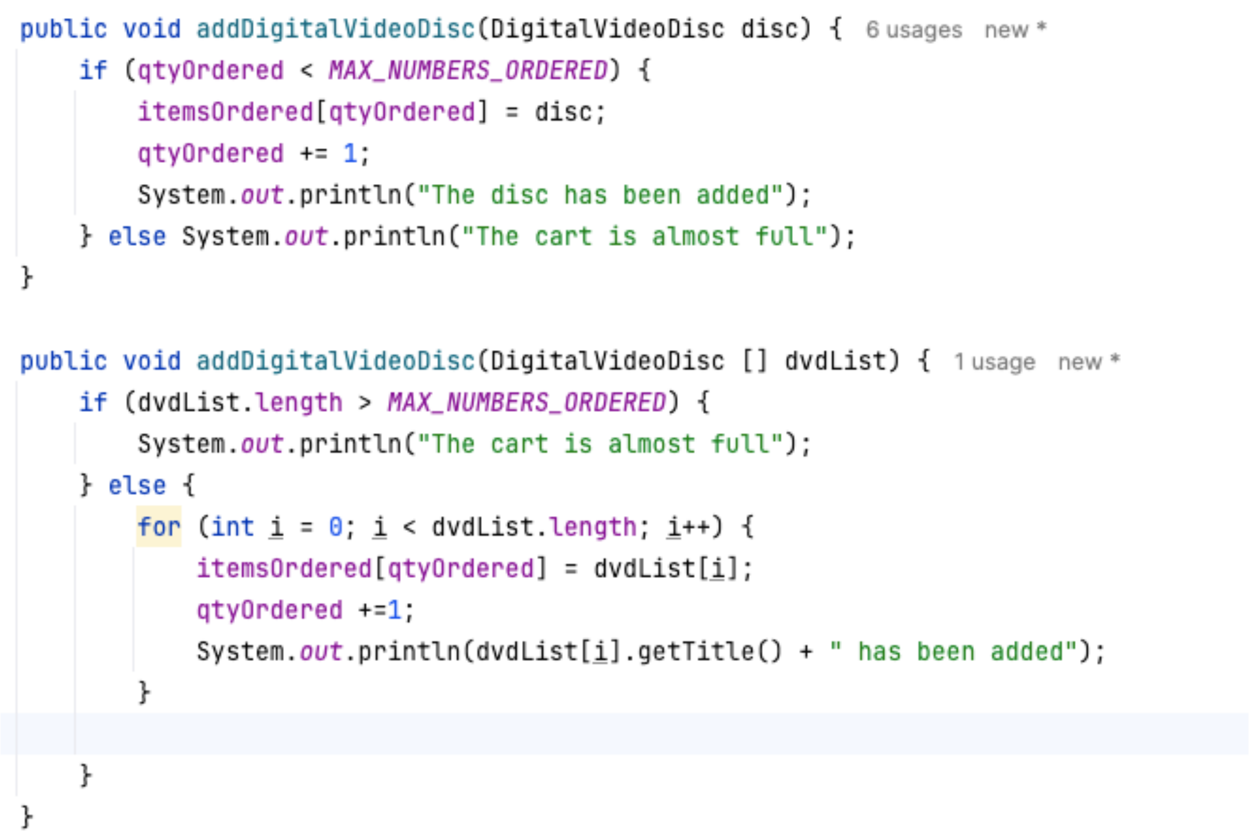
Report Object-Oriented Programming

Lab 03: Basic Object-Oriented Techniques

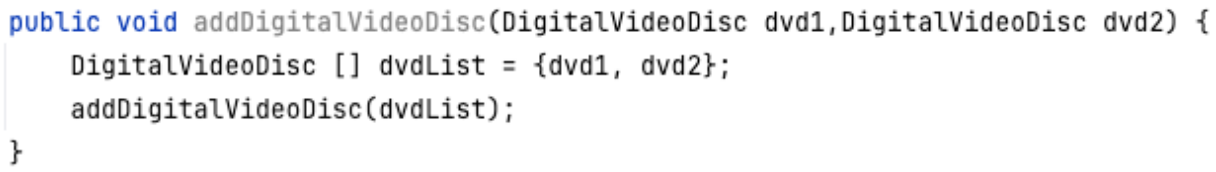
2. Working with method overloading

2.1. Overloading by differing types of parameter



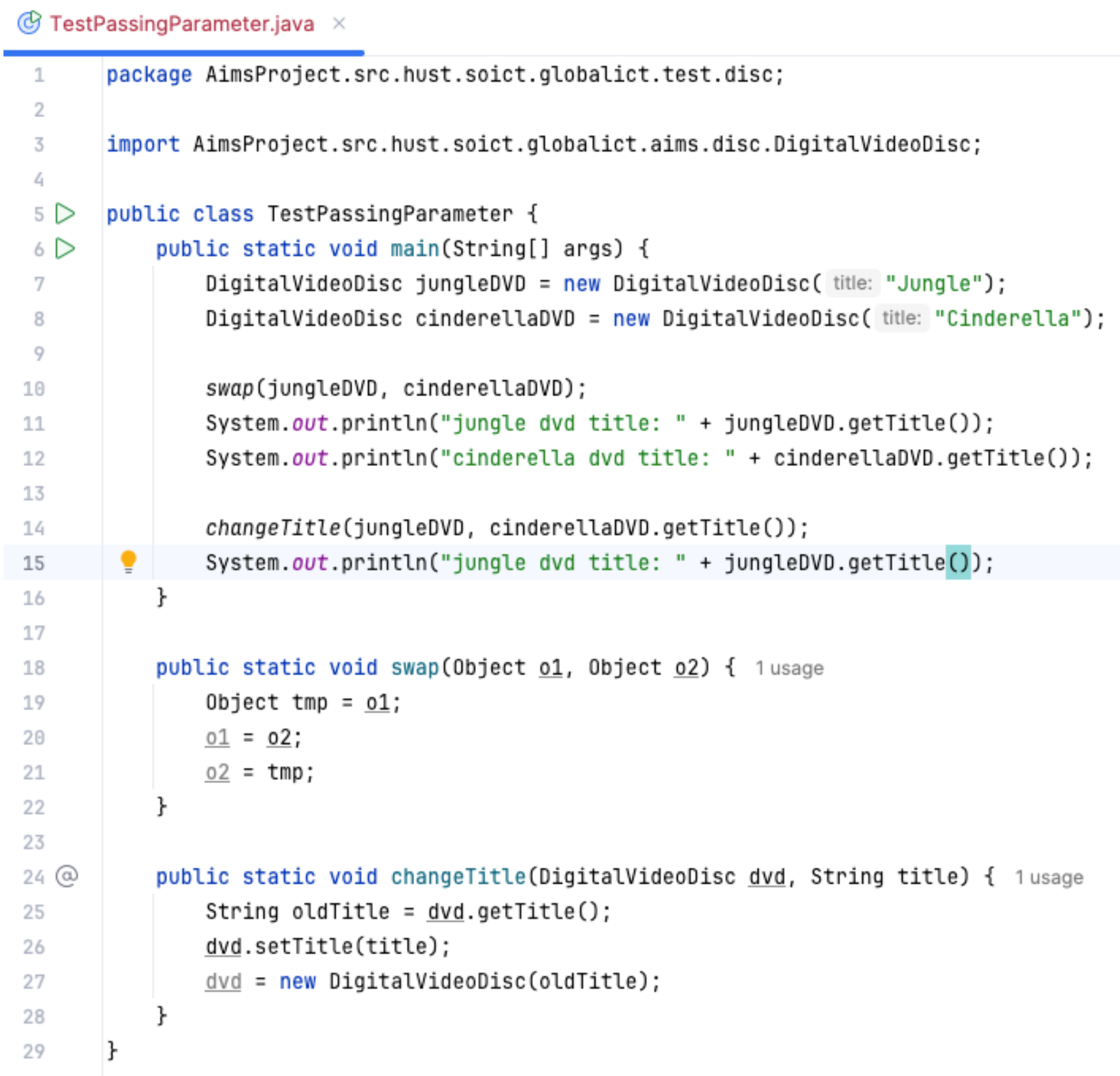
```
public void addDigitalVideoDisc(DigitalVideoDisc disc) {  6 usages  new *
    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = disc;
        qtyOrdered += 1;
        System.out.println("The disc has been added");
    } else System.out.println("The cart is almost full");
}

public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList) {  1 usage  new *
    if (dvdList.length > MAX_NUMBERS_ORDERED) {
        System.out.println("The cart is almost full");
    } else {
        for (int i = 0; i < dvdList.length; i++) {
            itemsOrdered[qtyOrdered] = dvdList[i];
            qtyOrdered +=1;
            System.out.println(dvdList[i].getTitle() + " has been added");
        }

    }
}
```

