

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

**BÁO CÁO THỰC HÀNH**  
**IT3103-744530-2024.1**  
**BÀI THỰC HÀNH OOP LAB**

Họ và tên sv: Nguyễn Thanh Hưng

MSSV: 20225633

Lớp: Việt Nhật 03 K67

GVHD: Lê Thị Hoa

HTGD: Bùi Trọng Dũng

Hà Nội 11/2024

## Table of Contents

1. Working with method overloading .....	4
1.1. Overloading by differing types of parameter .....	4
1.2. Overloading by differing the number of parameters .....	5
1.3. Explain .....	5
2. Passing parameter .....	6
2.1. Question: <i>Is JAVA a Pass by Value or a Pass by Reference programming language?</i> ....	6
2.2. Trước khi sửa lại Swap .....	6
2.3. Cách giải quyết .....	8
3. Classifier Member and Instance Member .....	8
4. Open the Cart class .....	10
5. Implement the Store class.....	13
6. String, StringBuilder and StringBuffer .....	15
7. Class Diagram.....	17

## Table of Figures

<u>Figure 1: Method add Digital Video Disc (dvdList)</u> .....	4
<u>Figure 2: Result of method add Digital Video Disc (dvdList)</u> .....	4
<u>Figure 3: Method add Digital Video Disc (discs)</u> .....	4
<u>Figure 4: Result of method add Digital Video Disc (discs)</u> .....	5
<u>Figure 5: Method add Digital Video Disc (dvd1, dvd2)</u> .....	5
<u>Figure 6: Result of method add Digital Video Disc (dvd1, dvd2)</u> .....	5
<u>Figure 7: Passing parameter code</u> .....	7
<u>Figure 8: Result of passing parameter code</u> .....	7
<u>Figure 9: Swap code</u> .....	8
<u>Figure 10: Result of swap code</u> .....	8
<u>Figure 11: Class member and Instance member</u> .....	10
<u>Figure 12: Result of class member and Instance member</u> .....	10
<u>Figure 13: Code method to print the content in Cart</u> .....	11
<u>Figure 14: Code method to search by ID in Cart</u> .....	11
<u>Figure 15: Code method to search by Title in Cart</u> .....	11
<u>Figure 16: IsMatch() in DigitalVideoDisc class</u> .....	12
<u>Figure 17: CartTest class</u> .....	12
<u>Figure 17: CartTest class</u> .....	13
<u>Figure 18: Store class</u> .....	13
<u>Figure 19: StoreTest class</u> .....	14
<u>Figure 20: Result of Store</u> .....	15
<u>Figure 21: ConcatenationInLoops Class</u> .....	15
<u>Figure 22: Result of ConcatenationInLoops Class</u> .....	15
<u>Figure 23: GarbageCreator Class</u> .....	16
<u>Figure 24: NoGarbage Class</u> .....	16
<u>Figure 24: Class Diagram</u> .....	18

# BÁO CÁO THỰC HÀNH LAB03

## LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### 1. Working with method overloading

#### 1.1. Overloading by differing types of parameter

Code\_1: Phương thức nhận tham số là mảng DVD:

```
//Phương thức nhận tham số là mảng DVD
public void addDigitalVideoDisc1(DigitalVideoDisc_NTH[] dvdList) {
    for (DigitalVideoDisc_NTH disc : dvdList) {
        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered.add(disc);
            qtyOrdered++;
            System.out.println("The disc \"" + disc.getTitle() + "\" has been added to the cart.");
        } else {
            System.out.println("The cart is full. Cannot add more discs.");
            break;
        }
    }
}
```

Figure 1: Method add Digital Video Disc (dvdList)

Result\_1:

```
--- Thêm DVD bằng mảng ---
The disc "The Matrix" has been added to the cart.
The disc "Inception" has been added to the cart.
```

Figure 2: Result of method add Digital Video Disc (dvdList)

