

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH
IT3103-744528-2024.1

BÀI THỰC HÀNH - LAB03

Họ và tên: Nguyễn Đăng Phúc
Hưng

MSSV: 20226084

Lớp: IT-EP 01-K67

GVHD: Lê Thị Hoa

HTGD: Đặng Mạnh Cường

BÁO CÁO THỰC HÀNH LAB 03

THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. Working with method overloading.....	2
1.1. Overloading by differing types of parameter	2
Code	2
Result.....	2
1.2. Overloading by differing the number of parameters	3
Code	3
Result.....	5
2. Passing parameter	6
Code	6
Result.....	6
3. Class Member and Instance Member.....	7
Code	7
Result.....	9
4. Open the Cart class.....	9
Code	9
Result.....	11
5. Implement the Store class.....	12
Code	12
Result.....	13
6. String, StringBuilder and StringBuffer.....	14
Code	14
Result.....	14
7 Class Diagram	16
Table of Figures	17

1. Working with method overloading

1.1. Overloading by differing types of parameter

Code

```
public void addDigitalVideoDisc(DVD dvd) {
    if (quantityOrder >= MAX_NUMBER_ORDER) {
        System.out.println("Cart is full! Cannot add more DVDs.");
        return;
    }
    listDVDInCart.add(dvd);
    quantityOrder++;
    System.out.println("Added DVD '" + dvd.getName() + "' to cart.");
}

public void addDigitalVideoDisc(DVD[] dvdList) {
    if (dvdList == null) {
        return;
    }
    for (DVD dvd : dvdList) {
        addDigitalVideoDisc(dvd);
    }
}
```

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

```
public class Aims {
    Run | Debug
    public static void main(String[] args) {
        Cart anOrder = new Cart();

        DVD dvd1 = new DVD(name:"The Lion King", genre:"Animation", author:"Roger Allers", releaseDate:87, price:19.95f);
        anOrder.addDigitalVideoDisc(dvd1);

        DVD dvd2 = new DVD(name:"Star Wars", genre:"Science Fiction", author:"George Lucas", releaseDate:87, price:24.95f);
        anOrder.addDigitalVideoDisc(dvd2);

        DVD dvd3 = new DVD(name:"Aladdin", genre:"Animation", price:18.99f);
        anOrder.addDigitalVideoDisc(dvd3);

        System.out.println("Total Cost is: ");
        System.out.println(anOrder.totalCost());

        System.out.println("List before remove:");
        anOrder.print();

        anOrder.removeDigitalVideoDisc(dvdTitle:"Star Wars");

        System.out.println("\nList after remove:");
        anOrder.print();
    }
}
```

Figure 2 Aims Class

Result

```

Added DVD 'The Lion King' to cart.
Added DVD 'Star Wars' to cart.
Added DVD 'Aladdin' to cart.
Total Cost is:
63.89000129699707
List before remove:
List of DVDs in cart:
- The Lion King - Price: 19.950000762939453
- Star Wars - Price: 24.950000762939453
- Aladdin - Price: 18.989999771118164
Total cost: 63.89000129699707
Removed DVD 'Star Wars' from cart.

List after remove:
List of DVDs in cart:
- The Lion King - Price: 19.950000762939453
- Aladdin - Price: 18.989999771118164
Total cost: 38.94000053405762

```

Figure 3 Result

1.2. Overloading by differing the number of parameters

Code

```

public void addDigitalVideoDisc(DVD dvd) {
    if (quantityOrder >= MAX_NUMBER_ORDER) {
        System.out.println("Cart is full! Cannot add more DVDs.");
        return;
    }
    listDVDInCart.add(dvd);
    quantityOrder++;
    System.out.println("Added DVD '" + dvd.getName() + "' to cart.");
}

public void addDigitalVideoDisc(DVD[] dvdList) {
    if (dvdList == null) {
        return;
    }
    for (DVD dvd : dvdList) {
        addDigitalVideoDisc(dvd);
    }
}

public void addDigitalVideoDisc(DVD dvd1, DVD dvd2) {
    addDigitalVideoDisc(dvd1);
    addDigitalVideoDisc(dvd2);
}

```

Figure 4 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

```
public class Aims {  
    Run | Debug  
    public static void main(String[] args) {  
        Cart anOrder = new Cart();  
  
        DVD dvd1 = new DVD("The Lion King", "Animation", "Roger Allers", 87, 19.95f);  
        anOrder.addDigitalVideoDisc(dvd1);  
  
        DVD dvd2 = new DVD("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);  
        anOrder.addDigitalVideoDisc(dvd2);  
  
        DVD dvd3 = new DVD("Aladdin", "Animation", 18.99f);  
        anOrder.addDigitalVideoDisc(dvd3);  
  
        System.out.println("Total Cost is: ");  
        System.out.println(anOrder.totalCost());  
  
        System.out.println("List before remove:");  
        anOrder.print();  
  
        anOrder.removeDigitalVideoDisc(dvdTitle:"Star Wars");  
  
        System.out.println("\nList after remove:");  
        anOrder.print();  
    }  
}
```

