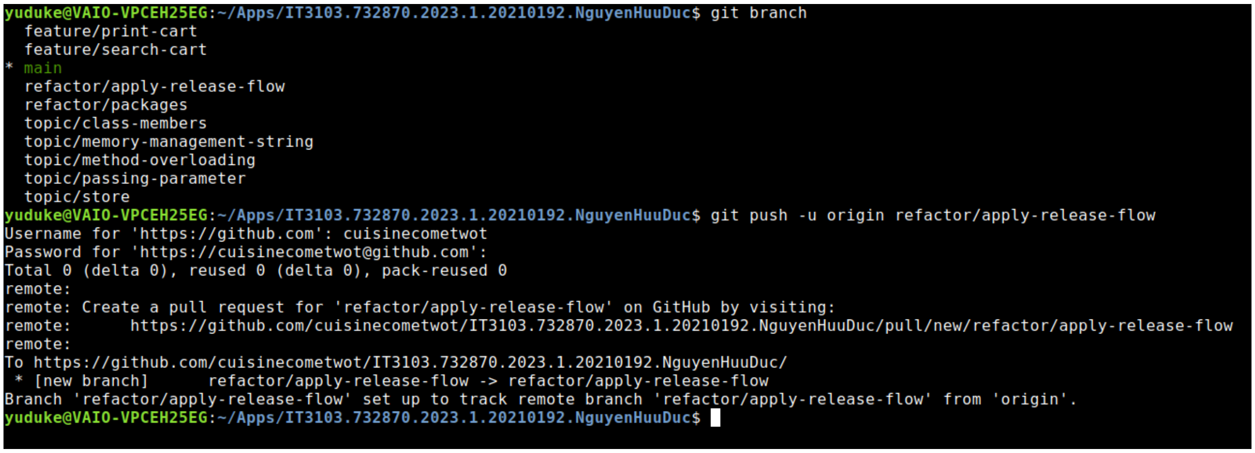
BÁO CÁO THỰC HÀNH LAB 3
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
LỚP: 143577 (LT) – 732870 (TN)

Table of Contents

1. Branch your repository



```
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc$ git branch
  feature/print-cart
  feature/search-cart
* main
  refactor/apply-release-flow
  refactor/packages
  topic/class-members
  topic/memory-management-string
  topic/method-overloading
  topic/passing-parameter
  topic/store
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc$ git push -u origin refactor/apply-release-flow
Username for 'https://github.com': cuisinecometwot
Password for 'https://cuisinecometwot@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'refactor/apply-release-flow' on GitHub by visiting:
remote:      https://github.com/cuisinecometwot/IT3103.732870.2023.1.20210192.NguyenHuuDuc/pull/new/refactor/apply-release-flow
remote:
To https://github.com/cuisinecometwot/IT3103.732870.2023.1.20210192.NguyenHuuDuc/
 * [new branch]      refactor/apply-release-flow -> refactor/apply-release-flow
Branch 'refactor/apply-release-flow' set up to track remote branch 'refactor/apply-release-flow' from 'origin'.
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc$ 
```

Typical steps for a new branch:

- Create and switch to a new branch in the local repo: git checkout -b ~

- Make changes in the local repo

- Commit the change in the local repo: git commit -m “Message”

- Create a new branch in the remote repo

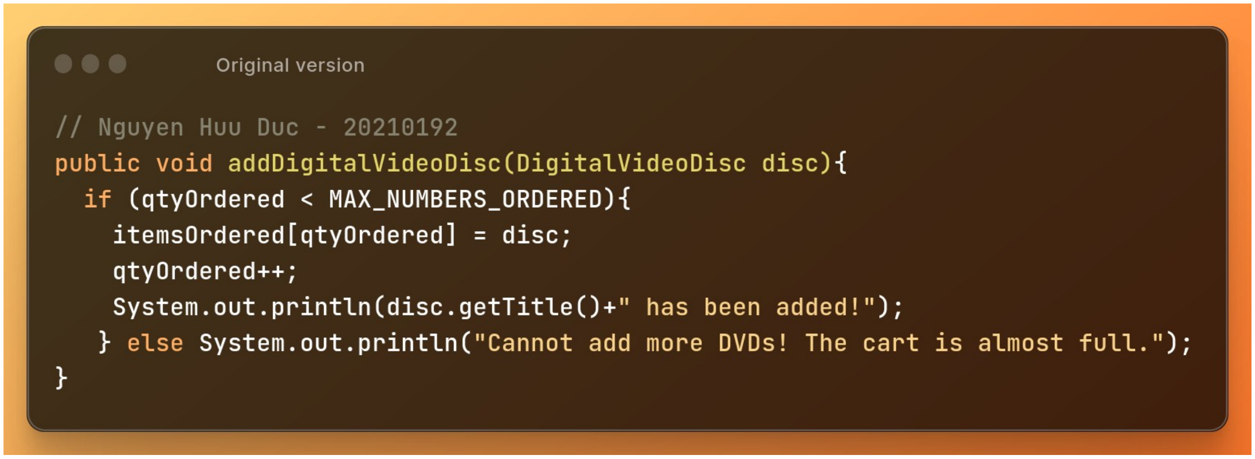
- Push the local branch to the remote branch: git push

- Merge the remote branch to the master branch

2. Working with Method Overloading

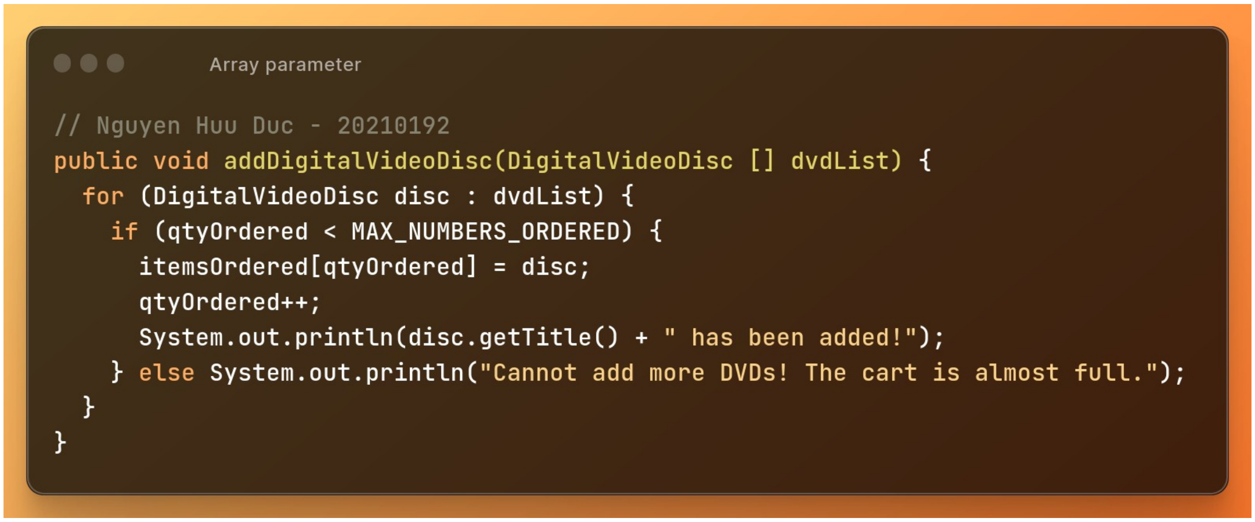
2.1 Overloading by differing types of parameters

- The current method has one input parameter of class DigitalVideoDisc



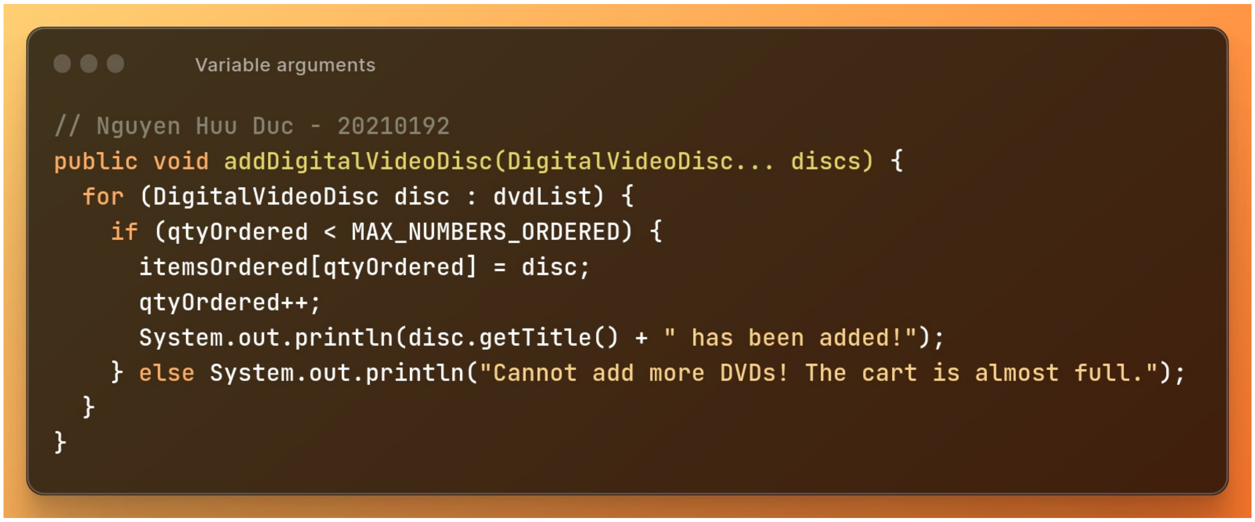
```java
// Nguyen Huu Duc - 20210192
public void addDigitalVideoDisc(DigitalVideoDisc disc){
  if (qtyOrdered < MAX_NUMBERS_ORDERED){
    itemsOrdered[qtyOrdered] = disc;
    qtyOrdered++;
    System.out.println(disc.getTitle()+" has been added!");
   } else System.out.println("Cannot add more DVDs! The cart is almost full.");
}
```

- A new method that has the same name but with array type parameter



```java
// Nguyen Huu Duc - 20210192
public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList) {
  for (DigitalVideoDisc disc : dvdList) {
    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
      itemsOrdered[qtyOrdered] = disc;
      qtyOrdered++;
      System.out.println(disc.getTitle() + " has been added!");
    } else System.out.println("Cannot add more DVDs! The cart is almost full.");
  }
}
```