Code\_2: Phương thức nhận tham số là danh sách số lượng tùy ý (varargs):

```
//Phương thức nhận tham số là danh sách số lượng tùy ý (varargs)
public void addDigitalVideoDisc2(DigitalVideoDisc_NTH... discs) {
    for (DigitalVideoDisc_NTH disc : discs) {
        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered.add(disc);
            qtyOrdered++;
            System.out.println("The disc \"" + disc.getTitle() + "\" has been added to the cart.");
        } else {
            System.out.println("The cart is full. Cannot add more discs.");
            break;
        }
    }
}
```

Figure 3: Method add Digital Video Disc (discs)

## Result\_2:

```
--- Thêm DVD bằng varargs ---
The disc "Interstellar" has been added to the cart.
The disc "Finding Nemo" has been added to the cart.
```

Figure 4: Result of method add Digital Video Disc (discs)

## 1.2. Overloading by differing the number of parameters

### Code:

```
//Nạp chồng phương thức với số lượng tham số khác nhau
public void addDigitalVideoDisc(DigitalVideoDisc_NTH dvd1, DigitalVideoDisc_NTH dvd2) {
    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered.add(dvd1);
        qtyOrdered++;
        System.out.println("The disc \"" + dvd1.getTitle() + "\" has been added to the cart.");
    } else {
        System.out.println("The cart is full. Cannot add \"" + dvd1.getTitle() + "\".");
    }

    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered.add(dvd2);
        qtyOrdered++;
        System.out.println("The disc \"" + dvd2.getTitle() + "\" has been added to the cart.");
    } else {
        System.out.println("The cart is full. Cannot add \"" + dvd2.getTitle() + "\".");
    }
}
```

Figure 5: Method add Digital Video Disc (dvd1, dvd2)

### Result:

```
--- Thêm DVD bằng phương thức 2 tham số ---
The disc "The Matrix" has been added to the cart.
The disc "Interstellar" has been added to the cart.

--- Hiển thị giỏ hàng ---

--- Cart Contents ---
[The Matrix | Sci-Fi | $19.99] [Inception | Sci-Fi | $24.99] [Interstellar | Sci-Fi | $29.99] [Finding Nemo | Animation | $15.99] [The Matrix | Sci-Fi | $19.99] [Interstellar | Sci-Fi | $29.99]
-----
Total cost: $140.94
Total items in cart: 6
```

Figure 6: Result of method add Digital Video Disc (dvd1, dvd2)

## 1.3. Explain

### So sánh giữa mảng và varargs:

- **Mảng:** Cần thiết khi bạn đã có một mảng các đối tượng DVD từ trước.

- **Varargs:** Linh hoạt hơn, giúp người dùng truyền bao nhiêu tham số tùy ý mà không cần tạo mảng trước. **Ưu tiên sử dụng varargs** vì tính tiện lợi và dễ đọc.

## 2. Passing parameter

### 2.1. Question: *Is JAVA a Pass by Value or a Pass by Reference programming language?*

- **Pass by Value:** Giá trị của tham số được sao chép sang một biến khác, và bản sao đó được truyền vào phương thức. Vì vậy, bất kỳ thay đổi nào trong phương thức đều không ảnh hưởng đến giá trị gốc.
- **Pass by Reference:** Tham chiếu đến tham số thực tế được truyền vào phương thức, nên bất kỳ thay đổi nào trong phương thức sẽ ảnh hưởng trực tiếp đến tham số gốc.