2.2. Overloading by differing the number of parameters



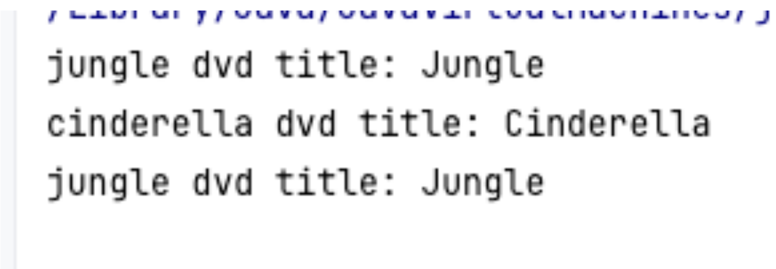
```
public void addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2) {
    DigitalVideoDisc [] dvdList = {dvd1, dvd2};
    addDigitalVideoDisc(dvdList);
}
```

3. Passing parameter



```java
package AimsProject.src.hust.soict.globalict.test.disc;

import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;

public class TestPassingParameter {
    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc( title: "Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc( title: "Cinderella");

        swap(jungleDVD, cinderellaDVD);
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
        System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
    }

    public static void swap(Object o1, Object o2) {  1 usage
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

    public static void changeTitle(DigitalVideoDisc dvd, String title) {  1 usage
        String oldTitle = dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
    }
}
```

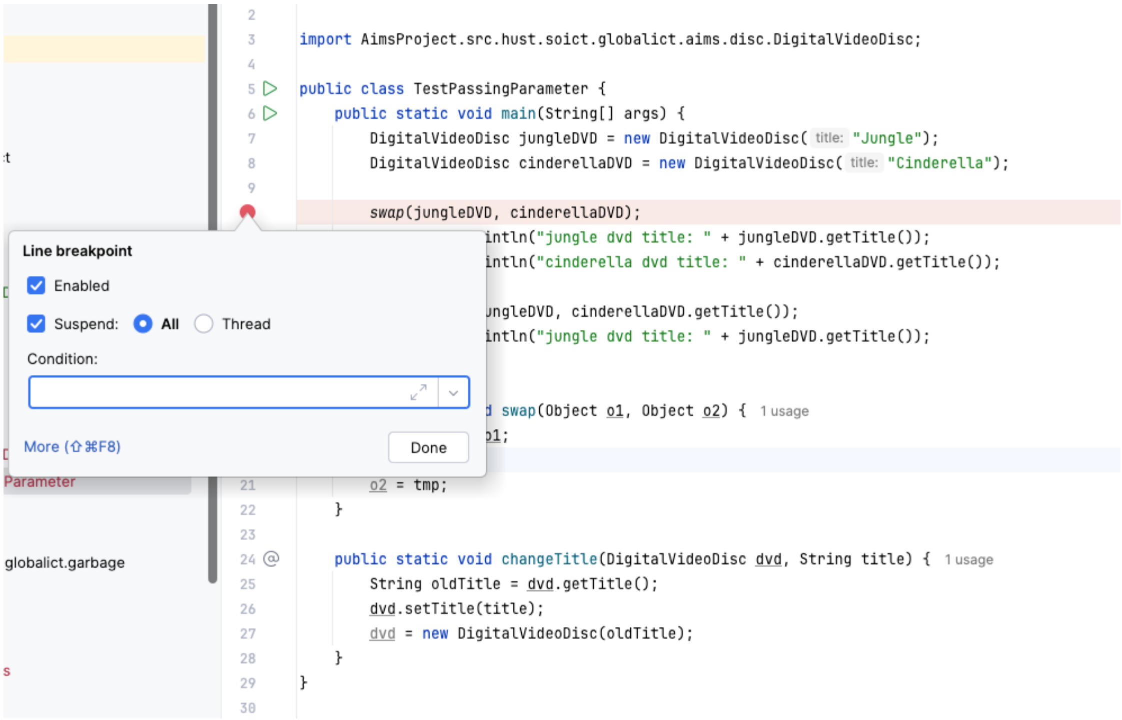
Result:



```
/Library/Java/JavaVirtualMachines/j
jungle dvd title: Jungle
cinderella dvd title: Cinderella
jungle dvd title: Jungle
```

4. Use debug run

4.1. Debug run for the swap method of TestPassingParameter



```
import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;

public class TestPassingParameter {
    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc( title: "Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc( title: "Cinderella");

        swap(jungleDVD, cinderellaDVD);
                        intln("jungle dvd title: " + jungleDVD.getTitle());
                        intln("cinderella dvd title: " + cinderellaDVD.getTitle());

                        ungleDVD, cinderellaDVD.getTitle());
                        intln("jungle dvd title: " + jungleDVD.getTitle());

                      d swap(Object o1, Object o2) {  1 usage
                        o1;

        o2 = tmp;
    }

    public static void changeTitle(DigitalVideoDisc dvd, String title) {  1 usage
        String oldTitle = dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
    }
}
```