- A new method which allows to pass an arbitrary number of DVDs

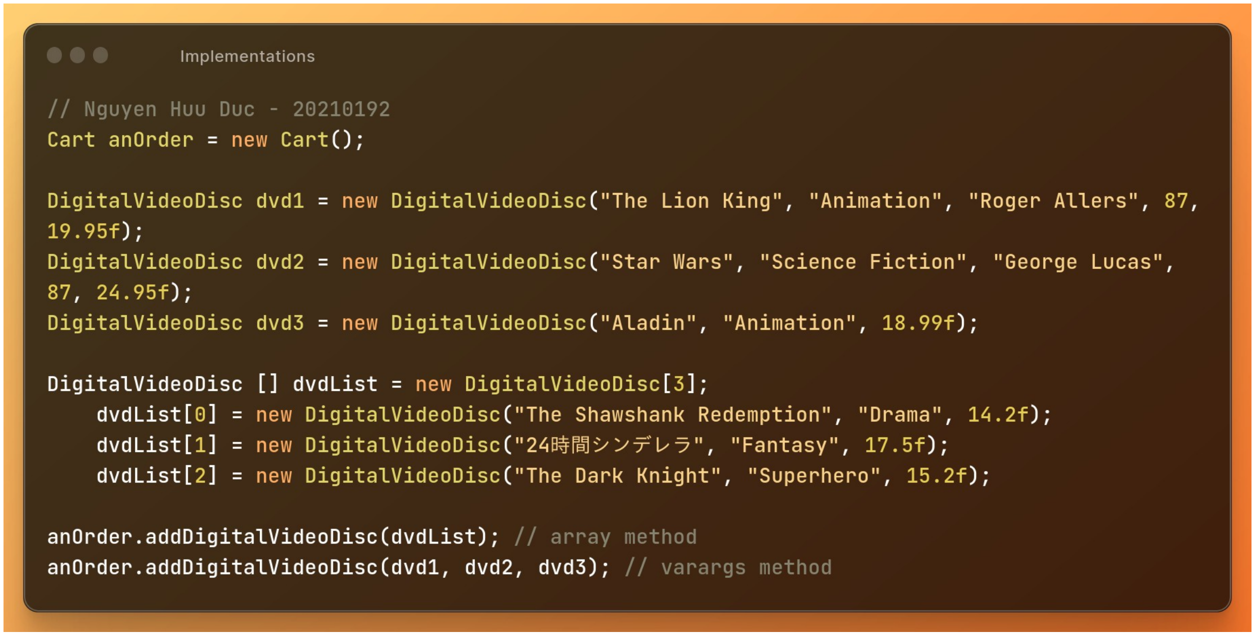


```java
// Nguyen Huu Duc - 20210192
public void addDigitalVideoDisc(DigitalVideoDisc... discs) {
  for (DigitalVideoDisc disc : dvdList) {
    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
      itemsOrdered[qtyOrdered] = disc;
      qtyOrdered++;
      System.out.println(disc.getTitle() + " has been added!");
    } else System.out.println("Cannot add more DVDs! The cart is almost full.");
  }
}
```

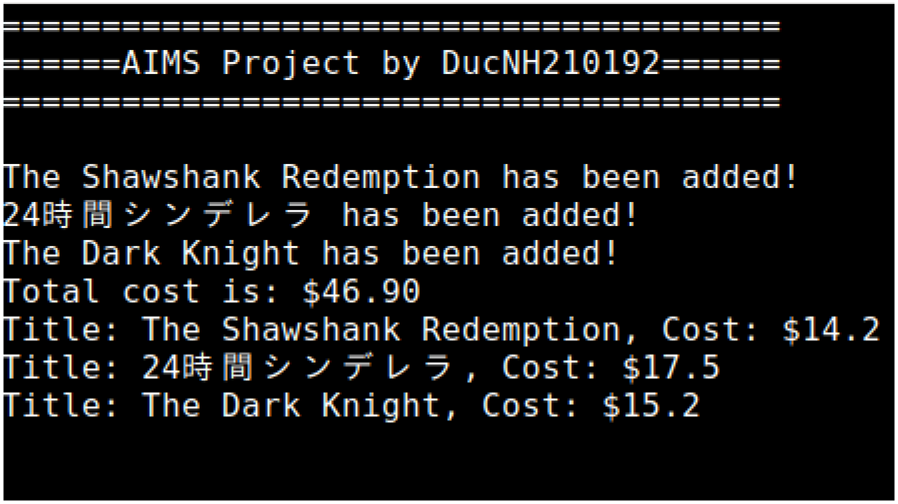
- Implementations of new methods:

These methods will add a list of DVDs to the current cart.

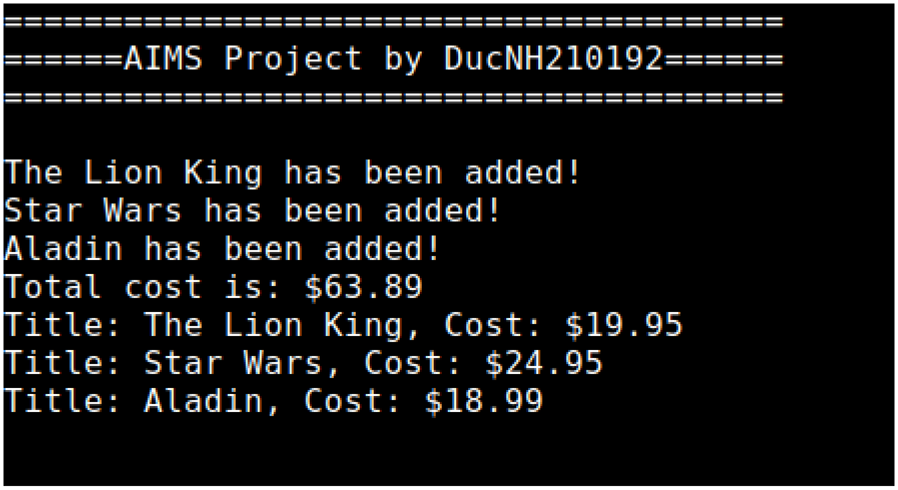
Please note that we cannot compile both because they have the same name and signature (one list as a parameter).



```java
// Nguyen Huu Duc - 20210192
Cart anOrder = new Cart();

DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87,
19.95f);
DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas",
87, 24.95f);
DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);

DigitalVideoDisc [] dvdList = new DigitalVideoDisc[3];
    dvdList[0] = new DigitalVideoDisc("The Shawshank Redemption", "Drama", 14.2f);
    dvdList[1] = new DigitalVideoDisc("24時間シンデレラ", "Fantasy", 17.5f);
    dvdList[2] = new DigitalVideoDisc("The Dark Knight", "Superhero", 15.2f);

anOrder.addDigitalVideoDisc(dvdList); // array method
anOrder.addDigitalVideoDisc(dvd1, dvd2, dvd3); // varargs method
```



```
=======================================
======AIMS Project by DucNH210192======
=======================================

The Shawshank Redemption has been added!
24時間シンデレラ has been added!
The Dark Knight has been added!
Total cost is: $46.90
Title: The Shawshank Redemption, Cost: $14.2
Title: 24時間シンデレラ, Cost: $17.5
Title: The Dark Knight, Cost: $15.2
```

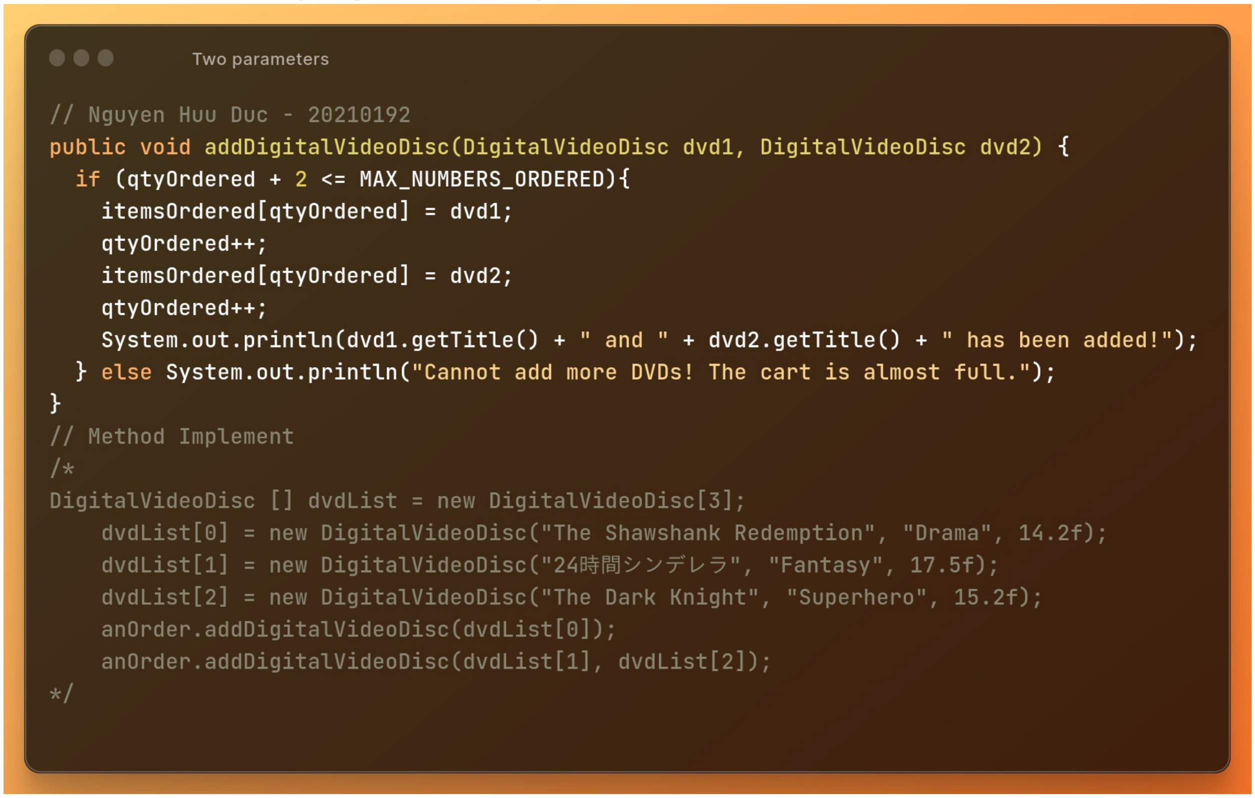


```
=======================================
======AIMS Project by DucNH210192======
=======================================

The Lion King has been added!
Star Wars has been added!
Aladin has been added!
Total cost is: $63.89
Title: The Lion King, Cost: $19.95
Title: Star Wars, Cost: $24.95
Title: Aladin, Cost: $18.99
```

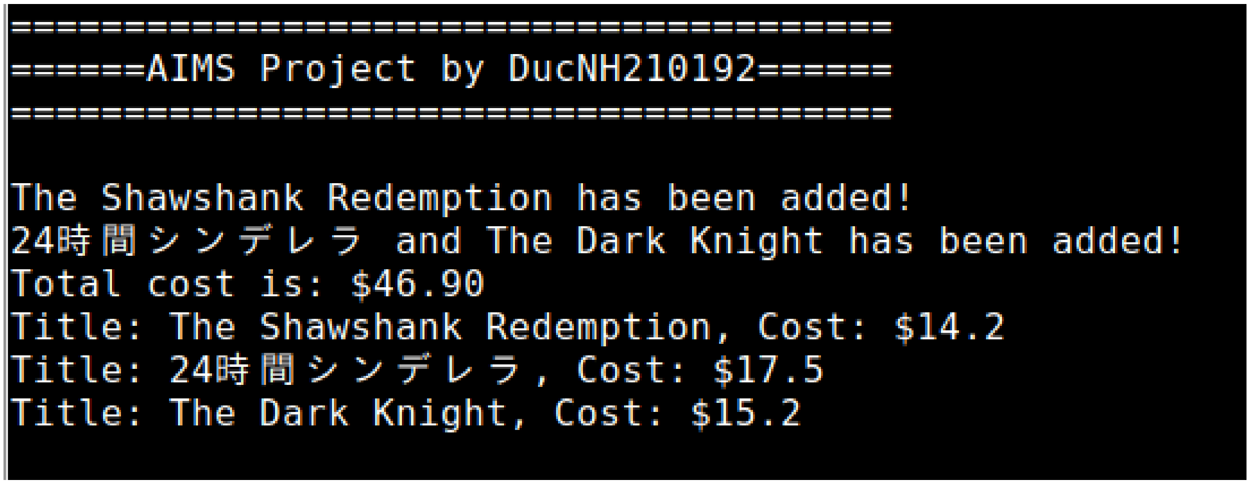
- Which method do you prefer in this case?