### 2.2. Trước khi sửa lại Swap

Code:

```

1 public class TestPassingParameter {
2     public static void main(String[] args) {
3         DigitalVideoDisc_NTH jungleDVD = new DigitalVideoDisc_NTH("Jungle");
4         DigitalVideoDisc_NTH cinderellaDVD = new DigitalVideoDisc_NTH("Cinderella");
5
6         System.out.println("Before swapping:");
7         System.out.println("jungleDVD title: " + jungleDVD.getTitle());
8         System.out.println("cinderellaDVD title: " + cinderellaDVD.getTitle());
9
10        swap(jungleDVD, cinderellaDVD);
11
12        System.out.println("After swapping:");
13        System.out.println("jungleDVD title: " + jungleDVD.getTitle());
14        System.out.println("cinderellaDVD title: " + cinderellaDVD.getTitle());
15
16        changeTitle(jungleDVD, cinderellaDVD.getTitle());
17        System.out.println("After changing title:");
18        System.out.println("jungleDVD title: " + jungleDVD.getTitle());
19    }
20
21    public static void swap(DigitalVideoDisc_NTH dvd1, DigitalVideoDisc_NTH dvd2) {
22        DigitalVideoDisc_NTH temp = dvd1;
23        dvd1 = dvd2;
24        dvd2 = temp;
25    }
26
27    public static void changeTitle(DigitalVideoDisc_NTH dvd, String title) {
28        dvd.setTitle(title);
29    }
30 }
31

```

Figure 7: Passing parameter code

Result:

```

<terminated> TestPassingParameter [Java Application] C:\U:
Before swapping:
jungleDVD title: Jungle
cinderellaDVD title: Cinderella
After swapping:
jungleDVD title: Jungle
cinderellaDVD title: Cinderella
After changing title:
jungleDVD title: Cinderella

```

Figure 8: Result of passing parameter code

- After the call of `swap(jungleDVD, cinderellaDVD)` why does the title of these two objects still remain?  
 ⇒ Java truyền tham số theo giá trị (Pass by Value). Ở đây, giá trị là tham chiếu (reference) đến đối tượng. Khi thực hiện `swap()`, các tham chiếu được hoán đổi cục bộ trong phương thức, nhưng không ảnh hưởng đến tham chiếu gốc bên ngoài.

- After the call of `changeTitle(jungleDVD, cinderellaDVD.getTitle())` why is the title of the JungleDVD changed?
  - ⇒ Khi truyền tham số jungleDVD vào phương thức changeTitle, giá trị của tham chiếu đến đối tượng được truyền. Mặc dù tham số này được sao chép, phương thức có thể thay đổi nội dung của đối tượng mà tham chiếu trỏ đến (ở đây là thay đổi title của jungleDVD).

### 2.3. Cách giải quyết

Sử dụng một mảng để hoán đổi tham chiếu thực tế.

Code:

```
public static void swap(DigitalVideoDisc_NTH[] dvds) {
    if (dvds.length == 2) {
        DigitalVideoDisc_NTH temp = dvds[0];
        dvds[0] = dvds[1];
        dvds[1] = temp;
    }
}
```

Figure 9: Swap code

Result:

```
<terminated> testPassingParameter [Java Applicatio
Before swapping:
jungleDVD title: Jungle
cinderellaDVD title: Cinderella

After swapping:
jungleDVD title: Cinderella
cinderellaDVD title: Jungle

After changing title:
jungleDVD title: Jungle
```

Figure 10: Result of swap code

- ⇒ Java là ngôn ngữ Pass by Value, ngay cả khi làm việc với tham chiếu đối tượng. Tham chiếu được sao chép và truyền vào phương thức, nhưng không thay đổi tham chiếu gốc.
- ⇒ Để hoán đổi hai đối tượng thực sự, cần sử dụng cấu trúc bên ngoài (như mảng) để thay đổi tham chiếu thực tế.

