

BÁO CÁO THỰC HÀNH LAB 03

THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Table of Contents

1. Working with method overloading	3
1.1. Overloading by differing types of parameter	3
Code	3
Result.....	5
1.2. Overloading by differing the number of parameters	6
Code	6
Result.....	7
2. Passing parameter	7
Code	7
Result.....	8
3. Class Member and Instance Member	9
Code	9
Result.....	12
4. Open the Cart class	12
Code	12
5. Implement the Store class.....	17
Code	17
Result.....	18
6. String, StringBuilder and StringBuffer	19
Code	19
Result.....	20
7 Class Diagram	22

Table of Figures

Figure 1: Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList).....	3
Figure 2 Aims Class.....	5
Figure 3 Result	5
Figure 4 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList).....	6
Figure 5 Aims Class.....	6
Figure 6 Result	7
Figure 7 Passing parameter code.....	8
Figure 8 Passing parameter result	8
Figure 9 Class member and Instance member	11
Figure 10 Result	12
Figure 11 Code method to print the content in Cart	13
Figure 12 Result	13
Figure 13 Code method search by id	14
Figure 14 Code method search by title.....	14
Figure 15 isMatch() in DigitalVideoDisc class.....	15
Figure 16 CartTest class.....	15
Figure 17 Result search by Id	16
Figure 18 Result search by Title	16
Figure 19 Store Class.....	17
Figure 20 StoreTest class.....	18
Figure 21 Result	18
Figure 22 So sánh String, StringBuilder, StringBuffer	19
Figure 23 GarbageCreator class	20
Figure 24 Result: program hangs	20
Figure 25 NoGarbage class.....	21
Figure 26 Result: Program finishes quickly	21
Figure 27 Class Diagram	22

1. Working with method overloading

1.1. Overloading by differing types of parameter

Code

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

Figure 1: Method `addDigitalVideoDisc(DigitalVideoDisc [] dvdList)`

Try to add a method `addDigitalVideoDisc` which allows to pass an arbitrary number of arguments for dvd. Compare to an array parameter. What do you prefer in this case?

Khi thay bằng: `addDigitalVideoDisc(DigitalVideoDisc... dvdList)`:

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

Kết quả:

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

Aims x

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_r
The disc has been added.
The disc has been added.
The disc has been added.
Total cost is:
63.89
Process finished with exit code 0

So sánh:

...	[]
Dùng để truyền số lượng tham số không xác định vào một phương thức.	Dùng để lưu trữ một tập hợp các giá trị cố định với kích thước xác định.
Thích hợp cho các phương thức linh hoạt với số lượng tham số không biết trước.	Phù hợp cho các trường hợp mà ta biết chính xác số lượng phần tử ngay từ đầu.
Tiện lợi hơn khi gọi phương thức với nhiều tham số mà không cần tạo mảng thủ công.	Ít tiện lợi hơn vì luôn cần tạo mảng trước khi truyền.

Em thích dùng `addDigitalVideoDisc(DigitalVideoDisc... dvdList)` hơn vì không cần tạo mảng thủ công.

```
package hust.soict.hedspi.aims;

import hust.soict.hedspi.aims.cart.Cart;
import hust.soict.hedspi.aims.disc.DigitalVideoDisc;

public class Aims {
    public static void main(String[] args) {
        //Create a new card
        Cart anOrder = new Cart();
        //Trinh Huu An 20225593
        DigitalVideoDisc dvd1 = new DigitalVideoDisc(
            title: "The Lion King",
            category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
        // anOrder.addDigitalVideoDisc(dvd1);

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

        DigitalVideoDisc dvd3 = new DigitalVideoDisc(
            title: "Aladdin",
            category: "Animation", cost: 18.99f);
        // anOrder.addDigitalVideoDisc(dvd3);

        // anOrder.removeDigitalVideoDisc(dvd2);

        DigitalVideoDisc[] dvdList = {dvd1, dvd2, dvd3};
        anOrder.addDigitalVideoDisc(dvdList);

        System.out.println("Total cost is: ");
        System.out.print(anOrder.totalCost());
    }
}
```