Figure 5 Aims Class

Result

```
Added DVD 'The Lion King' to cart.  
Added DVD 'Star Wars' to cart.  
Added DVD 'Aladdin' to cart.  
Total Cost is:  
63.89000129699707  
List before remove:  
List of DVDs in cart:  
- The Lion King - Price: 19.950000762939453  
- Star Wars - Price: 24.950000762939453  
- Aladdin - Price: 18.989999771118164  
Total cost: 63.89000129699707  
Removed DVD 'Star Wars' from cart.  
  
List after remove:  
List of DVDs in cart:  
- The Lion King - Price: 19.950000762939453  
- Aladdin - Price: 18.989999771118164  
Total cost: 38.94000053405762
```

Figure 6 Result

2. Passing parameter

Code

```
package hust.soict.dsai.test.test;

import hust.soict.dsai.aims.disc.*;

public class TestingPassingParameter {
    public static void main(String[] args) {
        DVD jungleDVD = new DVD("Jungle");
        DVD cinderellaDVD = new DVD("Cinderella");
        swap(jungleDVD, cinderellaDVD);
        System.out.println("Jungle DVD title: " + jungleDVD.getName());
        System.out.println("Cinderella DVD title: " + cinderellaDVD.getName());

        changeName(jungleDVD, cinderellaDVD.getName());
        System.out.println("Jungle DVD title: " + jungleDVD.getName());
    }

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

    public static void changeName(DVD dvd, String name) {
        String old = dvd.getName();
        dvd.setName(name);
        dvd = new DVD(old);
    }
}
```

Figure 7 Passing parameter code

```
Jungle DVD title: Jungle
Cinderella DVD title: Cinderella
Jungle DVD title: Cinderella
```

Result

Figure 8 Passing parameter result

3. Class Member and Instance Member

Code

```
import java.util.List;

public class DVD {
    String name;
    String description;
    double price;
    int quantityInStock;
    int rating;
    String genre;
    int releaseDate;
    String category;
    String author;

    public DVD(String name, String genre, String author, int releaseDate, double price) {
        this.name = name;
        this.genre = genre;
        this.author = author;
        this.releaseDate = releaseDate;
        this.price = price;
    }

    public DVD(String name, String description, int price, int quantityInStock, int rating, String genre,
        int releaseDate) {
        this.name = name;
        this.description = description;
        this.price = price;
        this.quantityInStock = quantityInStock;
        this.rating = rating;
        this.genre = genre;
        this.releaseDate = releaseDate;
    }

    public DVD(String name, int price) {
        this.name = name;
        this.price = price;
    }
}
```



```
public DVD(String name, String genre, double price) {
    this.name = name;
    this.price = price;
    this.genre = genre;
}

public DVD(String name, String genre, int releaseDate) {
    this.name = name;
    this.genre = genre;
    this.releaseDate = releaseDate;
}

public String getName() {
    return name;
}

public String getDescription() {
    return description;
}

public double getPrice() {
    return price;
}

public int getQuantityInStock() {
    return quantityInStock;
}

public void updateQuantity(int quantity) {
    quantityInStock += quantity;
    System.out.println("Quantity of DVD '" + name + "' updated. Current quantity: " + quantityInStock);
}

public void checkAvailability() {
    if (quantityInStock > 0) {
        System.out.println("DVD '" + name + "' is currently available.");
    } else {
        System.out.println("DVD '" + name + "' is currently out of stock.");
    }
}

public void calculateAverageRating(List<Integer> ratings) {
    if (ratings.isEmpty()) {
        rating = 0;
        return;
    }
    int sum = 0;
    for (int r : ratings) {
        sum += r;
    }
    rating = sum / ratings.size();
    System.out.println("Average rating of DVD '" + name + "': " + rating);
}

public void displayDVDInfo() {
    System.out.println("DVD Information:");
    System.out.println("- Title: " + name);
    System.out.println("- Description: " + description);
    System.out.println("- Price: " + price);
    System.out.println("- Quantity in stock: " + quantityInStock);
    System.out.println("- Rating: " + rating);
    System.out.println("- Genre: " + genre);
    System.out.println("- Release year: " + releaseDate);
}

public String getGenre() {
    return genre;
}
```

Figure 9 Class member and Instance member

Result

Figure 10 Result

```
Added DVD 'The Lion King' to the store.  
Added DVD 'Star Wars' to the store.  
Added DVD 'Aladdin' to the store.  
Removed DVD 'Star Wars' from the store.  
DVD 'The Matrix' not found in the store.
```