Line breakpoint

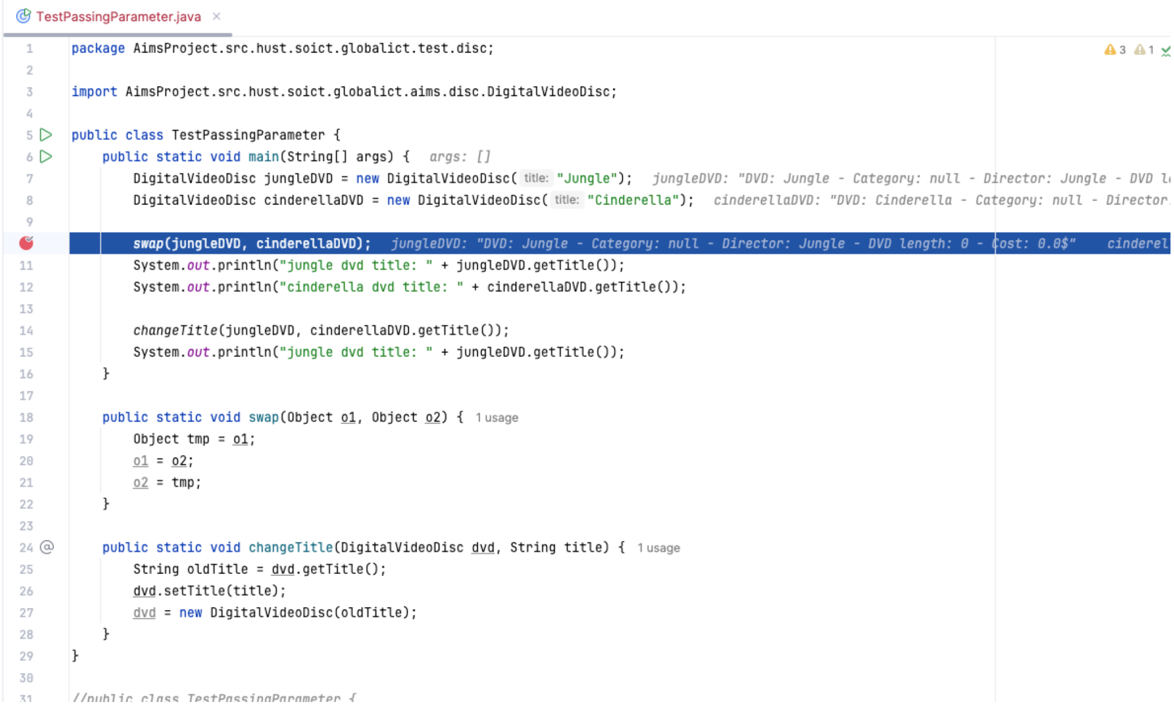
☑ Enabled

☑ Suspend: ● All ○ Thread

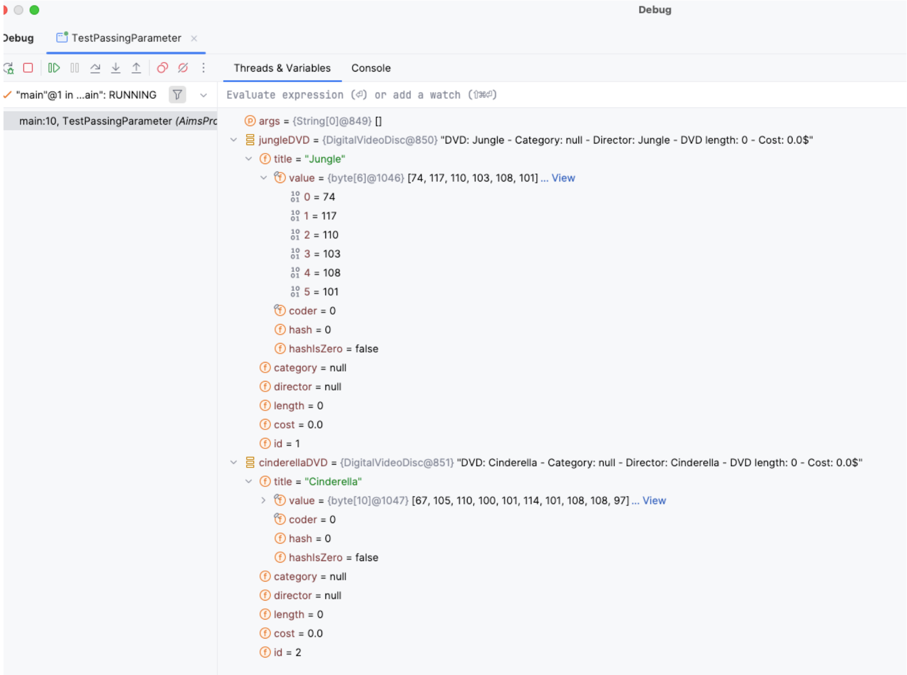
Condition:

More (⇧⌘F8) Done

4.2. Run in Debug mode:

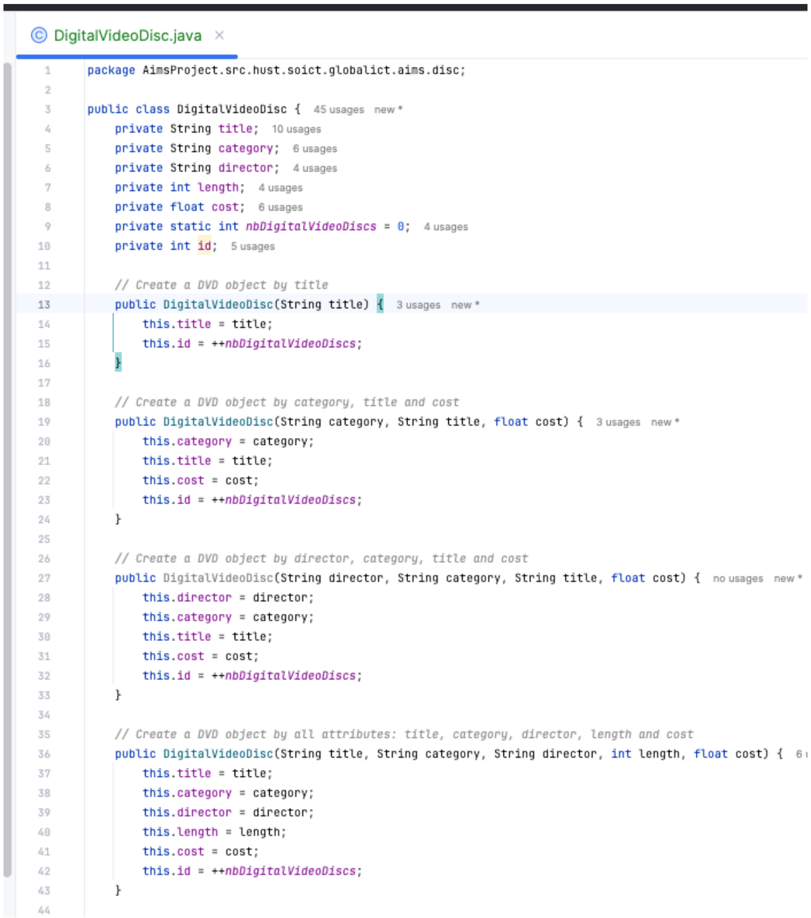


```
package AimsProject.src.hust.soict.globalict.test.disc;

import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;

public class TestPassingParameter {
    public static void main(String[] args) {   args: []
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc( title: "Jungle");   jungleDVD: "DVD: Jungle - Category: null - Director: Jungle - DVD l
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc( title: "Cinderella");   cinderellaDVD: "DVD: Cinderella - Category: null - Director

        swap(jungleDVD, cinderellaDVD);   jungleDVD: "DVD: Jungle - Category: null - Director: Jungle - DVD length: 0 - Cost: 0.0$"   cinderel
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
        System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
    }

    public static void swap(Object o1, Object o2) {  1 usage
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

    public static void changeTitle(DigitalVideoDisc dvd, String title) {  1 usage
        String oldTitle = dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
    }
}

//public class TestPassingParameter {
```



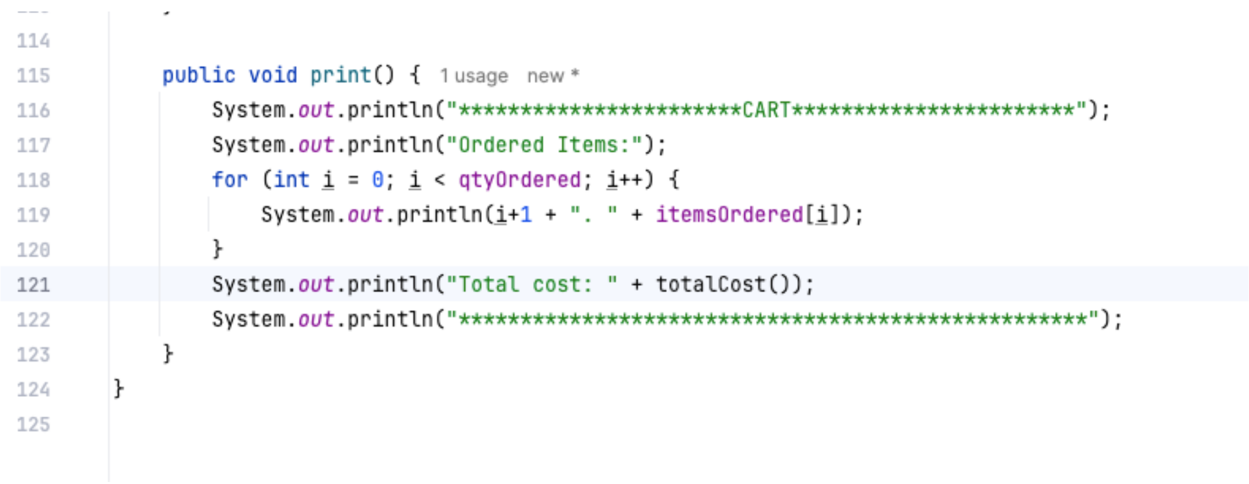
```
"main"@1 in ...ain": RUNNING
main:10, TestPassingParameter (AimsPro

Evaluate expression (⏎) or add a watch (⇧⌘⏎)

args = {String[0]@849} []
jungleDVD = {DigitalVideoDisc@850} "DVD: Jungle - Category: null - Director: Jungle - DVD length: 0 - Cost: 0.0$"
  title = "Jungle"
    value = {byte[6]@1046} [74, 117, 110, 103, 108, 101] ... View
      0 = 74
      1 = 117
      2 = 110
      3 = 103
      4 = 108
      5 = 101
    coder = 0
    hash = 0
    hashIsZero = false
  category = null
  director = null
  length = 0
  cost = 0.0
  id = 1
cinderellaDVD = {DigitalVideoDisc@851} "DVD: Cinderella - Category: null - Director: Cinderella - DVD length: 0 - Cost: 0.0$"
  title = "Cinderella"
    value = {byte[10]@1047} [67, 105, 110, 100, 101, 114, 101, 108, 108, 97] ... View
    coder = 0
    hash = 0
    hashIsZero = false
  category = null
  director = null
  length = 0
  cost = 0.0
  id = 2
```