Figure 2 Aims Class

Result

```
C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=64063:
The disc has been added.
The disc has been added.
The disc has been added.
Total cost is:
63.89
Process finished with exit code 0
```

Figure 3 Result

1.2. Overloading by differing the number of parameters

Code

```
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) { 1 usage new *
    if(qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = dvd1;
        itemsOrdered[qtyOrdered + 1] = dvd2;
        qtyOrdered += 2;
        System.out.println("The disc " + dvd1.getTitle() + " and" + dvd2.getTitle() + " has been added.");
    } else {
        System.out.println("The cart is almost full.");
    }
}
```

Figure 4 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

```
package hust.soict.hedspi.aims;

import hust.soict.hedspi.aims.cart.Cart;
import hust.soict.hedspi.aims.disc.DigitalVideoDisc;

> public class Aims { ① Carbon2301 *
>     public static void main(String[] args) { ① Carbon2301 *
        //Create a new card
        Cart anOrder = new Cart();
        //Trinh Huu An 20225593
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King",
            category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
        //
        anOrder.addDigitalVideoDisc(dvd1);

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

        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladdin",
            category: "Animation", cost: 18.99f);
        //
        anOrder.addDigitalVideoDisc(dvd3);
        //
        anOrder.removeDigitalVideoDisc(dvd2);

        DigitalVideoDisc[] dvdList = {dvd1, dvd2, dvd3};
        anOrder.addDigitalVideoDisc(dvd1, dvd2);

        System.out.println("Total cost is: ");
        System.out.print(anOrder.totalCost());
    }
}
```

Figure 5 Aims Class

Result



```

DigitalVideoDisc[] dvdList = {dvd1, dvd2, dvd3};
anOrder.addDigitalVideoDisc(dvd1, dvd2);

System.out.println("Total cost is: ");
System.out.print(anOrder.totalCost());
}
}

```

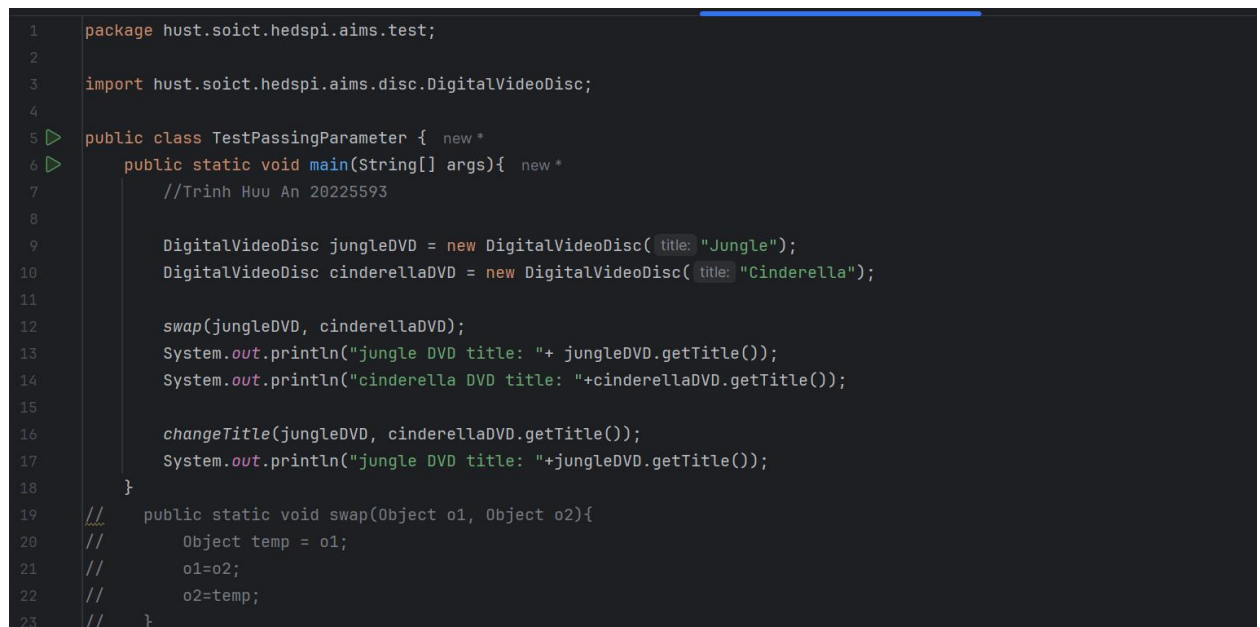
Aims x

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\
The disc The Lion King and Star Wars has been added.
Total cost is:
44.9
Process finished with exit code 0