Since we can compile only one method, I prefer array parameter method. It is more readable, convenient to make change and easier to understand.

2.2 Overloading by differing the number of parameters

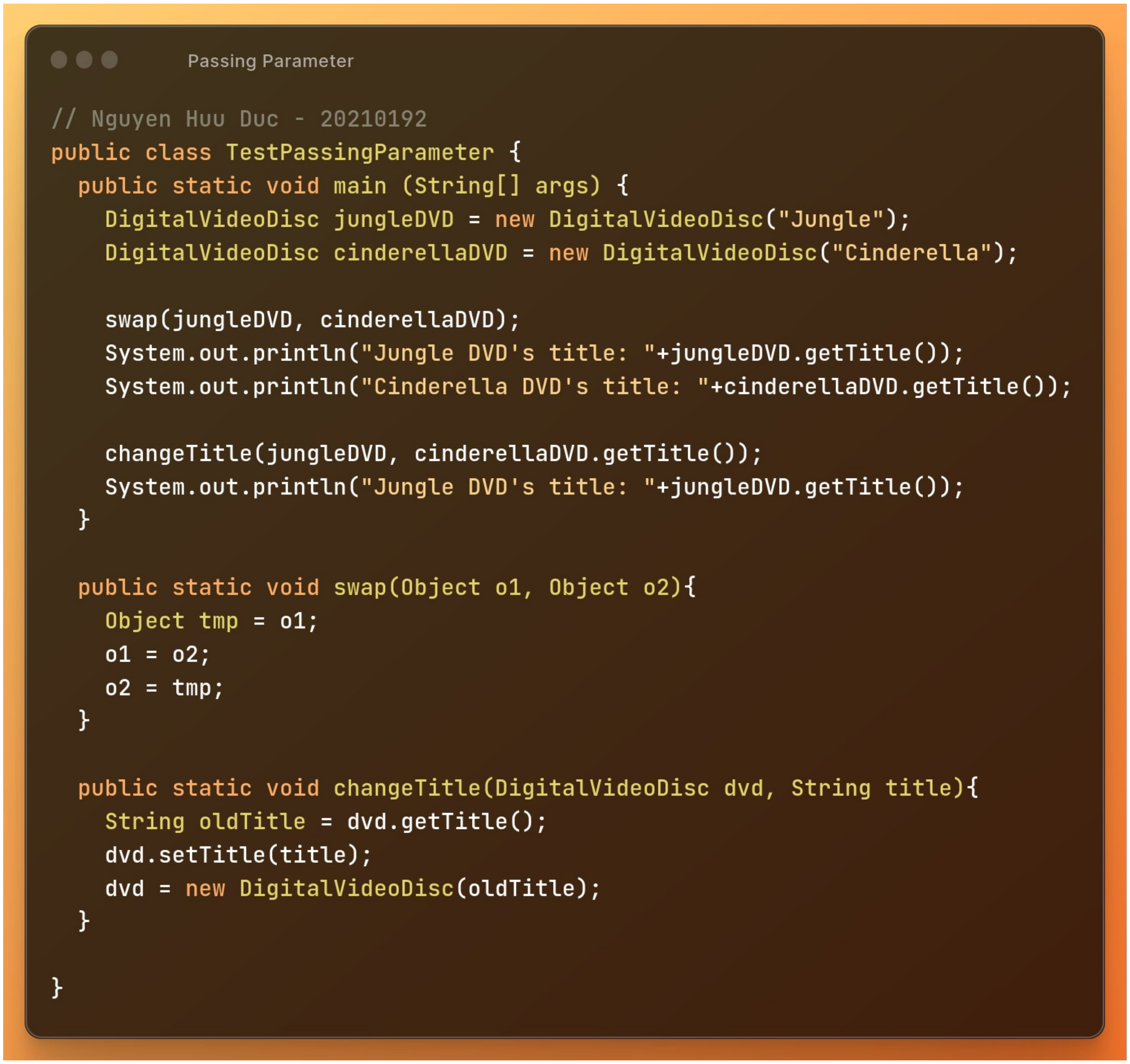


```java
// Nguyen Huu Duc - 20210192
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
  if (qtyOrdered + 2 <= MAX_NUMBERS_ORDERED){
    itemsOrdered[qtyOrdered] = dvd1;
    qtyOrdered++;
    itemsOrdered[qtyOrdered] = dvd2;
    qtyOrdered++;
    System.out.println(dvd1.getTitle() + " and " + dvd2.getTitle() + " has been added!");
  } else System.out.println("Cannot add more DVDs! The cart is almost full.");
}
// Method Implement
/*
DigitalVideoDisc [] dvdList = new DigitalVideoDisc[3];
    dvdList[0] = new DigitalVideoDisc("The Shawshank Redemption", "Drama", 14.2f);
    dvdList[1] = new DigitalVideoDisc("24時間シンデレラ", "Fantasy", 17.5f);
    dvdList[2] = new DigitalVideoDisc("The Dark Knight", "Superhero", 15.2f);
    anOrder.addDigitalVideoDisc(dvdList[0]);
    anOrder.addDigitalVideoDisc(dvdList[1], dvdList[2]);
*/
```

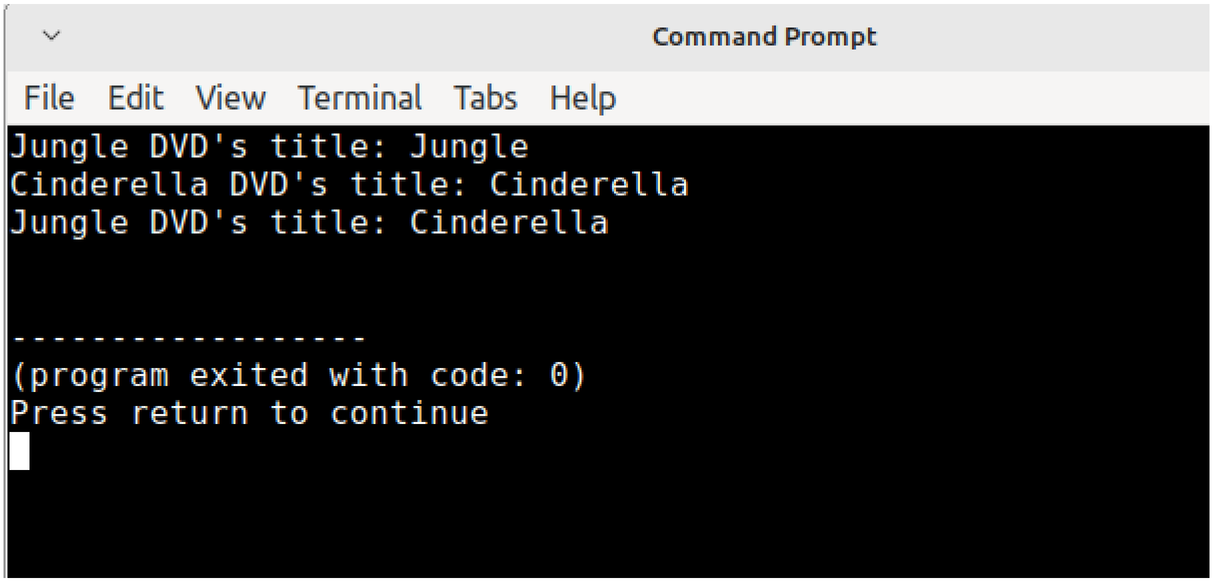


```
=======================================
======AIMS Project by DucNH210192======
=======================================

The Shawshank Redemption has been added!
24時間シンデレラ and The Dark Knight has been added!
Total cost is: $46.90
Title: The Shawshank Redemption, Cost: $14.2
Title: 24時間シンデレラ, Cost: $17.5
Title: The Dark Knight, Cost: $15.2
```

3. Passing parameter



```
// Nguyen Huu Duc - 20210192
public class TestPassingParameter {
  public static void main (String[] args) {
    DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
    DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");

    swap(jungleDVD, cinderellaDVD);
    System.out.println("Jungle DVD's title: "+jungleDVD.getTitle());
    System.out.println("Cinderella DVD's title: "+cinderellaDVD.getTitle());

    changeTitle(jungleDVD, cinderellaDVD.getTitle());
    System.out.println("Jungle DVD's title: "+jungleDVD.getTitle());
  }

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

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

}
```



```
Jungle DVD's title: Jungle
Cinderella DVD's title: Cinderella
Jungle DVD's title: Cinderella


------------------
(program exited with code: 0)
Press return to continue
```

- After the call of swap(), why does the title of these two objects still remain?

After the call of swap(), the title of these two objects still remains because the swap() method only swaps the references to the objects, not the objects themselves. This means that after the call to swap(), o1 and o2 still refer to the same objects as they did before the call.

- After the call of changeTitle(), why is the title of the JungleDVD changed?