5. Classifier Member and Instance Member

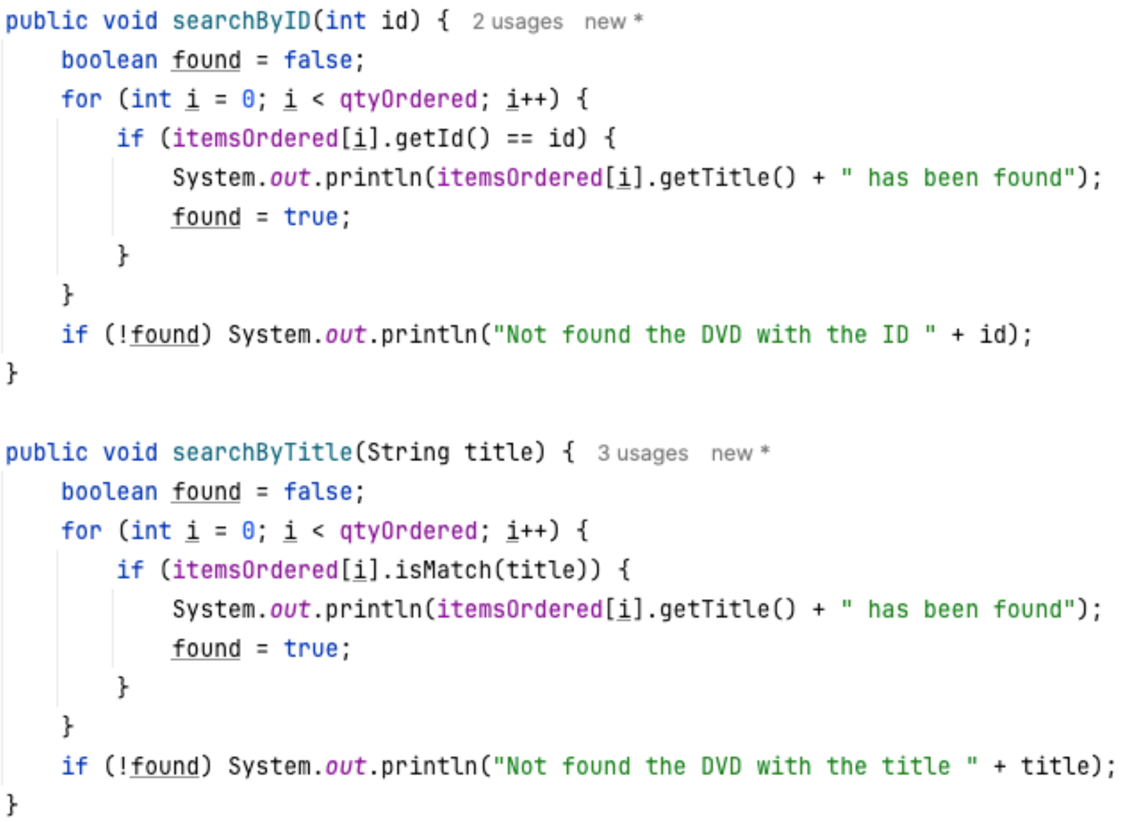


```java
package AimsProject.src.hust.soict.globalict.aims.disc;

public class DigitalVideoDisc {  45 usages  new *
    private String title;  10 usages
    private String category;  6 usages
    private String director;  4 usages
    private int length;  4 usages
    private float cost;  6 usages
    private static int nbDigitalVideoDiscs = 0;  4 usages
    private int id;  5 usages

    // Create a DVD object by title
    public DigitalVideoDisc(String title) {  3 usages  new *
        this.title = title;
        this.id = ++nbDigitalVideoDiscs;
    }

    // Create a DVD object by category, title and cost
    public DigitalVideoDisc(String category, String title, float cost) {  3 usages  new *
        this.category = category;
        this.title = title;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }

    // Create a DVD object by director, category, title and cost
    public DigitalVideoDisc(String director, String category, String title, float cost) {  no usages  new *
        this.director = director;
        this.category = category;
        this.title = title;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }

    // Create a DVD object by all attributes: title, category, director, length and cost
    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {  6 
        this.title = title;
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }

```

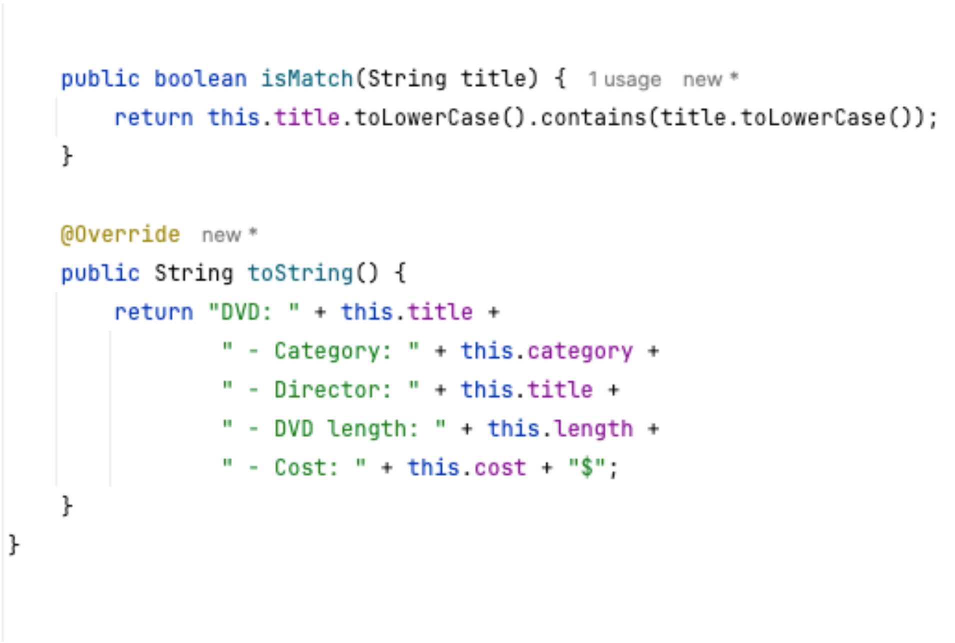
6. Open the Cart class



```
    public void print() {  1 usage   new *
        System.out.println("***********************CART***********************");
        System.out.println("Ordered Items:");
        for (int i = 0; i < qtyOrdered; i++) {
            System.out.println(i+1 + ". " + itemsOrdered[i]);
        }
        System.out.println("Total cost: " + totalCost());
        System.out.println("***************************************************");
    }
}
```

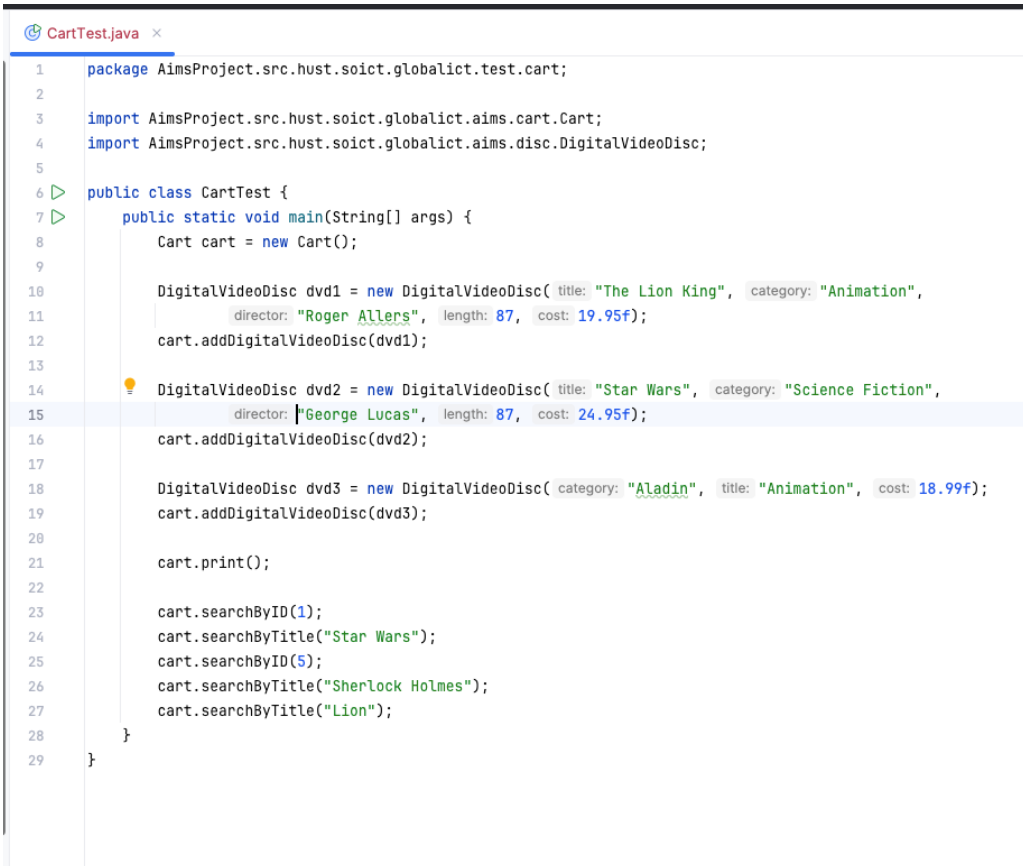


```
public void searchByID(int id) {  2 usages   new *
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].getId() == id) {
            System.out.println(itemsOrdered[i].getTitle() + " has been found");
            found = true;
        }
    }
    if (!found) System.out.println("Not found the DVD with the ID " + id);
}

public void searchByTitle(String title) {  3 usages   new *
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].isMatch(title)) {
            System.out.println(itemsOrdered[i].getTitle() + " has been found");
            found = true;
        }
    }
    if (!found) System.out.println("Not found the DVD with the title " + title);
}
```