4. Open the Cart class

Code

```
public void printCart() {  
    if (listDVDInCart.isEmpty()) {  
        System.out.println("Cart is empty.");  
        return;  
    }  
    System.out.println("List of DVDs in cart:");  
    for (DVD dvd : listDVDInCart) {  
        System.out.println("- " + dvd.getName() + " - Price: " + dvd.getPrice());  
    }  
    System.out.println("Total cost: " + totalCost());  
}
```

Figure 11 Code method to print the content in Cart

```
public void searchById(int dvdId) {
    for (DVD dvd : listDVDInCart) {
        if (dvd.getId() == dvdId) {
            System.out.println("DVD found:");
            dvd.displayDVDInfo();
            return;
        }
    }
    System.out.println("No DVD found with ID: " + dvdId);
}

public void searchByTitle(String dvdTitle) {
    for (DVD dvd : listDVDInCart) {
        if (dvd.getName().equals(dvdTitle)) {
            System.out.println("DVD Information:");
            dvd.displayDVDInfo();
            return;
        }
    }
    System.out.println("DVD '" + dvdTitle + "' not found in cart.");
}
}
```

*Figure 12 Code method search by id**Figure 13 Code method search by title*

```
package hust.soict.dsai.test.cart;

import java.util.*;

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

        DVD dvd1 = new DVD("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        cart.addDigitalVideoDisc(dvd1);

        DVD dvd2 = new DVD("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
        cart.addDigitalVideoDisc(dvd2);

        DVD dvd3 = new DVD("Aladdin", "Animation", "John Musker, Ron Clements", 90, 18.99f);
        cart.addDigitalVideoDisc(dvd3);

        cart.printCart();
    }
}
```

Figure 15 CartTest class

Result

```
Added DVD 'The Lion King' to cart.
Added DVD 'Star Wars' to cart.
Added DVD 'Aladdin' to cart.
List of DVDs in cart:
- The Lion King - Price: 19.950000762939453
- Star Wars - Price: 24.950000762939453
- Aladdin - Price: 18.989999771118164
Total cost: 63.89000129699707
```

Figure 16 Result

5. Implement the Store class

Code

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

import java.util.*;
import hust.soict.dsai.aims.disc.*;

public class Store {
    private DVD[] itemsInStore;
    private int numberOfItems;

    public Store(int capacity) {
        itemsInStore = new DVD[capacity];
        numberOfItems = 0;
    }

    public void addDVD(DVD dvd) {
        if (numberOfItems < itemsInStore.length) {
            itemsInStore[numberOfItems] = dvd;
            numberOfItems++;
            System.out.println("Added DVD '" + dvd.getName() + "' to the store.");
        } else {
            System.out.println("Store is full. Cannot add more DVDs.");
        }
    }

    public void removeDVD(String dvdTitle) {
        for (int i = 0; i < numberOfItems; i++) {
            if (itemsInStore[i].getName().equals(dvdTitle)) {
                for (int j = i; j < numberOfItems - 1; j++) {
                    itemsInStore[j] = itemsInStore[j + 1];
                }
                itemsInStore[numberOfItems - 1] = null;
                numberOfItems--;
                System.out.println("Removed DVD '" + dvdTitle + "' from the store.");
                return;
            }
        }
        System.out.println("DVD '" + dvdTitle + "' not found in the store.");
    }
}
```

F
i
g
u
r
e
1
7
S
t
o
r
e
C
l
a
s
s

```
package hust.soict.dsai.test.store;

import hust.soict.dsai.aims.disc.DVD;
import hust.soict.dsai.aims.store.Store;

public class StoreTest {
    public static void main(String[] args) {
        Store store = new Store(5);

        DVD dvd1 = new DVD("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        DVD dvd2 = new DVD("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
        DVD dvd3 = new DVD("Aladdin", "Animation", "John Musker, Ron Clements", 90, 18.99f);

        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);

        store.removeDVD("Star Wars");

        store.removeDVD("The Matrix");
    }
}
```

Figure 18 StoreTest class

Result

```
Added DVD 'The Lion King' to the store.
Added DVD 'Star Wars' to the store.
Added DVD 'Aladdin' to the store.
Removed DVD 'Star Wars' from the store.
DVD 'The Matrix' not found in the store.
```

Figure 19 Result

6. String, StringBuilder and StringBuffer

Code

```
package hust.soict.dsai.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);
    }
}
```

Figure 20 So sánh String và StringBuilder

Result

```
package hust.soict.dsai.garbage;

import java.io.IOException;

public class GarbageCreator {

    public static void main(String[] args) throws IOException {
        String filename = "test.exe";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        inputBytes = Files.readAllBytes(Paths.get(filename));
        startTime = System.currentTimeMillis();