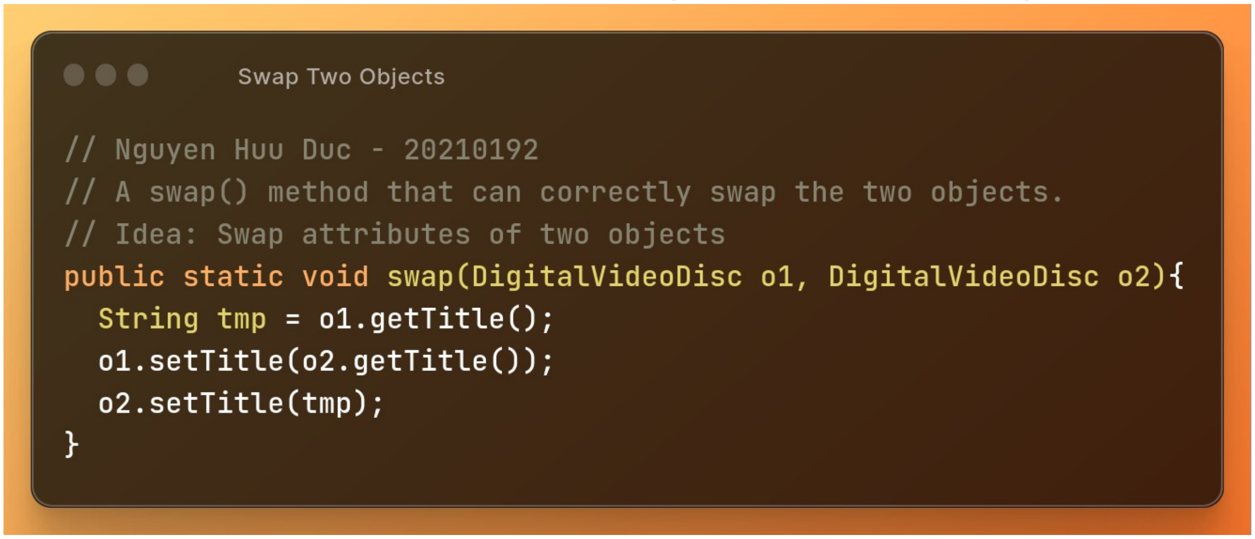
After the call of changeTitle(), the title of the DVD is changed because the setTitle(title) method changes the title directly.

The dvd = new DigitalVideoDisc(oldTitle) creates a new DVD object with the old title. It does not affect the original dvd object as the reference variable no longer points to the original object.

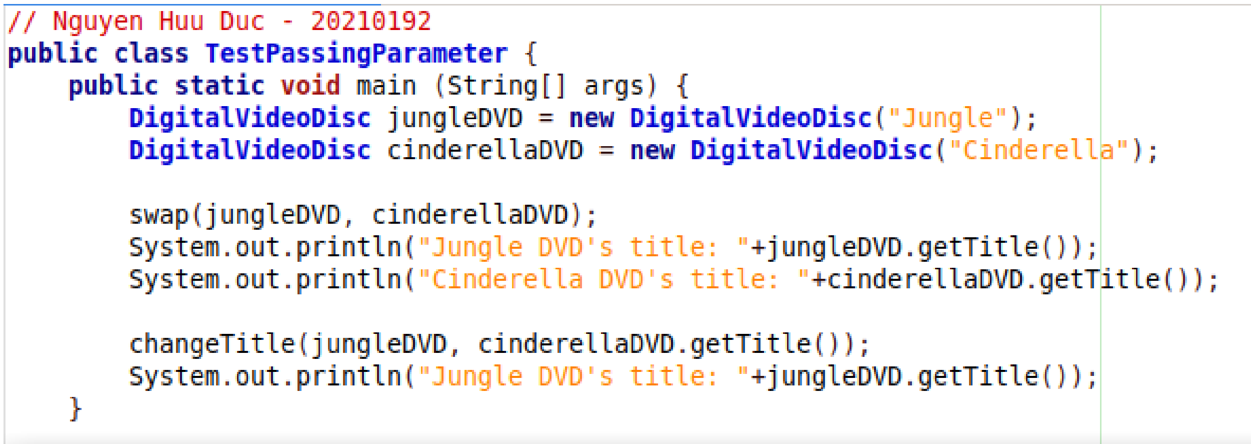
- Is Java a Pass-by-value or Pass-by-reference?

In conclusion, Java is always a pass-by-value programming language.

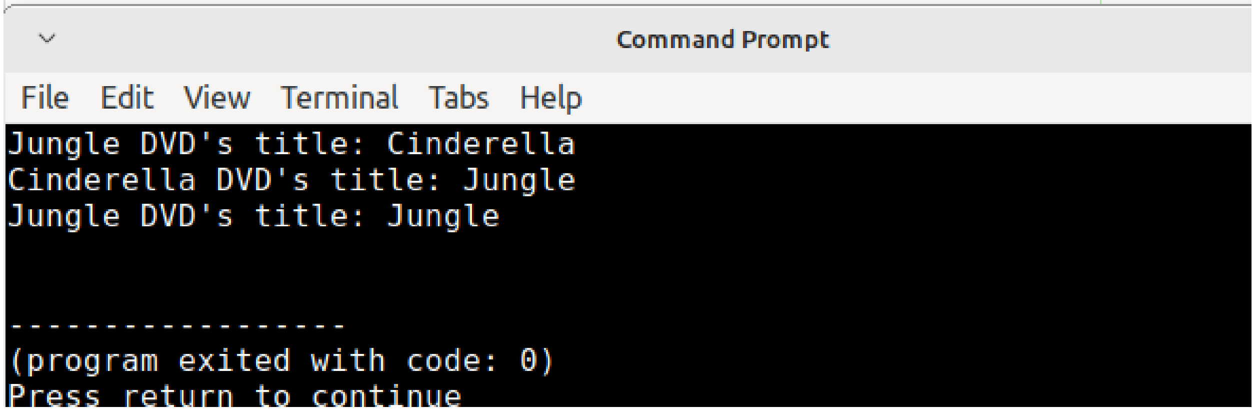
- A swap method that can correctly swap the two objects



```java
// Nguyen Huu Duc - 20210192
// A swap() method that can correctly swap the two objects.
// Idea: Swap attributes of two objects
public static void swap(DigitalVideoDisc o1, DigitalVideoDisc o2){
  String tmp = o1.getTitle();
  o1.setTitle(o2.getTitle());
  o2.setTitle(tmp);
}
```



```java
// Nguyen Huu Duc - 20210192
public class TestPassingParameter {
    public static void main (String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");

        swap(jungleDVD, cinderellaDVD);
        System.out.println("Jungle DVD's title: "+jungleDVD.getTitle());
        System.out.println("Cinderella DVD's title: "+cinderellaDVD.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("Jungle DVD's title: "+jungleDVD.getTitle());
    }
```



```
Jungle DVD's title: Cinderella
Cinderella DVD's title: Jungle
Jungle DVD's title: Jungle


------------------
(program exited with code: 0)
Press return to continue
```

4. Use debug run

4.1 Debugging Java in Eclipse

4.2 Example of debug run for the swap() method

► Follow instructions.