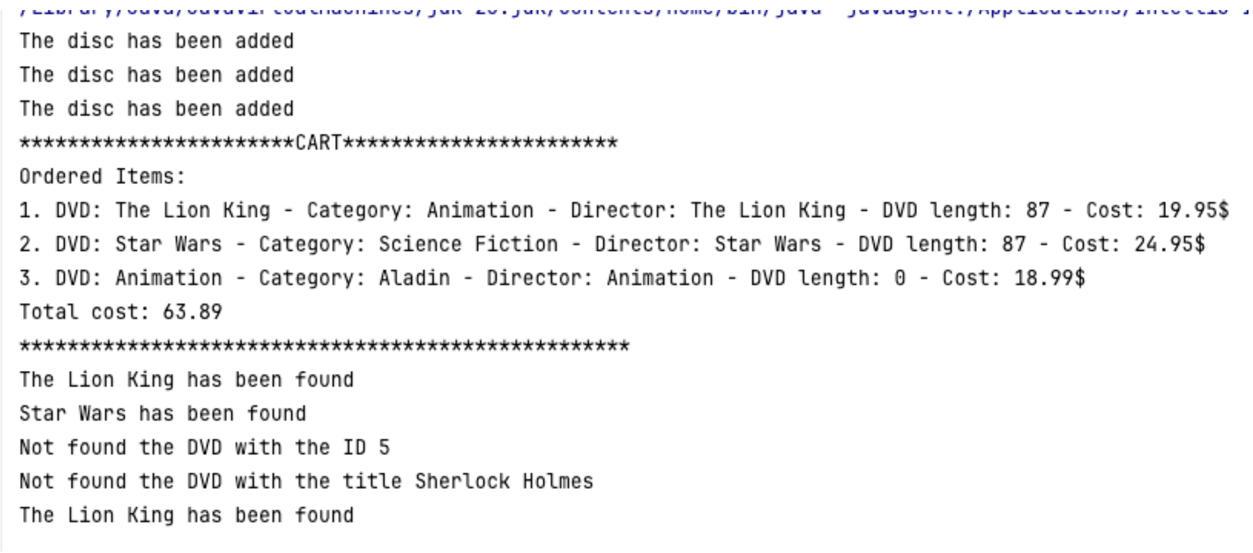
```
    public boolean isMatch(String title) {  1 usage   new *
        return this.title.toLowerCase().contains(title.toLowerCase());
    }

    @Override  new *
    public String toString() {
        return "DVD: " + this.title +
                " - Category: " + this.category +
                " - Director: " + this.title +
                " - DVD length: " + this.length +
                " - Cost: " + this.cost + "$";
    }
}
```

CartTest:



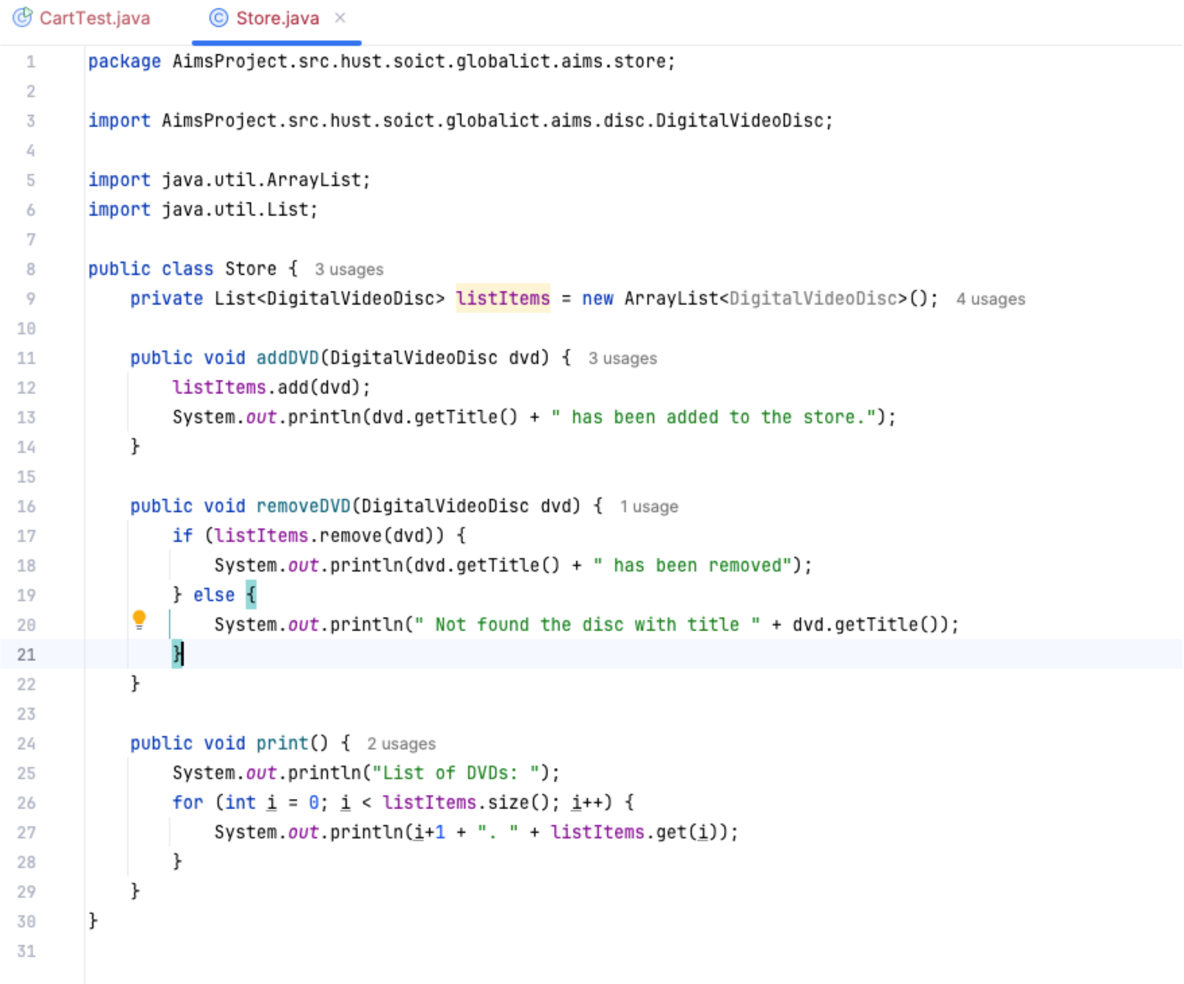
```
package AimsProject.src.hust.soict.globalict.test.cart;

import AimsProject.src.hust.soict.globalict.aims.cart.Cart;
import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;

public class CartTest {
    public static void main(String[] args) {
        Cart cart = new Cart();

        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King",  category: "Animation",
                 director: "Roger Allers",  length: 87,  cost: 19.95f);
        cart.addDigitalVideoDisc(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars",  category: "Science Fiction",
                 director: "George Lucas",  length: 87,  cost: 24.95f);
        cart.addDigitalVideoDisc(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc( category: "Aladin",  title: "Animation",  cost: 18.99f);
        cart.addDigitalVideoDisc(dvd3);

        cart.print();

        cart.searchByID(1);
        cart.searchByTitle("Star Wars");
        cart.searchByID(5);
        cart.searchByTitle("Sherlock Holmes");
        cart.searchByTitle("Lion");
    }
}
```

Result:

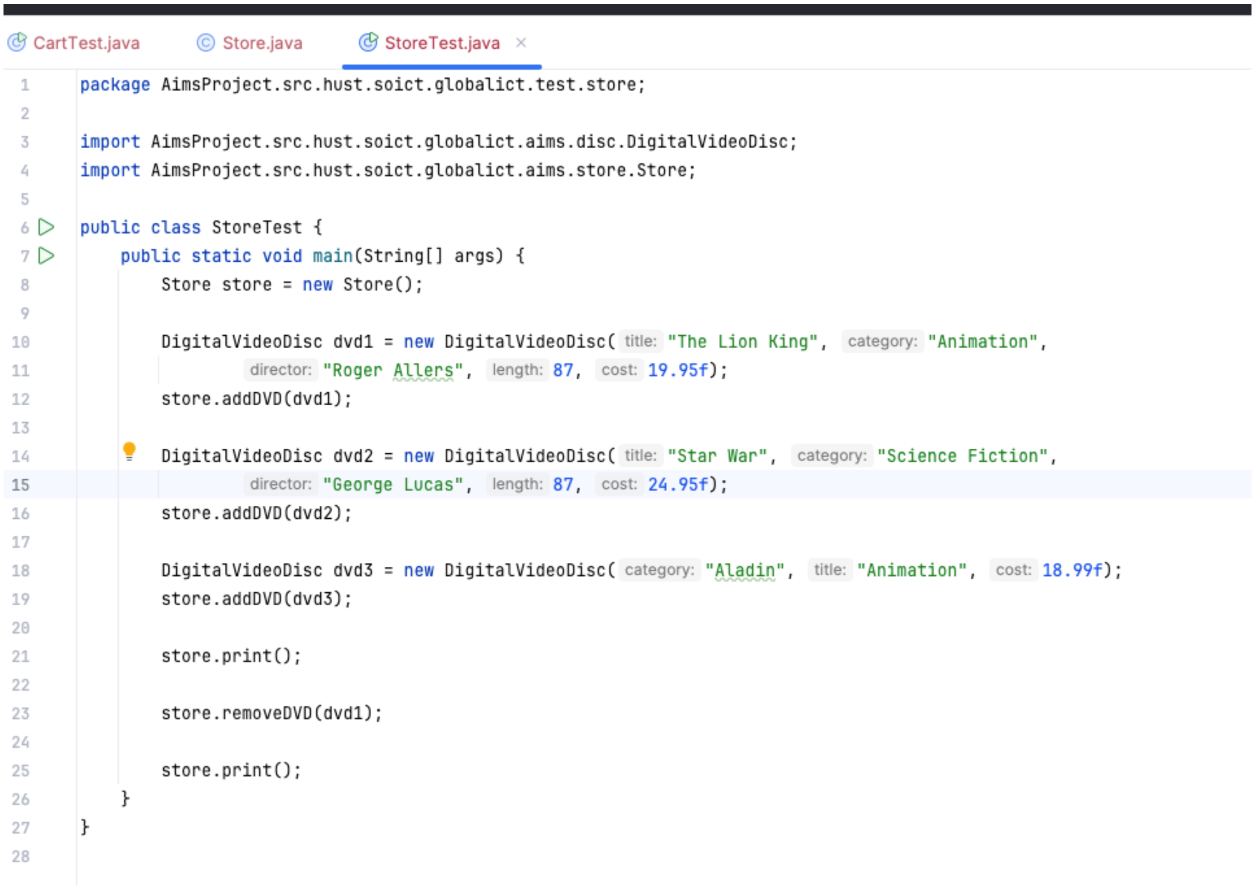


```
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ I
The disc has been added
The disc has been added
The disc has been added
***********************CART***********************
Ordered Items:
1. DVD: The Lion King - Category: Animation - Director: The Lion King - DVD length: 87 - Cost: 19.95$
2. DVD: Star Wars - Category: Science Fiction - Director: Star Wars - DVD length: 87 - Cost: 24.95$
3. DVD: Animation - Category: Aladin - Director: Animation - DVD length: 0 - Cost: 18.99$
Total cost: 63.89
***************************************************
The Lion King has been found
Star Wars has been found
Not found the DVD with the ID 5
Not found the DVD with the title Sherlock Holmes
The Lion King has been found
```

7. Implement the Store class

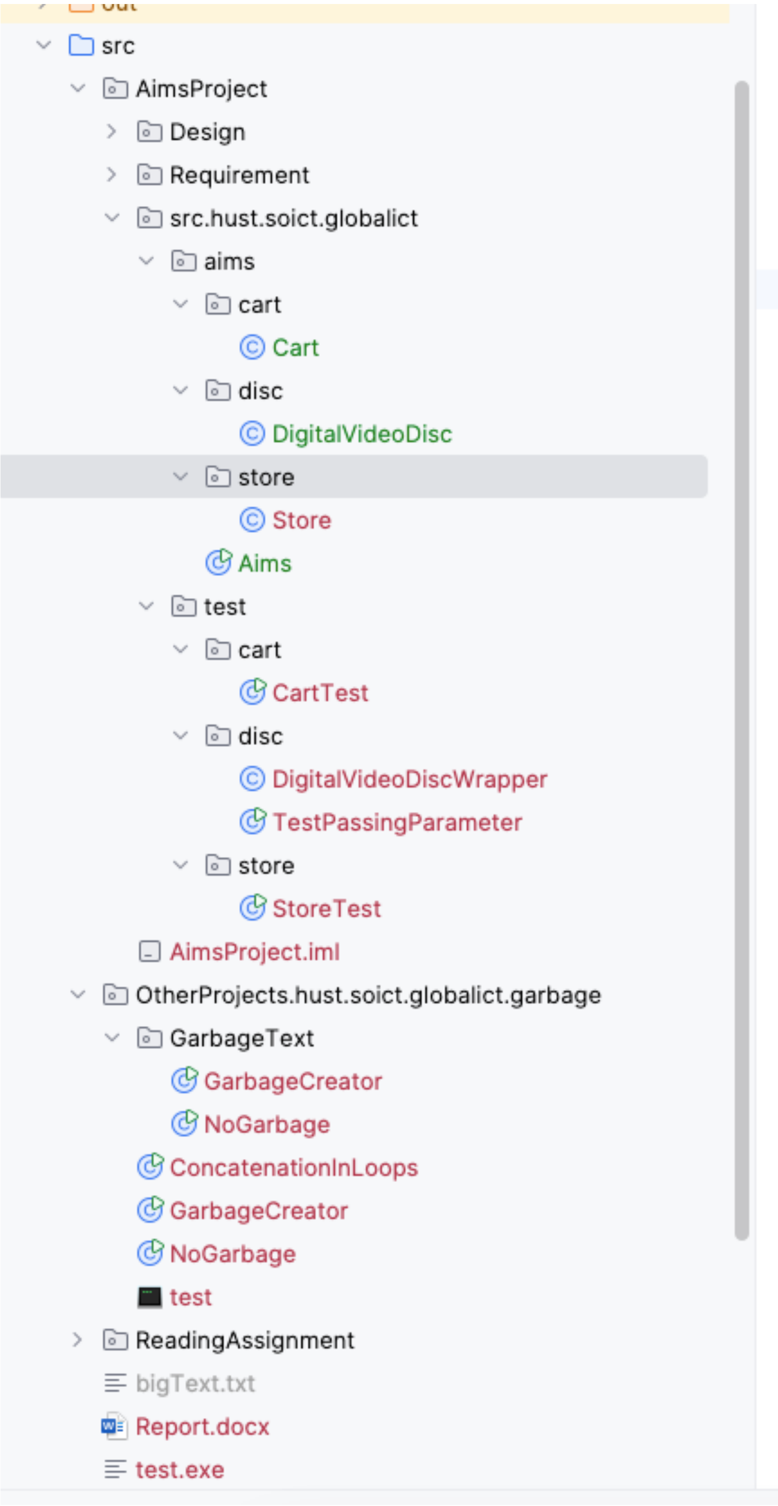


```java
package AimsProject.src.hust.soict.globalict.aims.store;

import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;

import java.util.ArrayList;
import java.util.List;

public class Store {  3 usages
    private List<DigitalVideoDisc> listItems = new ArrayList<DigitalVideoDisc>();  4 usages

    public void addDVD(DigitalVideoDisc dvd) {  3 usages
        listItems.add(dvd);
        System.out.println(dvd.getTitle() + " has been added to the store.");
    }

    public void removeDVD(DigitalVideoDisc dvd) {  1 usage
        if (listItems.remove(dvd)) {
            System.out.println(dvd.getTitle() + " has been removed");
        } else {
            System.out.println(" Not found the disc with title " + dvd.getTitle());
        }
    }

    public void print() {  2 usages
        System.out.println("List of DVDs: ");
        for (int i = 0; i < listItems.size(); i++) {
            System.out.println(i+1 + ". " + listItems.get(i));
        }
    }
}
```



```java
package AimsProject.src.hust.soict.globalict.test.store;

import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;
import AimsProject.src.hust.soict.globalict.aims.store.Store;

public class StoreTest {
    public static void main(String[] args) {
        Store store = new Store();

        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King",  category: "Animation",
                 director: "Roger Allers",  length: 87,  cost: 19.95f);
        store.addDVD(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star War",  category: "Science Fiction",
                 director: "George Lucas",  length: 87,  cost: 24.95f);
        store.addDVD(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc( category: "Aladin",  title: "Animation",  cost: 18.99f);
        store.addDVD(dvd3);

        store.print();

        store.removeDVD(dvd1);

        store.print();
    }
}
```