Figure 6 Result

2. Passing parameter

Code



```

1 package hust.soict.hedspi.aims.test;
2
3 import hust.soict.hedspi.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter { new *
6     public static void main(String[] args){ new *
7         //Trinh Huu An 20225593
8
9         DigitalVideoDisc jungleDVD = new DigitalVideoDisc( title: "Jungle");
10        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc( title: "Cinderella");
11
12        swap(jungleDVD, cinderellaDVD);
13        System.out.println("jungle DVD title: "+ jungleDVD.getTitle());
14        System.out.println("cinderella DVD title: "+cinderellaDVD.getTitle());
15
16        changeTitle(jungleDVD, cinderellaDVD.getTitle());
17        System.out.println("jungle DVD title: "+jungleDVD.getTitle());
18    }
19    // public static void swap(Object o1, Object o2){
20    //     Object temp = o1;
21    //     o1=o2;
22    //     o2=temp;
23    // }

```

```

public static void swap(DigitalVideoDisc o1, DigitalVideoDisc o2){ 1usage new *
    String tmp1 = o1.getTitle();
    String tmp2 = o2.getTitle();
    o1.setTitle(tmp2);
    o2.setTitle(tmp1);
}
public static void changeTitle(DigitalVideoDisc dvd, String title){ 1usage new *
    String oldTitle = dvd.getTitle(); // Lấy tiêu đề cũ của dvd
    dvd.setTitle(title); // Thay đổi tiêu đề của đối tượng dvd bằng setter
    dvd = new DigitalVideoDisc(oldTitle); // Thay đổi tham chiếu của đối tượng dvd (không ảnh hưởng đến đối tượng ban đầu)
}
}

```

Figure 7 Passing parameter code

Result

```

1 package hust.soict.hedspi.aims.test;
2
3 import hust.soict.hedspi.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter { new *
6     public static void main(String[] args){ new *
7         //Trinh Huu An 20225593
8     }
9 }

```

Run TestPassingParameter x

```

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=50
jungle DVD title: Cinderella
cinderella DVD title: Jungle
jungle DVD title: Jungle
Process finished with exit code 0

```

Figure 8 Passing parameter result

Question: Is JAVA a Pass by Value or a Pass by Reference programming language?

Java truyền tham số bằng giá trị:

- Khi một đối tượng được truyền vào phương thức, tham chiếu đến đối tượng được **sao chép** và truyền vào.
- Do đó, bên trong phương thức, bạn có thể sửa đổi các thuộc tính của đối tượng thông qua tham chiếu này, và sự thay đổi sẽ phản ánh ra bên ngoài.

Không phải Pass by Reference:

- Java **không truyền trực tiếp tham chiếu của biến** mà truyền bản sao của tham chiếu đó. Điều này khác với truyền tham chiếu thực sự trong một số ngôn ngữ như C hay C++.

Question: After the call of swap(jungleDVD, cinderellaDVD) why does the title of these two objects still remain?

- Khi ta gọi swap(jungleDVD, cinderellaDVD), giá trị của các tham chiếu (jungleDVD và cinderellaDVD) được sao chép và truyền vào tham số o1 và o2 trong phương thức swap.
- Bất kỳ thay đổi nào đối với o1 và o2 bên trong phương thức **chỉ ảnh hưởng đến bản sao** của các tham chiếu này, không ảnh hưởng đến bản gốc (jungleDVD và cinderellaDVD).

Question: After the call of `changeTitle(jungleDVD, cinderellaDVD.getTitle())` why is the title of the JungleDVD changed?