4.2.1 Setting, deleting & deactivating breakpoints

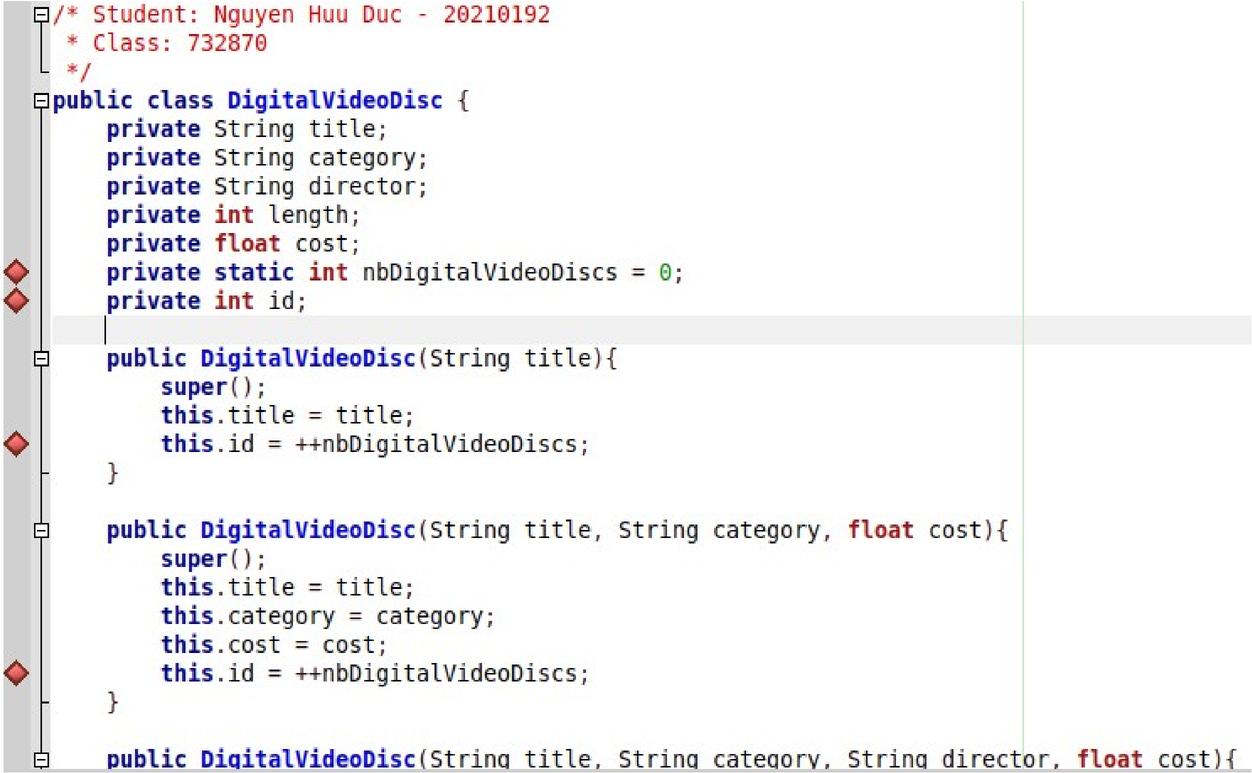
4.2.2 Run in Debug mode

4.2.3 Step Into, Step Over, Step Return, Resume

4.2.4 Investigate value of variables

4.2.5 Change value of variables

5. Classifier Member and Instance Member



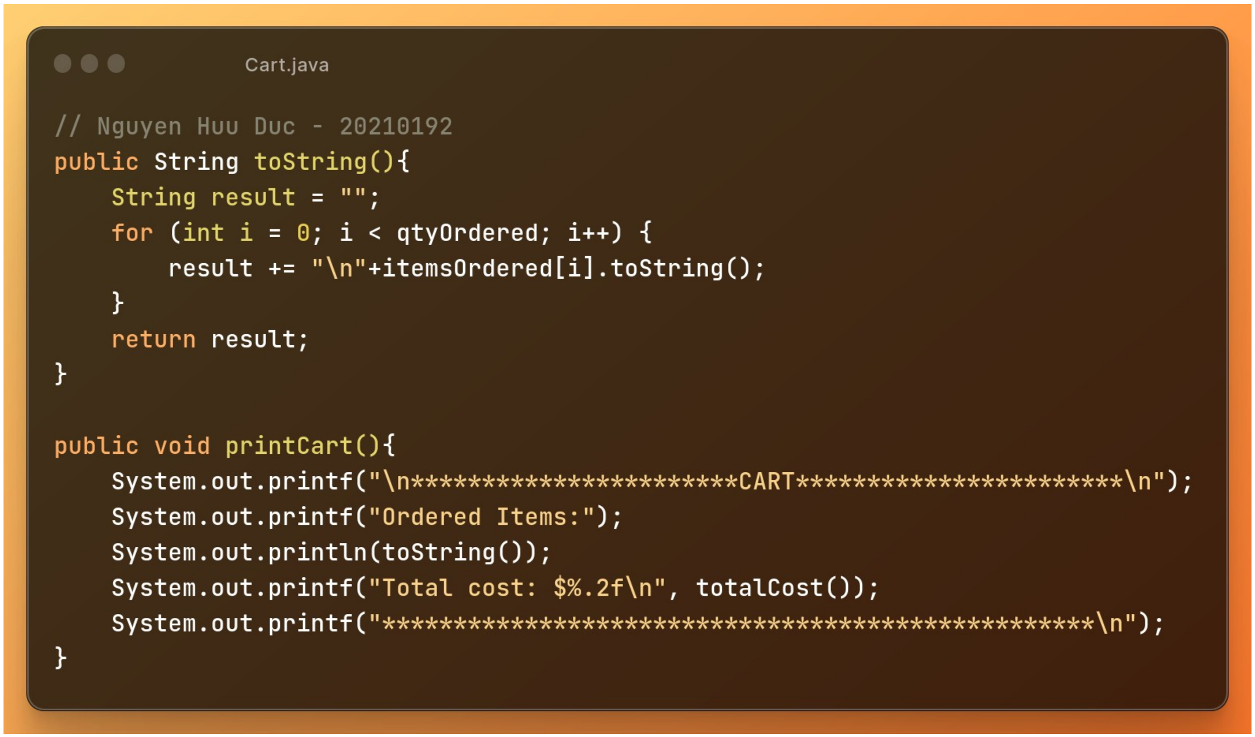
```
/* Student: Nguyen Huu Duc - 20210192
 * Class: 732870
 */
public class DigitalVideoDisc {
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;
    private static int nbDigitalVideoDiscs = 0;
    private int id;

    public DigitalVideoDisc(String title){
        super();
        this.title = title;
        this.id = ++nbDigitalVideoDiscs;
    }

    public DigitalVideoDisc(String title, String category, float cost){
        super();
        this.title = title;
        this.category = category;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }

    public DigitalVideoDisc(String title, String category, String director, float cost){
```

We have created a class attribute named "nbDigitalVideoDiscs" in and an instance attribute named "id" in the class DigitalVideoDisc.

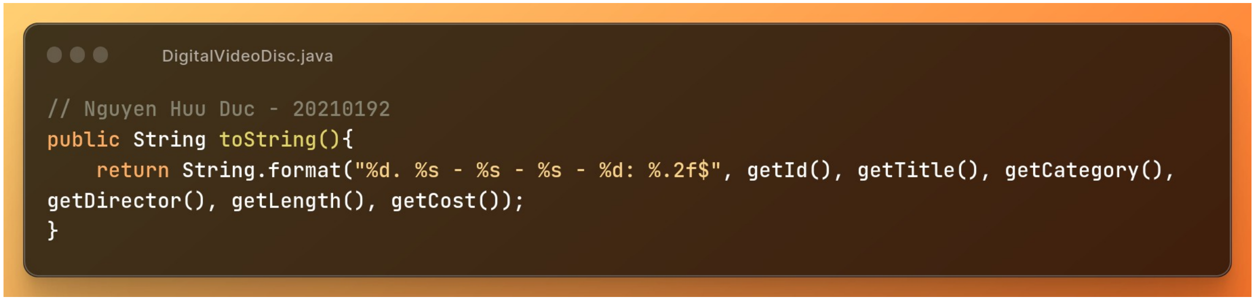
Each time an instance of the DigitalVideoDisc class is created, nbDigitalVideoDisc is updated and its ID is assigned.

6. Open the Cart class

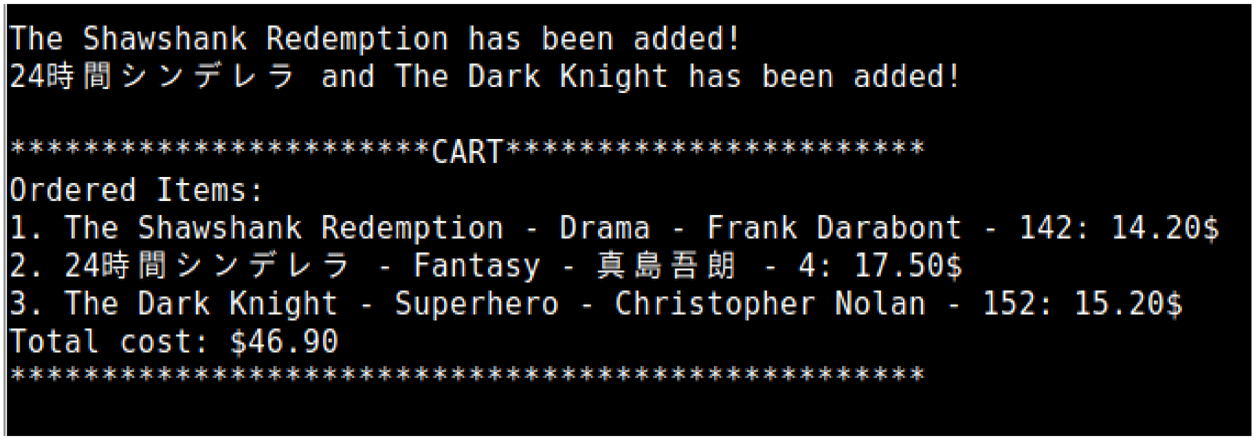
- Create a new method printCart() to show cart info in given format



```java
// Nguyen Huu Duc - 20210192
public String toString(){
    String result = "";
    for (int i = 0; i < qtyOrdered; i++) {
        result += "\n"+itemsOrdered[i].toString();
    }
    return result;
}

public void printCart(){
    System.out.printf("\n***********************CART***********************\n");
    System.out.printf("Ordered Items:");
    System.out.println(toString());
    System.out.printf("Total cost: $%.2f\n", totalCost());
    System.out.printf("**************************************************\n");
}
```



```java
// Nguyen Huu Duc - 20210192
public String toString(){
    return String.format("%d. %s - %s - %s - %d: %.2f$", getId(), getTitle(), getCategory(),
getDirector(), getLength(), getCost());
}
```

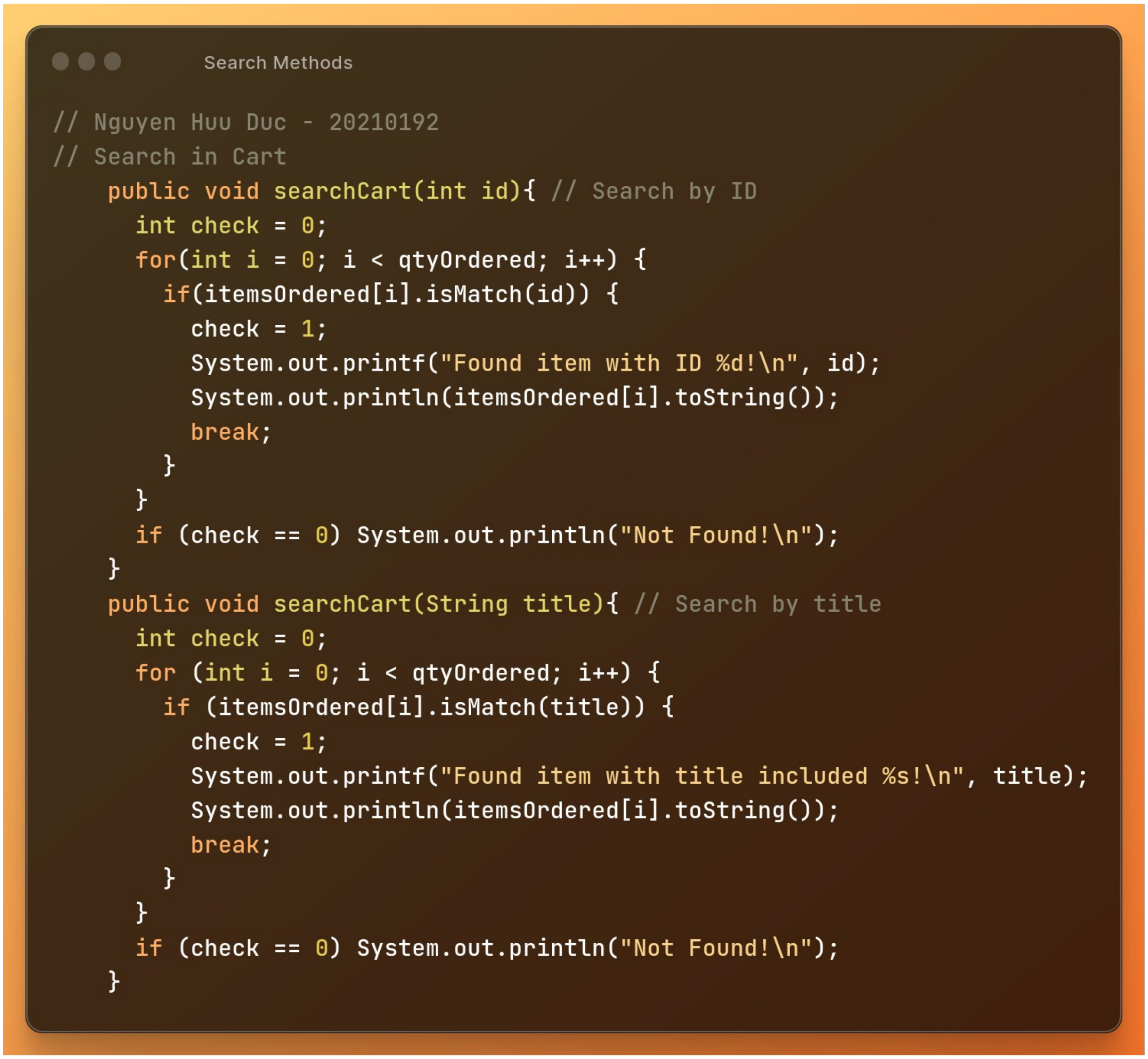


```
The Shawshank Redemption has been added!
24時間シンデレラ and The Dark Knight has been added!

***********************CART***********************
Ordered Items:
1. The Shawshank Redemption - Drama - Frank Darabont - 142: 14.20$
2. 24時間シンデレラ - Fantasy - 真島吾朗 - 4: 17.50$
3. The Dark Knight - Superhero - Christopher Nolan - 152: 15.20$
Total cost: $46.90
**************************************************
```