        //      String outputString = "";
        StringBuilder outputString = new StringBuilder();
        for (byte b : inputBytes) {
            //      outputString += (char) b;
            outputString.append((char) b);
        }

        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

Figure 21 GarbageCreator class

*Figure 22 Result: program hangs*

```
package hust.soict.dsai.garbage;

import java.io.IOException;

public class NoGarbage {

    public static void main(String[] args) throws IOException {
        String filename = "test.exe";
        byte[] inputBytes = Files.readAllBytes(Paths.get(filename));
        long startTime, endTime;

        startTime = System.currentTimeMillis();

        StringBuilder sb = new StringBuilder();
        for (byte b : inputBytes) {
            sb.append((char) b);
        }
        String outputString = sb.toString();

        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

Figure 23 NoGarbage class

```
Program took 62ms to execute.
```

Figure 24 Result: Program finishes quickly

7 Class Diagram

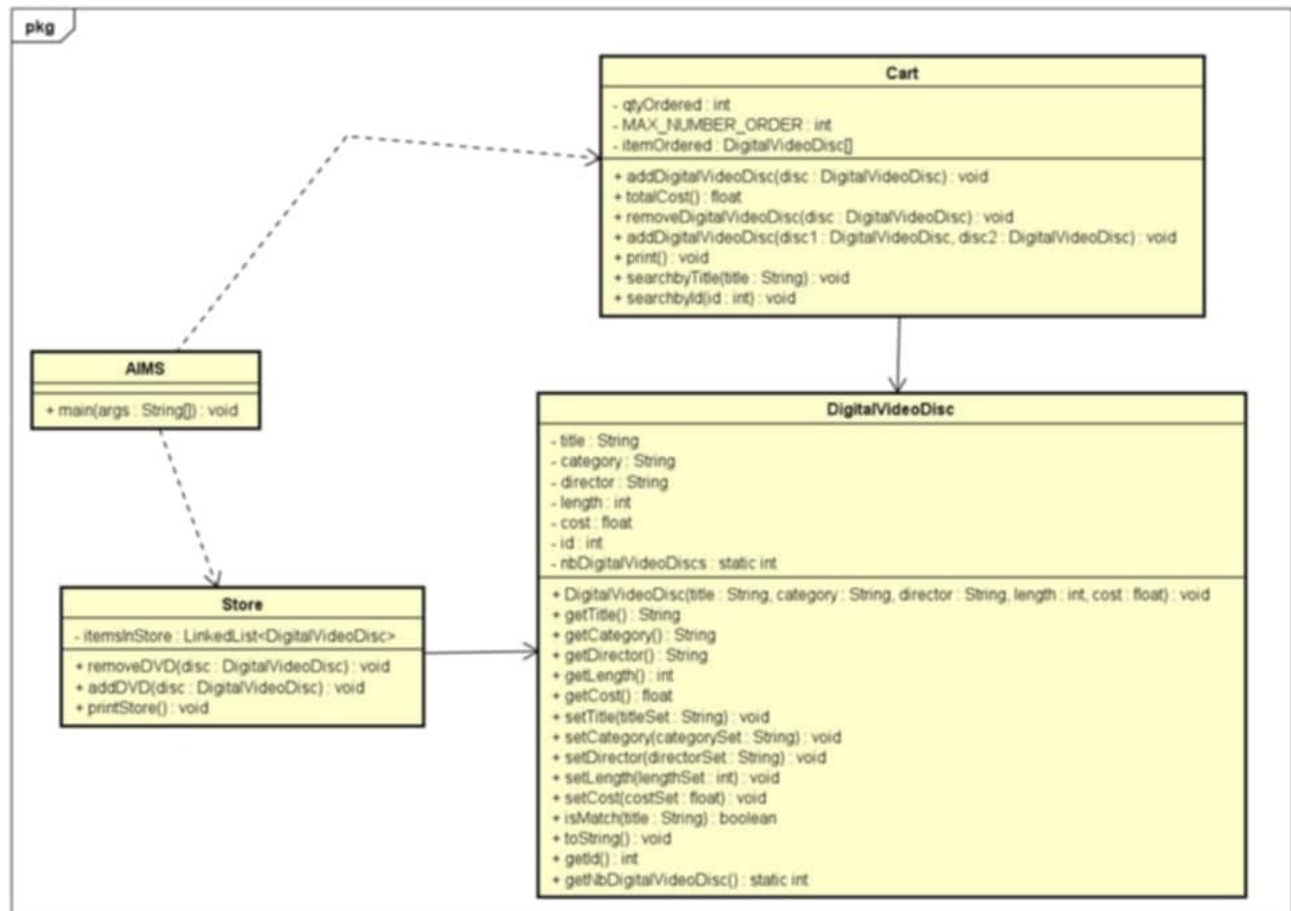


Figure 25 Class Diagram

Table of Figures

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