Tiêu đề của jungleDVD thay đổi vì:

- Gọi `dvd.setTitle(title)` đã trực tiếp sửa đổi nội dung của đối tượng gốc mà jungleDVD đang trỏ tới.
- Việc gán `dvd = new DigitalVideoDisc(oldTitle)` chỉ ảnh hưởng đến tham chiếu cục bộ dvd, không làm thay đổi tham chiếu jungleDVD bên ngoài.

3. Class Member and Instance Member

Code

```
package hust.soict.hedspi.aims.disc;

public class DigitalVideoDisc { 23 usages  ▲ Carbon2301 *
    private String title; 6 usages
    private String category; 5 usages
    private String director; 4 usages
    private int length; 3 usages
    private float cost; 5 usages
    private int id; 5 usages

    // Class variable
    private static int nbDigitalVideoDiscs = 0; 8 usages

    // Trinh Huu An 20225593
    public DigitalVideoDisc(String title) { 3 usages  ▲ Carbon2301 *
        this.title = title;

        // Update class variable and assign id
        nbDigitalVideoDiscs++;
        this.id = nbDigitalVideoDiscs;
    }
}
```

```
public DigitalVideoDisc(String title, String category, float cost) { 1 usage  ⬆ Carbon2301 *
    this.title = title;
    this.category = category;
    this.cost = cost;

    // Update class variable and assign id
    nbDigitalVideoDiscs++;
    this.id = nbDigitalVideoDiscs;
}

public DigitalVideoDisc(String title, String category, String director, float cost) { no usages  ⬆ Carbon2301 *
    this.title = title;
    this.category = category;
    this.director = director;
    this.cost = cost;

    // Update class variable and assign id
    nbDigitalVideoDiscs++;
    this.id = nbDigitalVideoDiscs;
}

public DigitalVideoDisc(String title, String category, String director, int length, float cost) { 2 usages  ⬆ Carbon2301 *
    this.title = title;
    this.category = category;
    this.director = director;
    this.length = length;
    this.cost = cost;

    // Update class variable and assign id
    nbDigitalVideoDiscs++;
    this.id = nbDigitalVideoDiscs;
}

public int getId() { 3 usages  new *
    return id;
}

public String getTitle() { 14 usages  ⬆ Carbon2301
    return title;
}
```

```
public String getCategory() { no usages  ↗ Carbon2301
    return category;
}

public String getDirector() { no usages  ↗ Carbon2301
    return director;
}

public int getLength() { no usages  ↗ Carbon2301
    return length;
}

public float getCost() { 1 usage  ↗ Carbon2301
    return cost;
}

public void setTitle(String title) { 3 usages  ↗ Carbon2301
    this.title = title;
}

public void setCategory(String category) { no usages  ↗ Carbon2301
    this.category = category;
}

public void setDirector(String director) { no usages  ↗ Carbon2301
    this.director = director;
}

public void setLength(int length) { no usages  ↗ Carbon2301
    this.length = length;
}

public void setCost(float cost) { no usages  ↗ Carbon2301
    this.cost = cost;
}
}
```

Figure 9 Class member and Instance member

Result



The screenshot shows an IDE with a Java file named 'Aims'. The code in the editor includes comments and three print statements for DVD objects. Below the editor, the 'Run' tab displays the execution output, which shows three 'The disc has been added.' messages followed by the IDs for 'The Lion King', 'Star Wars', and 'Aladdin'. The process finished with exit code 0.

```
//      System.out.println("Total cost is: ");
//      System.out.print(anOrder.totalCost());

System.out.println("The id of " + dvd1.getTitle() + " is " + dvd1.getId());
System.out.println("The id of " + dvd2.getTitle() + " is " + dvd2.getId());
System.out.println("The id of " + dvd3.getTitle() + " is " + dvd3.getId());
}
}
```

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=51319:C:\SET_UP\..."
The disc has been added.
The disc has been added.
The disc has been added.
The id of The Lion King is 1
The id of Star Wars is 2
The id of Aladdin is 3