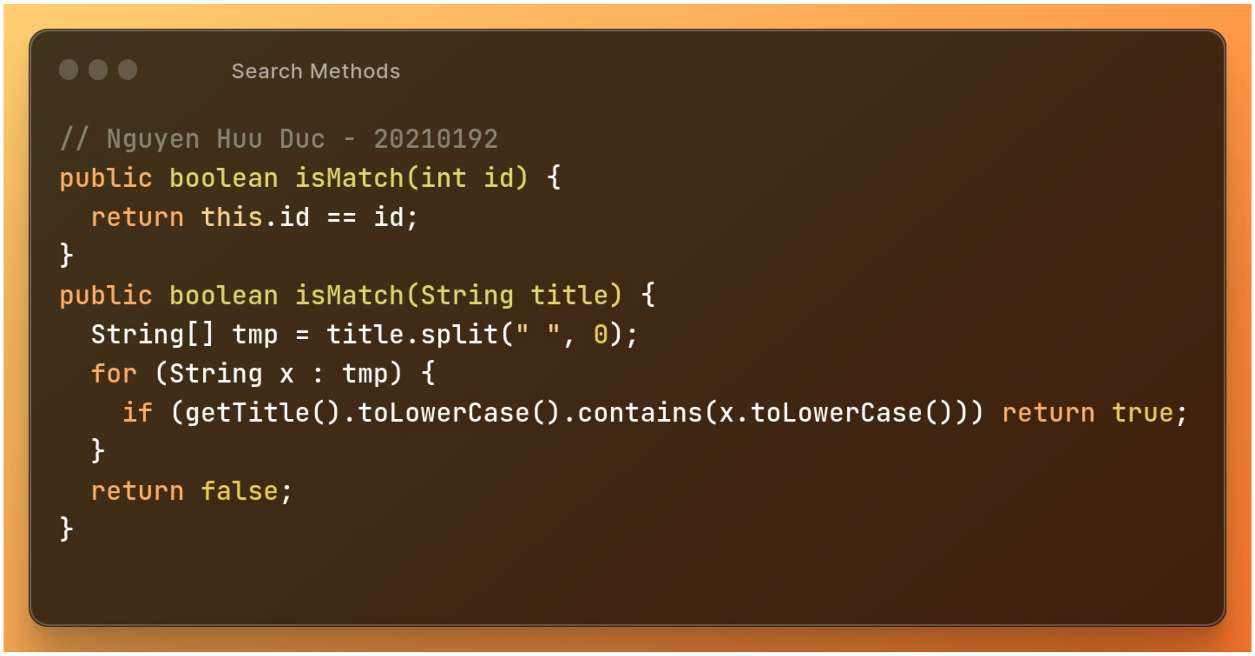
- What should be the return type of toString() method for the DigitalVideoDisc class?

String should be the return type.

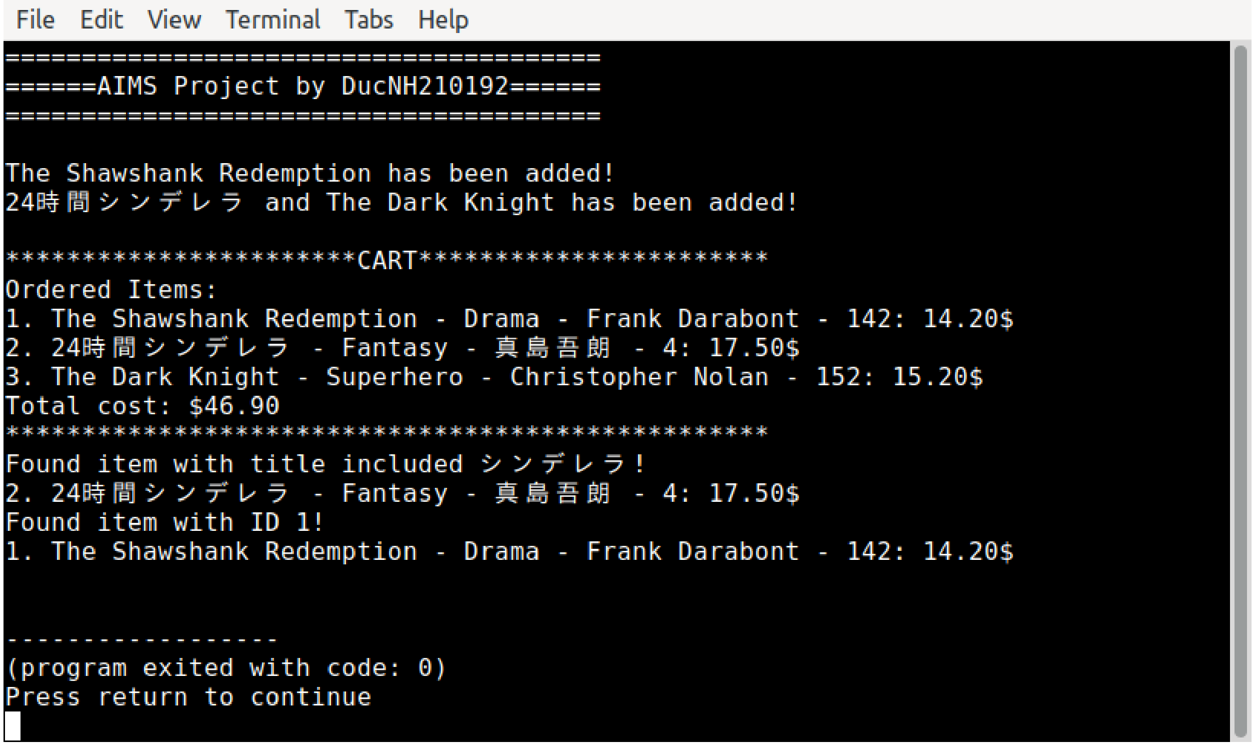
- Search methods and isMatch method



```
// Nguyen Huu Duc - 20210192
// Search in Cart
    public void searchCart(int id){ // Search by ID
      int check = 0;
      for(int i = 0; i < qtyOrdered; i++) {
        if(itemsOrdered[i].isMatch(id)) {
          check = 1;
          System.out.printf("Found item with ID %d!\n", id);
          System.out.println(itemsOrdered[i].toString());
          break;
        }
      }
      if (check == 0) System.out.println("Not Found!\n");
    }
    public void searchCart(String title){ // Search by title
      int check = 0;
      for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].isMatch(title)) {
          check = 1;
          System.out.printf("Found item with title included %s!\n", title);
          System.out.println(itemsOrdered[i].toString());
          break;
        }
      }
      if (check == 0) System.out.println("Not Found!\n");
    }
```

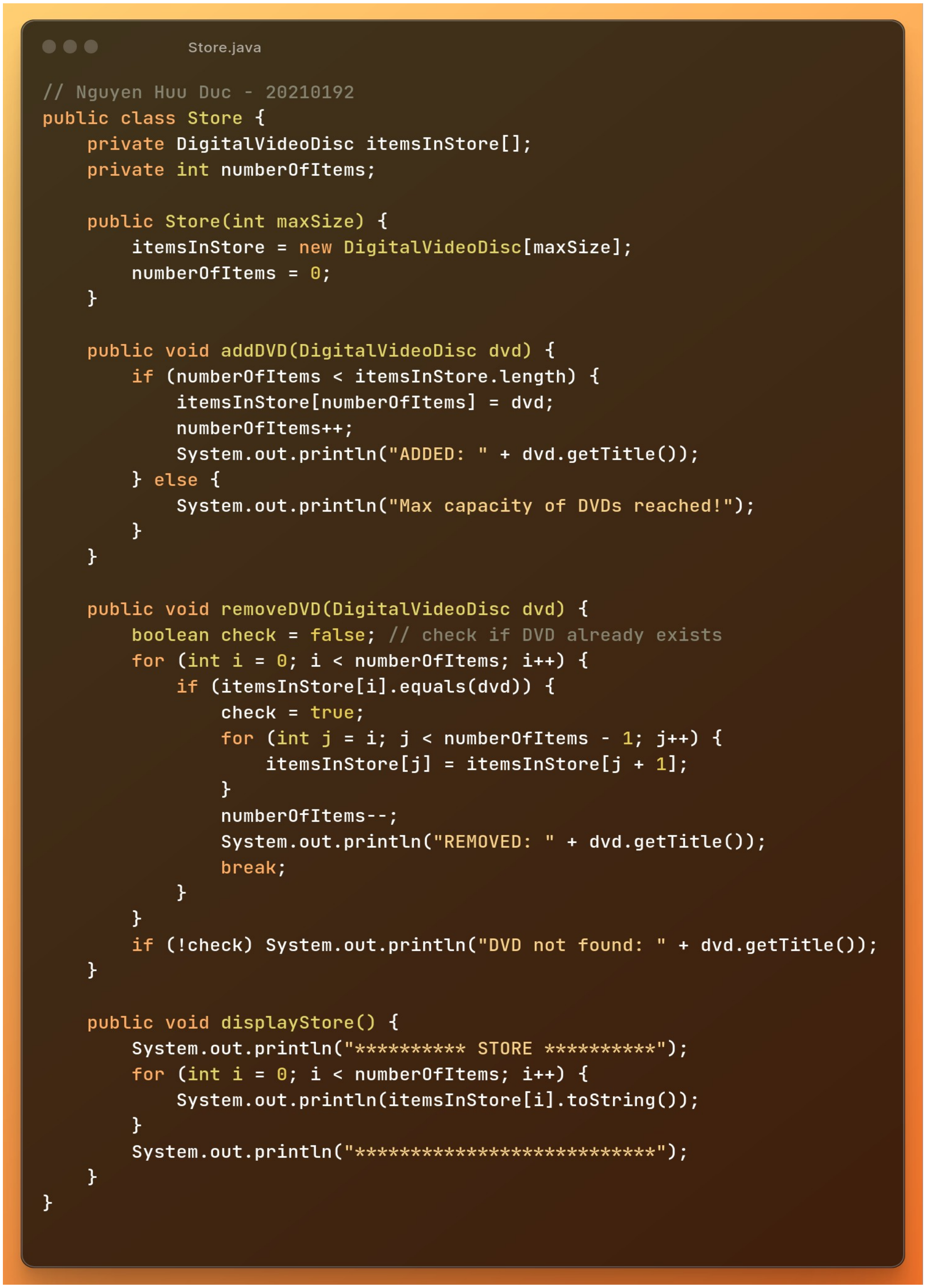


```java
// Nguyen Huu Duc - 20210192
public boolean isMatch(int id) {
  return this.id == id;
}
public boolean isMatch(String title) {
  String[] tmp = title.split(" ", 0);
  for (String x : tmp) {
    if (getTitle().toLowerCase().contains(x.toLowerCase())) return true;
  }
  return false;
}
```