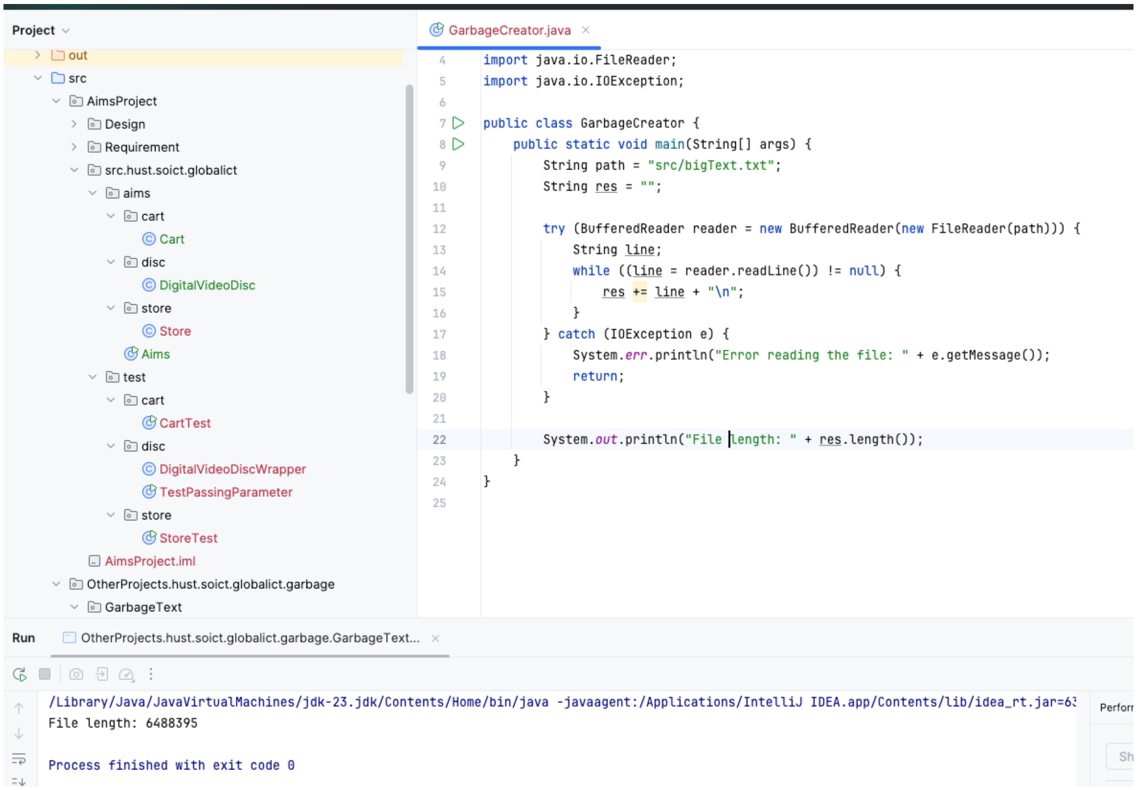
8. Re-organize your projects



```
out
src
  AimsProject
    Design
    Requirement
    src.hust.soict.globalict
      aims
        cart
          Cart
        disc
          DigitalVideoDisc
        store
          Store
        Aims
      test
        cart
          CartTest
        disc
          DigitalVideoDiscWrapper
          TestPassingParameter
        store
          StoreTest
    AimsProject.iml
  OtherProjects.hust.soict.globalict.garbage
    GarbageText
      GarbageCreator
      NoGarbage
    ConcatenationInLoops
    GarbageCreator
    NoGarbage
    test
  ReadingAssignment
  bigText.txt
  Report.docx
  test.exe
```

9. String, StringBuilder and StringBuffer

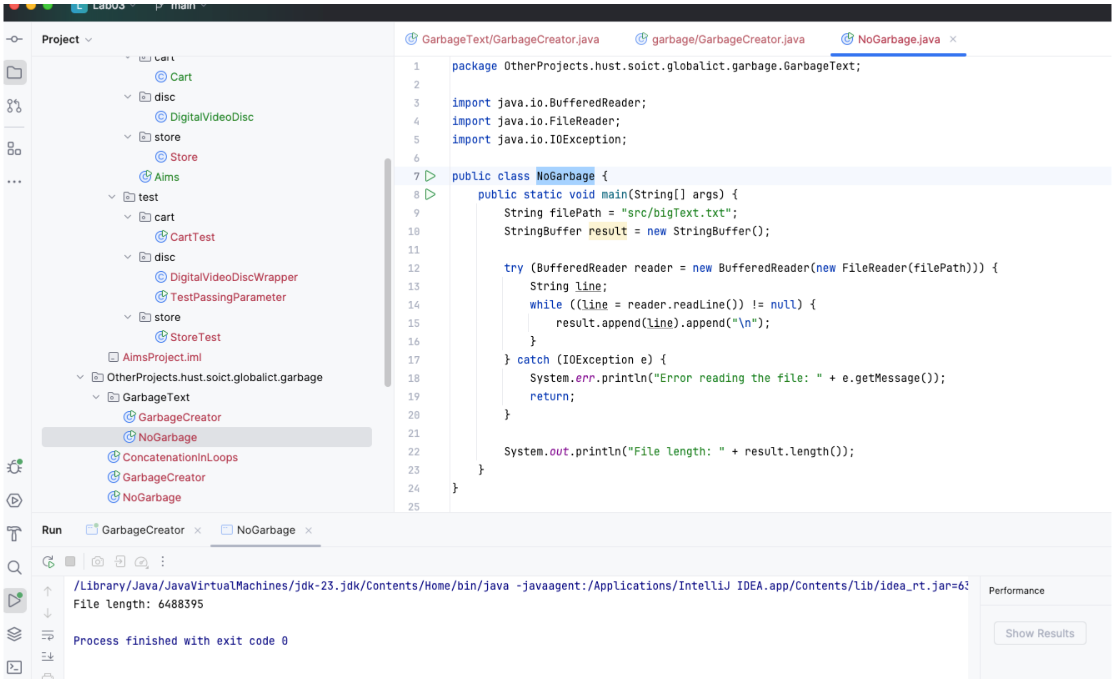
GarbageCreator:



```java
import java.io.FileReader;
import java.io.IOException;

public class GarbageCreator {
    public static void main(String[] args) {
        String path = "src/bigText.txt";
        String res = "";

        try (BufferedReader reader = new BufferedReader(new FileReader(path))) {
            String line;
            while ((line = reader.readLine()) != null) {
                res += line + "\n";
            }
        } catch (IOException e) {
            System.err.println("Error reading the file: " + e.getMessage());
            return;
        }

        System.out.println("File length: " + res.length());
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-23.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=63
File length: 6488395

Process finished with exit code 0
```