Process finished with exit code 0

Figure 10 Result

4. Open the Cart class

Code



The screenshot shows the 'toString()' method of the 'Cart' class. The method is annotated with '@Override' and 'new *'. It returns a string representation of the cart contents, including the title, category, director, length, and price of each item, separated by commas and enclosed in curly braces.

```
100      @Override new *
101      public String toString() {
102          return "DVD{" +
103              "Title='" + title + '\'' +
104              ", Category='" + category + '\'' +
105              ", Director='" + director + '\'' +
106              ", Length=" + length +
107              ", Price=" + cost +
108              '}';
109      }
110
111 }
```

```

    public void printCart() { 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].toString());
        }
        System.out.println("Total cost: " + totalCost() + "$");
        System.out.println("*****");
    }
}

package hust.soict.hedspi.aims.cart;

import hust.soict.hedspi.aims.disc.DigitalVideoDisc;
import java.util.Scanner;

public class CartTest { new *
    public static void main(String[] args) { new *
        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( title: "Aladin",
            category: "Animation", cost: 18.99f);
        cart.addDigitalVideoDisc(dvd3);

        cart.printCart();
    }
}

```

Figure 11 Code method to print the content in Cart

```

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=
The disc 'The Lion King' has been added.
The disc 'Star Wars' has been added.
The disc 'Aladin' has been added.
*****CART*****
Ordered Items:
1. DVD{Title='The Lion King', Category='Animation', Director='Roger Allers', Length=87, Price=19.95}
2. DVD{Title='Star Wars', Category='Science Fiction', Director='George Lucas', Length=87, Price=24.95}
3. DVD{Title='Aladin', Category='Animation', Director='null', Length=0, Price=18.99}
Total cost: 63.89$
*****
Process finished with exit code 0

```

Figure 12 Result

```
public boolean search(int id) { 1usage new *
    for (DigitalVideoDisc disc : itemsOrdered) {
        if (disc != null && disc.getId() == id) {
            return true;
        }
    }
    return false;
}

public DigitalVideoDisc getDiscById(int id) { 1usage new *
    for (DigitalVideoDisc disc : itemsOrdered) {
        if (disc != null && disc.getId() == id) {
            return disc;
        }
    }
    return null;
}
```

Figure 13 Code method search by id

```
public boolean search(String title) { 1usage new *
    for (DigitalVideoDisc disc : itemsOrdered) {
        if (disc != null && disc.isMatch(title)) {
            return true;
        }
    }
    return false;
}

public DigitalVideoDisc getDiscByTitle(String title) { 1usage new *
    for (DigitalVideoDisc disc : itemsOrdered) {
        if (disc != null && disc.isMatch(title)) {
            return disc;
        }
    }
    return null;
}
```

Figure 14 Code method search by title

```
public boolean isMatch(String title) { 2 usages new *
    return this.title.equalsIgnoreCase(title); // So sánh không phân biệt hoa thường
}
```

Figure 15 isMatch() in DigitalVideoDisc class

```
// Lựa chọn tìm kiếm theo ID hoặc tiêu đề
Scanner sc = new Scanner(System.in);
System.out.println("Choose search type (1: By ID, 2: By Title): ");
int searchType = sc.nextInt();
sc.nextLine(); // Đọc dòng trống sau nextInt()

if (searchType == 1) {
    // Tìm kiếm theo ID
    System.out.println("Enter the ID of DVD you want to search: ");
    int idForSearch = sc.nextInt();

    if (cart.search(idForSearch)) {
        System.out.println("Found DVD with ID " + idForSearch + ":");
        System.out.println(cart.getDiscById(idForSearch));
    } else {
        System.out.println("Not found DVD with ID " + idForSearch);
    }
} else if (searchType == 2) {
    // Tìm kiếm theo tiêu đề
    System.out.println("Enter the title of DVD you want to search: ");
    String titleForSearch = sc.nextLine();

    if (cart.search(titleForSearch)) {
        System.out.println("Found DVD with title \"" + titleForSearch + "\"");
        System.out.println(cart.getDiscByTitle(titleForSearch));
    } else {
        System.out.println("Not found DVD with title \"" + titleForSearch + "\"");
    }
} else {
    System.out.println("Invalid search type. Please enter 1 or 2.");
}

sc.close();
}
```