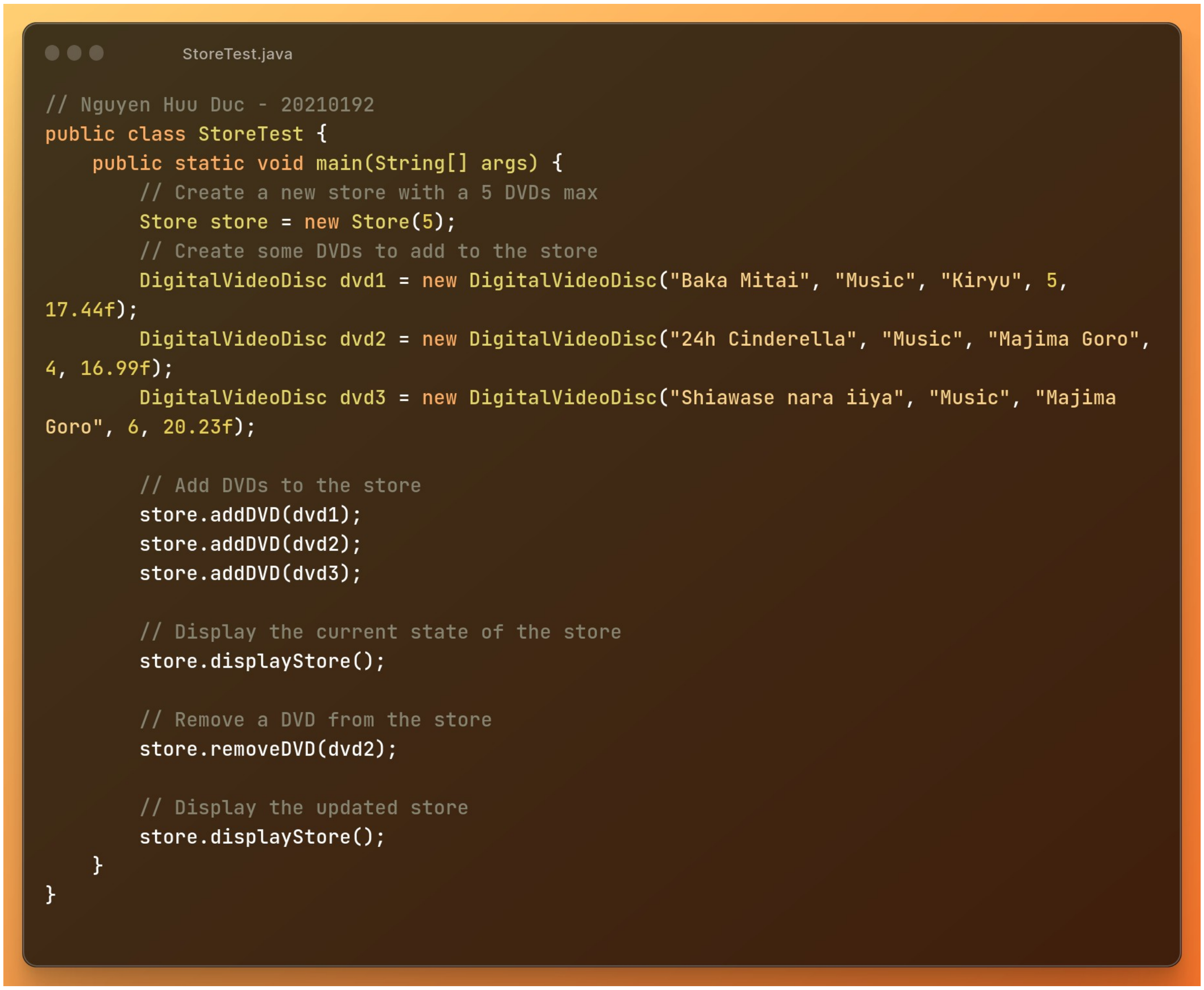


```
=======================================
======AIMS Project by DucNH210192======
=======================================

The Shawshank Redemption has been added!
24時間シンデレラ and The Dark Knight has been added!

**********************CART***********************
Ordered Items:
1. The Shawshank Redemption - Drama - Frank Darabont - 142: 14.20$
2. 24時間シンデレラ - Fantasy - 真島吾朗 - 4: 17.50$
3. The Dark Knight - Superhero - Christopher Nolan - 152: 15.20$
Total cost: $46.90
*************************************************
Found item with title included シンデレラ!
2. 24時間シンデレラ - Fantasy - 真島吾朗 - 4: 17.50$
Found item with ID 1!
1. The Shawshank Redemption - Drama - Frank Darabont - 142: 14.20$


------------------
(program exited with code: 0)
Press return to continue
```

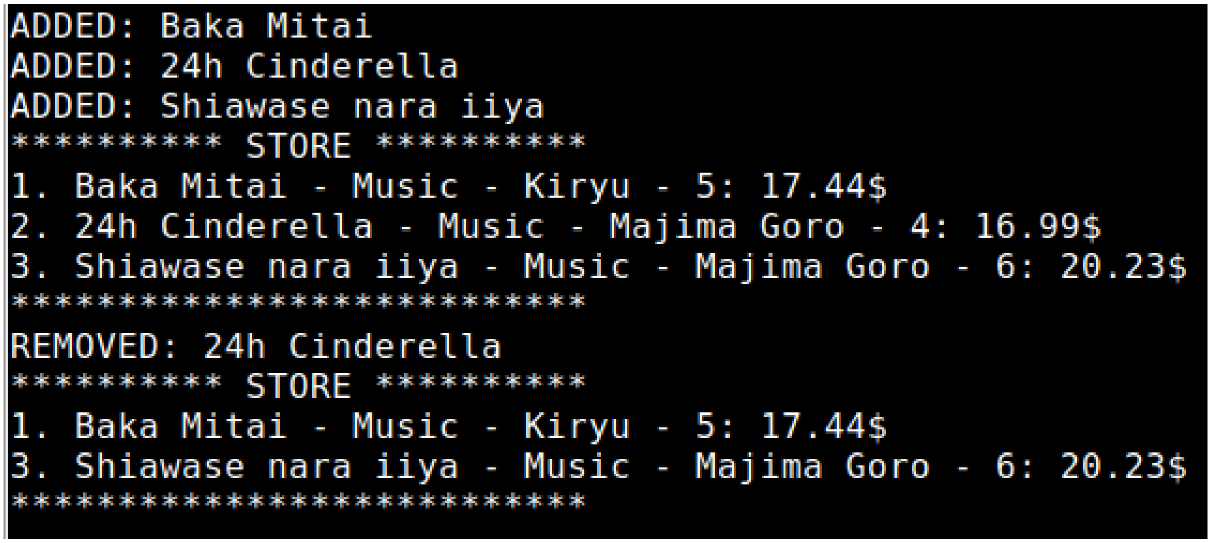
7. Implement the Store class



```java
// Nguyen Huu Duc - 20210192
public class Store {
    private DigitalVideoDisc itemsInStore[];
    private int numberOfItems;

    public Store(int maxSize) {
        itemsInStore = new DigitalVideoDisc[maxSize];
        numberOfItems = 0;
    }

    public void addDVD(DigitalVideoDisc dvd) {
        if (numberOfItems < itemsInStore.length) {
            itemsInStore[numberOfItems] = dvd;
            numberOfItems++;
            System.out.println("ADDED: " + dvd.getTitle());
        } else {
            System.out.println("Max capacity of DVDs reached!");
        }
    }

    public void removeDVD(DigitalVideoDisc dvd) {
        boolean check = false; // check if DVD already exists
        for (int i = 0; i < numberOfItems; i++) {
            if (itemsInStore[i].equals(dvd)) {
                check = true;
                for (int j = i; j < numberOfItems - 1; j++) {
                    itemsInStore[j] = itemsInStore[j + 1];
                }
                numberOfItems--;
                System.out.println("REMOVED: " + dvd.getTitle());
                break;
            }
        }
        if (!check) System.out.println("DVD not found: " + dvd.getTitle());
    }

    public void displayStore() {
        System.out.println("********** STORE **********");
        for (int i = 0; i < numberOfItems; i++) {
            System.out.println(itemsInStore[i].toString());
        }
        System.out.println("***************************");
    }
}
```



```java
// Nguyen Huu Duc - 20210192
public class StoreTest {
    public static void main(String[] args) {
        // Create a new store with a 5 DVDs max
        Store store = new Store(5);
        // Create some DVDs to add to the store
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("Baka Mitai", "Music", "Kiryu", 5,
17.44f);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("24h Cinderella", "Music", "Majima Goro",
4, 16.99f);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Shiawase nara iiya", "Music", "Majima
Goro", 6, 20.23f);

        // Add DVDs to the store
        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);

        // Display the current state of the store
        store.displayStore();

        // Remove a DVD from the store
        store.removeDVD(dvd2);

        // Display the updated store
        store.displayStore();
    }
}
```

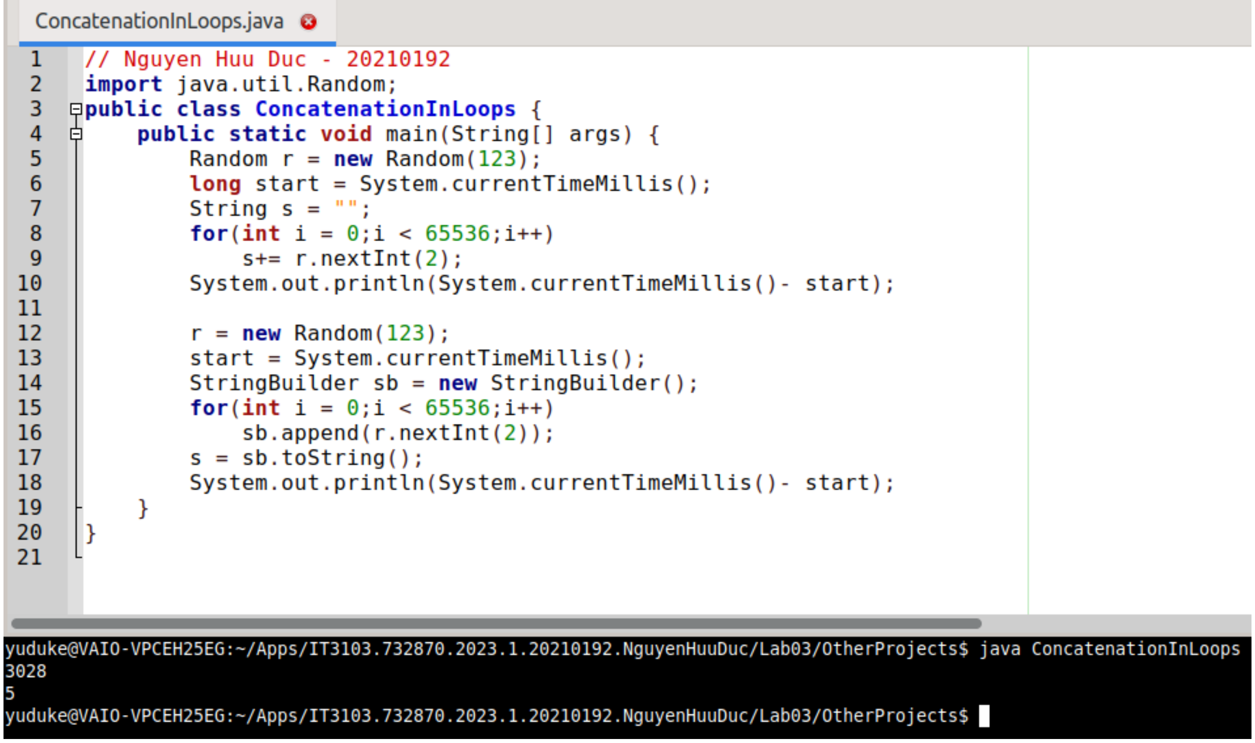


```
ADDED: Baka Mitai
ADDED: 24h Cinderella
ADDED: Shiawase nara iiya
********** STORE **********
1. Baka Mitai - Music - Kiryu - 5: 17.44$
2. 24h Cinderella - Music - Majima Goro - 4: 16.99$
3. Shiawase nara iiya - Music - Majima Goro - 6: 20.23$
***************************
REMOVED: 24h Cinderella
********** STORE **********
1. Baka Mitai - Music - Kiryu - 5: 17.44$
3. Shiawase nara iiya - Music - Majima Goro - 6: 20.23$
***************************
```