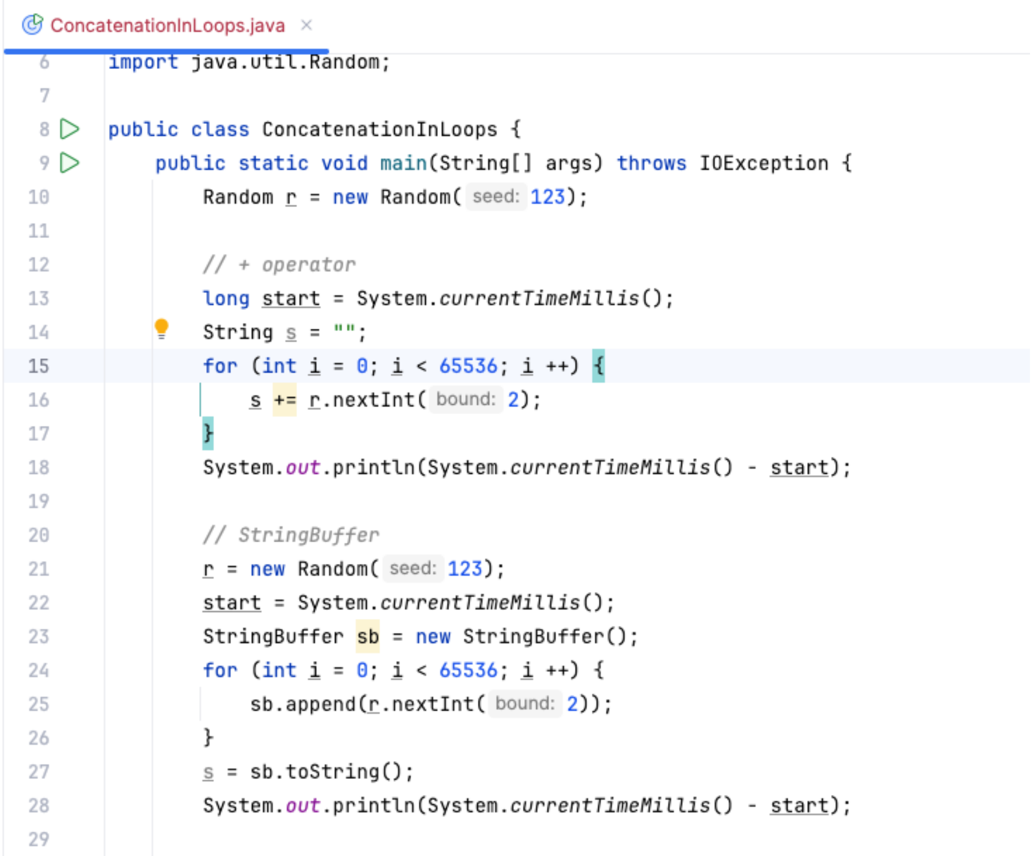
NoGarbage:



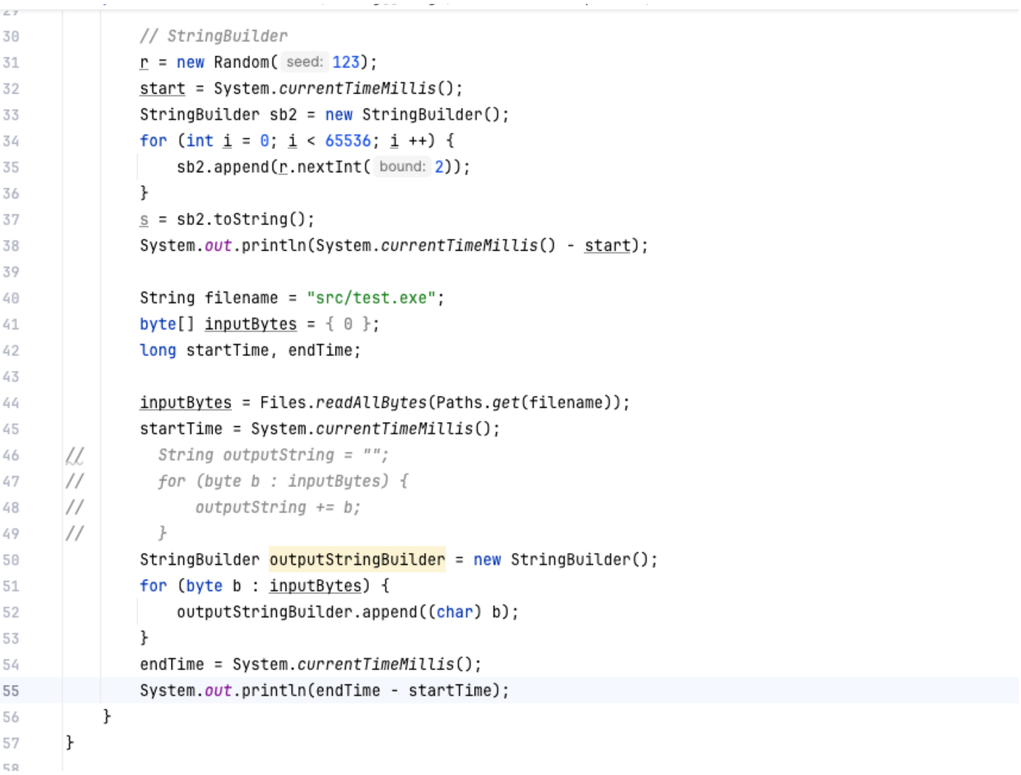
```java
package OtherProjects.hust.soict.globalict.garbage.GarbageText;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class NoGarbage {
    public static void main(String[] args) {
        String filePath = "src/bigText.txt";
        StringBuffer result = new StringBuffer();

        try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
            String line;
            while ((line = reader.readLine()) != null) {
                result.append(line).append("\n");
            }
        } catch (IOException e) {
            System.err.println("Error reading the file: " + e.getMessage());
            return;
        }

        System.out.println("File length: " + result.length());
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-23.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=63
File length: 6488395

Process finished with exit code 0
```

ConcatenationInLoops:

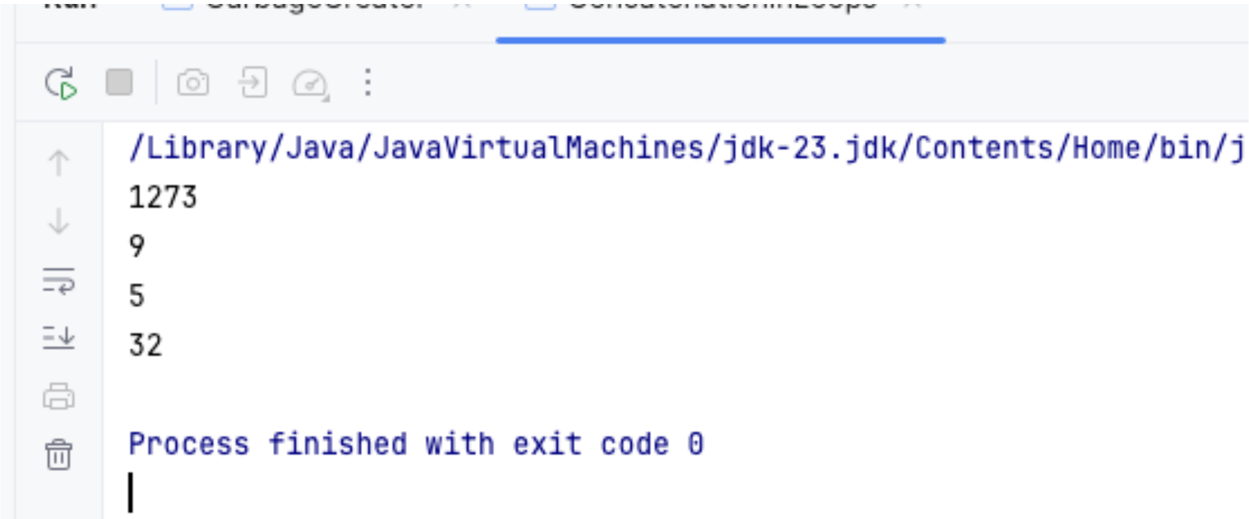


```java
import java.util.Random;

public class ConcatenationInLoops {
    public static void main(String[] args) throws IOException {
        Random r = new Random( seed: 123);

        // + operator
        long start = System.currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i ++) {
            s += r.nextInt( bound: 2);
        }
        System.out.println(System.currentTimeMillis() - start);

        // StringBuffer
        r = new Random( seed: 123);
        start = System.currentTimeMillis();
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < 65536; i ++) {
            sb.append(r.nextInt( bound: 2));
        }
        s = sb.toString();
        System.out.println(System.currentTimeMillis() - start);

```



```java
        // StringBuilder
        r = new Random( seed: 123);
        start = System.currentTimeMillis();
        StringBuilder sb2 = new StringBuilder();
        for (int i = 0; i < 65536; i ++) {
            sb2.append(r.nextInt( bound: 2));
        }
        s = sb2.toString();
        System.out.println(System.currentTimeMillis() - start);

        String filename = "src/test.exe";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        inputBytes = Files.readAllBytes(Paths.get(filename));
        startTime = System.currentTimeMillis();
//        String outputString = "";
//        for (byte b : inputBytes) {
//            outputString += b;
//        }
        StringBuilder outputStringBuilder = new StringBuilder();
        for (byte b : inputBytes) {
            outputStringBuilder.append((char) b);
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

Result:



```
/Library/Java/JavaVirtualMachines/jdk-23.jdk/Contents/Home/bin/j
1273
9
5
32

Process finished with exit code 0
```