Figure 16 CartTest class

```

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=52274:C:\SET_UP\
The disc 'The Lion King' has been added.
The disc 'Star Wars' has been added.
The disc 'Aladin' has been added.
*****CART*****
Ordered Items:
1. DVD{Title='The Lion King', Category='Animation', Director='Roger Allers', Length=87, Price=19.95}
2. DVD{Title='Star Wars', Category='Science Fiction', Director='George Lucas', Length=87, Price=24.95}
3. DVD{Title='Aladin', Category='Animation', Director='null', Length=0, Price=18.99}
Total cost: 63.89$
*****
Choose search type (1: By ID, 2: By Title):
1
Enter the ID of DVD you want to search:
2
Found DVD with ID 2:
DVD{Title='Star Wars', Category='Science Fiction', Director='George Lucas', Length=87, Price=24.95}

Process finished with exit code 0

```

Figure 17 Result search by Id

```

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=52284:C:\SET_UP\
The disc 'The Lion King' has been added.
The disc 'Star Wars' has been added.
The disc 'Aladin' has been added.
*****CART*****
Ordered Items:
1. DVD{Title='The Lion King', Category='Animation', Director='Roger Allers', Length=87, Price=19.95}
2. DVD{Title='Star Wars', Category='Science Fiction', Director='George Lucas', Length=87, Price=24.95}
3. DVD{Title='Aladin', Category='Animation', Director='null', Length=0, Price=18.99}
Total cost: 63.89$
*****
Choose search type (1: By ID, 2: By Title):
2
Enter the title of DVD you want to search:
star wars
Found DVD with title "star wars":
DVD{Title='Star Wars', Category='Science Fiction', Director='George Lucas', Length=87, Price=24.95}

Process finished with exit code 0

```

Figure 18 Result search by Title

5. Implement the Store class

Code

```
package hust.soict.hedspi.aims.store;

import hust.soict.hedspi.aims.disc.DigitalVideoDisc;
//Trinh Huu An 20225593

public class Store { 3 usages new *
    private DigitalVideoDisc[] itemsInStore; // Mảng DVD có sẵn trong cửa hàng 8 usages
    private int numItems; // Số lượng DVD hiện có trong cửa hàng 9 usages

    public Store() { 1 usage new *
        itemsInStore = new DigitalVideoDisc[100]; // Giả sử cửa hàng có thể chứa tối đa 100 DVD
        numItems = 0;
    }

    // Phương thức thêm DVD vào cửa hàng
    public void addDVD(DigitalVideoDisc dvd) { 3 usages new *
        if (numItems < itemsInStore.length) {
            itemsInStore[numItems] = dvd;
            numItems++;
            System.out.println("DVD added: " + dvd.getTitle());
        } else {
            System.out.println("Store is full, cannot add more DVDs.");
        }
    }

    // Phương thức xóa DVD khỏi cửa hàng
    public void removeDVD(DigitalVideoDisc dvd) { 1 usage new *
        for (int i = 0; i < numItems; i++) {
            if (itemsInStore[i].getId() == dvd.getId()) { // So sánh theo ID
                for (int j = i; j < numItems - 1; j++) {
                    itemsInStore[j] = itemsInStore[j + 1]; // Dịch các phần tử về trước
                }
                itemsInStore[numItems - 1] = null; // Xóa phần tử cuối cùng
                numItems--;
                System.out.println("DVD removed: " + dvd.getTitle());
            }
        }
        return;
    }

    System.out.println("DVD not found in the store.");
}

// Phương thức in ra danh sách các DVD trong cửa hàng
public void printStore() { 2 usages new *
    System.out.println("***** Store Inventory *****");
    for (int i = 0; i < numItems; i++) {
        System.out.println((i + 1) + ". " + itemsInStore[i]);
    }
    System.out.println("*****");
}
}
```