8. Re-organize your project

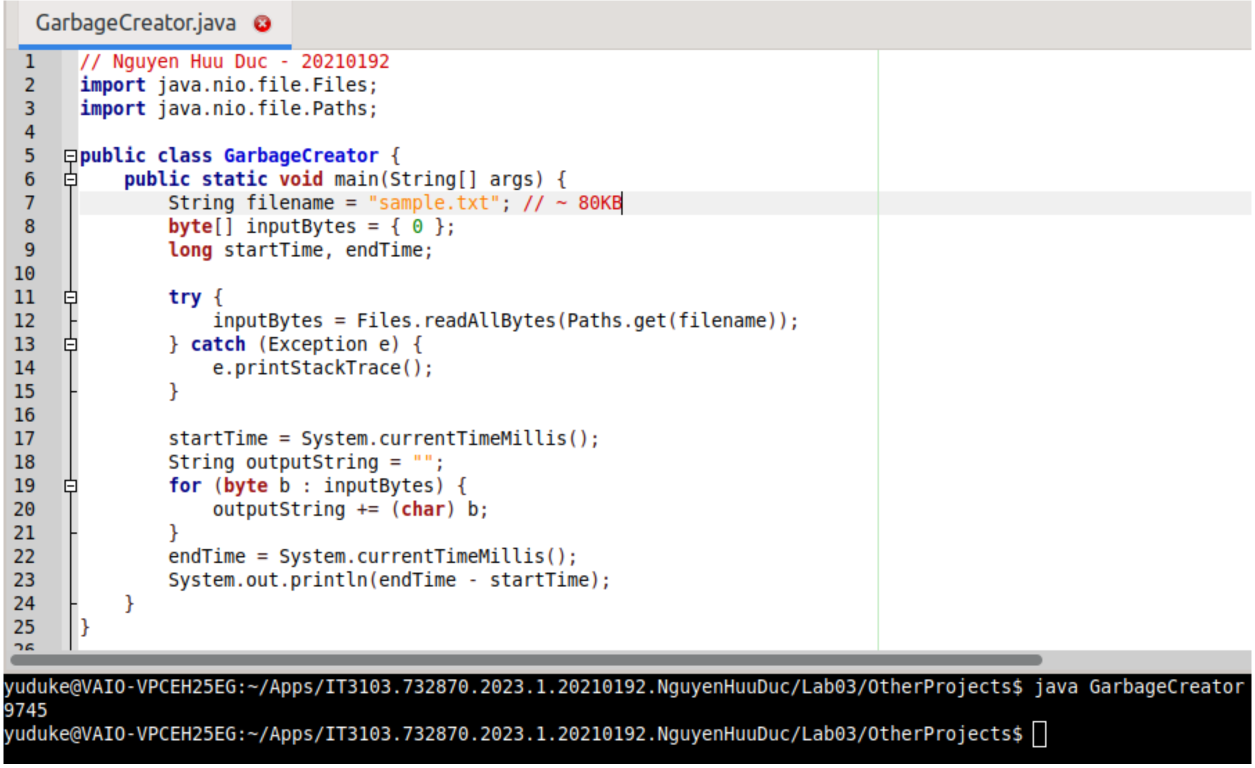
Rename project, use packages and re-organize all hands-on labs and exercises.

9. String, StringBuilder and StringBuffer



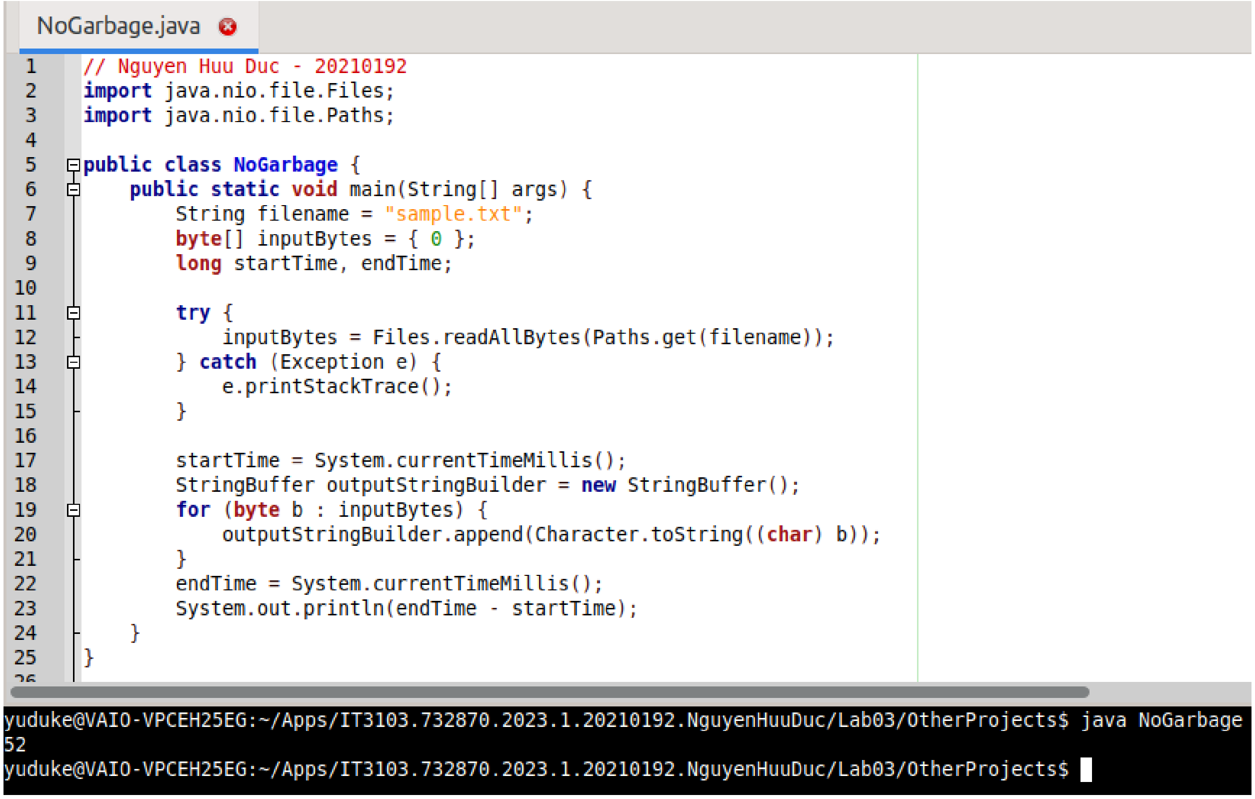
```java
// Nguyen Huu Duc - 20210192
import java.util.Random;
public class ConcatenationInLoops {
    public static void main(String[] args) {
        Random r = new Random(123);
        long start = System.currentTimeMillis();
        String s = "";
        for(int i = 0;i < 65536;i++)
            s+= r.nextInt(2);
        System.out.println(System.currentTimeMillis()- start);

        r = new Random(123);
        start = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for(int i = 0;i < 65536;i++)
            sb.append(r.nextInt(2));
        s = sb.toString();
        System.out.println(System.currentTimeMillis()- start);
    }
}
```

```
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc/Lab03/OtherProjects$ java ConcatenationInLoops
3028
5
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc/Lab03/OtherProjects$
```



```java
// Nguyen Huu Duc - 20210192
import java.nio.file.Files;
import java.nio.file.Paths;

public class GarbageCreator {
    public static void main(String[] args) {
        String filename = "sample.txt"; // ~ 80KB
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (Exception e) {
            e.printStackTrace();
        }

        startTime = System.currentTimeMillis();
        String outputString = "";
        for (byte b : inputBytes) {
            outputString += (char) b;
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

```
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc/Lab03/OtherProjects$ java GarbageCreator
9745
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc/Lab03/OtherProjects$
```



```java
// Nguyen Huu Duc - 20210192
import java.nio.file.Files;
import java.nio.file.Paths;

public class NoGarbage {
    public static void main(String[] args) {
        String filename = "sample.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (Exception e) {
            e.printStackTrace();
        }

        startTime = System.currentTimeMillis();
        StringBuffer outputStringBuilder = new StringBuffer();
        for (byte b : inputBytes) {
            outputStringBuilder.append(Character.toString((char) b));
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

```
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc/Lab03/OtherProjects$ java NoGarbage
52
yuduke@VAIO-VPCEH25EG:~/Apps/IT3103.732870.2023.1.20210192.NguyenHuuDuc/Lab03/OtherProjects$
```

When concatenates, Java creates a new string that contains the combined contents of the two original strings. This process involves allocating memory for the new string, copying the characters from the original strings, and updating the reference to the new string.

In contrast, appending to a StringBuilder does not create a new string . StringBuilder maintains a buffer that stores the characters of the string. When appends, the new characters are simply added to the end of the buffer.

The first loop using string concatenation takes significantly longer time than the second loop using StringBuilder. This is because the first loop creates a new string for each iteration of the loop, whereas the second loop only appends to a single StringBuilder.

10. Release flow demonstration

- Update local repository.

- Create, switch to a new branch in local repository and make modifications.

- Commit the change.

- Create a new branch in remote repository.

- Push the local repo to the remote repo.

- Create a pull request.

- Merge the new remote branch to the master branch.