Figure 19 Store Class

```

package hust.soict.hedspi.aims.test.store;

import hust.soict.hedspi.aims.store.Store;
import hust.soict.hedspi.aims.disc.DigitalVideoDisc;
//Trinh Huu An 20225593
public class StoreTest {
    public static void main(String[] args) {
        // Tạo cửa hàng và các DVD
        Store store = new Store();

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

        // Thêm DVD vào cửa hàng
        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);

        // In ra các DVD trong cửa hàng
        store.printStore();

        // Xóa một DVD khỏi cửa hàng
        store.removeDVD(dvd2);

        // In ra các DVD trong cửa hàng sau khi xóa
        store.printStore();
    }
}

```

Figure 20 StoreTest class

Result

```

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=52722:C:\SET_UP\IntelliJ
DVD added: The Lion King
DVD added: Star Wars
DVD added: Aladin
***** Store Inventory *****
1. DVD{Title='The Lion King', Category='Animation', Director='Roger Allers', Length=87, Price=19.95}
2. DVD{Title='Star Wars', Category='Science Fiction', Director='George Lucas', Length=87, Price=24.95}
3. DVD{Title='Aladin', Category='Animation', Director='null', Length=0, Price=18.99}
*****
DVD removed: Star Wars
***** Store Inventory *****
1. DVD{Title='The Lion King', Category='Animation', Director='Roger Allers', Length=87, Price=19.95}
2. DVD{Title='Aladin', Category='Animation', Director='null', Length=0, Price=18.99}
*****
Process finished with exit code 0

```

Figure 21 Result

6. String, StringBuilder and StringBuffer

Code

```

package hust.soict.hedspi.garbage;

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);

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

```

Run ConcatenationInLoops

```

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024
1316
0
16
Process finished with exit code 0

```

Figure 22 So sánh String, StringBuilder, StringBuffer

Nhận xét:

String: Tốn nhiều thời gian nhất vì phải tạo và copy lại chuỗi mới ở mỗi vòng lặp.

StringBuilder: Nhanh nhất vì sử dụng vùng nhớ linh hoạt, không cần đồng bộ hóa.

StringBuffer: Nhanh, nhưng chậm hơn StringBuilder vì có thêm cơ chế đồng bộ hóa

Result

```

package hust.soict.hedspi.garbage;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
//Trinh Huu An 20225593

public class GarbageCreator { new *
    public static void main(String[] args) throws IOException { new *
        String filename = "C:/CODE/IT3103.744527.2024.1.20225593.TrinhHuuAn/OtherProjects/src/hust/soict/hedspi/garbage/test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;
        inputBytes = Files.readAllBytes(Paths.get(filename));
        startTime = System.currentTimeMillis();
        String outputString = "";
        for (byte b : inputBytes) {
            outputString += (char) b;
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}

```

Figure 23 GarbageCreator class

```

public class GarbageCreator { new *
    public static void main(String[] args) throws IOException { new *
        String filename = "C:/CODE/IT3103.744527.2024.1.20225593.TrinhHuuAn/OtherProjects/src/hust/soict/hedspi/garbage/test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;
        inputBytes = Files.readAllBytes(Paths.get(filename));
    }
}

```

GarbageCreator x

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=55179:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\bin" -Dfile.encoding=UTF-8

Process finished with exit code 0

Figure 24 Result: program hangs

```
package hust.soict.hedspi.garbage;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

public class NoGarbage { new *
    //Trinh Huu An
    public static void main(String[] args) throws IOException { new *
        String filename = "C:/CODE/IT3103.744527.2024.1.20225593.TrinhHuuAn/OtherProjects/src/hust/soict/hedspi/garbage/test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;
        inputBytes = Files.readAllBytes(Paths.get(filename));
        startTime = System.currentTimeMillis();
        // using StringBuilder
        StringBuilder outputStringBuilder = new StringBuilder();
        for (byte b : inputBytes) {
            outputStringBuilder.append((char) b);
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

Figure 25 NoGarbage class

```
> public class NoGarbage { new *
    //Trinh Huu An
    public static void main(String[] args) throws IOException { new *
        String filename = "C:/CODE/IT3103.744527.2024.1.20225593.TrinhHuuAn/OtherProjects/src/hust/soict/hedspi/garbage/test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

C:\SET_UP\JDK\bin\java.exe "-javaagent:C:\SET_UP\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=55185:C:\SET_UP\IntelliJ IDEA
2

Process finished with exit code 0
```

Figure 26 Result: Program finishes quickly

7 Class Diagram

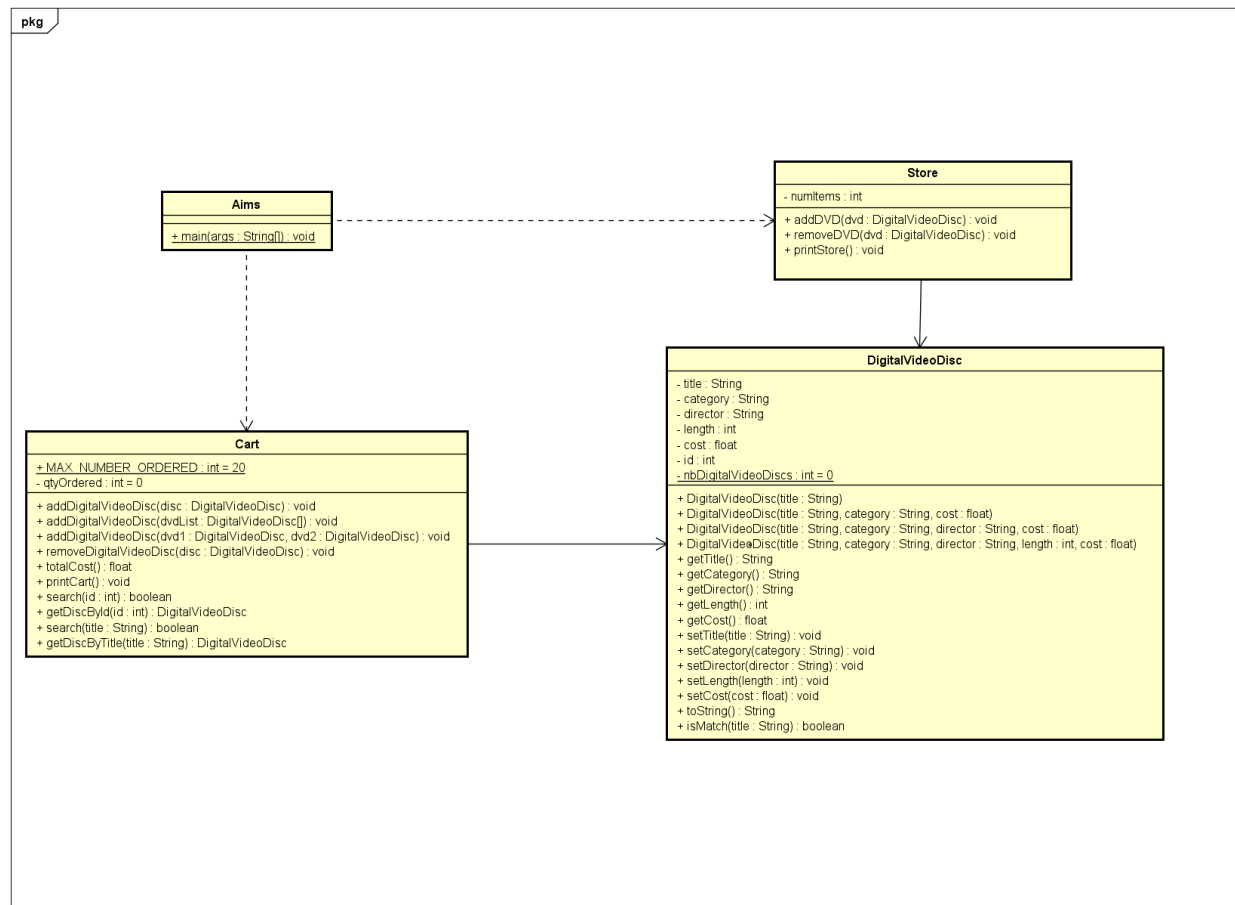


Figure 27 Class Diagram