## 3. Classifier Member and Instance Member



Code:

```

Aims_NTH.java  Cart_NTH.java  DigitalVideoDisc_NTH.java ×  TestPassingParameter.java  TestDigitalVideoDisc.java
1  import java.text.SimpleDateFormat;
2  import java.util.Date;
3
4  public class DigitalVideoDisc_NTH {
5      private String title;
6      private String category;
7      private String director;
8      private int length;
9      private float cost;
10     private String addDate;
11     private int id;
12     private static int nbDigitalVideoDiscs = 0;
13
14
15     public DigitalVideoDisc_NTH(String title) {
16         this.title = title;
17         this.id = ++nbDigitalVideoDiscs;
18         setAddDate();
19     }
20
21
22     public DigitalVideoDisc_NTH(String title, String category, float cost) {
23         this.title = title;
24         this.category = category;
25         this.cost = cost;
26         this.id = ++nbDigitalVideoDiscs;
27         setAddDate();
28     }
29
30     public DigitalVideoDisc_NTH(String title, String category, String director, float cost) {
31         this.title = title;
32         this.category = category;
33         this.director = director;
34         this.cost = cost;
35         this.id = ++nbDigitalVideoDiscs;
36         setAddDate();
37     }
38
39
79     public String getAddDate() {
80         return addDate;
81     }
82
83     public int getId() {
84         return id;
85     }
86
87     public static int getNbDigitalVideoDiscs() {
88         return nbDigitalVideoDiscs;
89     }
90
91     private void setAddDate() {
92         SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
93         addDate = formatter.format(new Date());
94     }
95
96     public void displayInfo() {
97         System.out.println("\nID: " + id);
98         System.out.println("Title: " + title);
99         System.out.println("Category: " + category);
100        System.out.println("Director: " + director);
101        System.out.println("Length: " + length + " minutes");
102        System.out.println("Cost: $" + cost);
103        System.out.println("Added Date: " + addDate);
104    }
105 }

```

Figure 11: Class member and Instance member

Result:

```
<terminated> testDigitalVideoDisc [Java Application]

--- DVD Information ---

ID: 1
Title: The Matrix
Category: Sci-Fi
Director: Wachowskis
Length: 120 minutes
Cost: $19.99
Added Date: 20/11/2024

ID: 2
Title: Inception
Category: Sci-Fi
Director: Christopher Nolan
Length: 148 minutes
Cost: $24.99
Added Date: 20/11/2024

ID: 3
Title: Interstellar
Category: Sci-Fi
Director: Christopher Nolan
Length: 169 minutes
Cost: $29.99
Added Date: 20/11/2024

Total DVDs created: 3
```

Figure 12: Result of class member and Instance member

#### 4. Open the Cart class

Code:

```
public void print() {
    System.out.println("*****CART*****");
    System.out.println("Ordered Items:");
    float totalCost = 0;
    int index = 1;
    for (DigitalVideoDisc_NTH disc : itemsOrdered) {
        System.out.println(index++ + ". " + disc.toString());
        totalCost += disc.getCost();
    }
    System.out.println("Total cost: " + totalCost + " $");
    System.out.println("*****");
}
```

Figure 13: Code method to print the content in Cart

```

public void searchById(int id) {
    boolean found = false;
    for (DigitalVideoDisc_NTH disc : itemsOrdered) {
        if (disc.getId() == id) {
            System.out.println("Found DVD with ID " + id + ":");
            System.out.println(disc.toString());
            found = true;
            break;
        }
    }
    if (!found) {
        System.out.println("No DVD found with ID " + id);
    }
}

```

Figure 14: Code method to search by ID in Cart

```

public void searchByTitle(String title) {
    boolean found = false;
    for (DigitalVideoDisc_NTH disc : itemsOrdered) {
        if (disc.isMatch(title)) {
            System.out.println("Found DVD with title \"" + title + "\":");
            System.out.println(disc.toString());
            found = true;
        }
    }
    if (!found) {
        System.out.println("No DVD found with title \"" + title + "\"");
    }
}

```

Figure 15: Code method to search by Title in Cart

```

public String toString() {
    return "DVD - " + title + " - " + category + " - " + director + " - " + length + ": " + cost + " $";
}

public boolean isMatch(String title) {
    return this.title.equalsIgnoreCase(title);
}

```

Figure 16: IsMatch() in DigitalVideoDisc class

```

1 public class CartTest {
2     public static void main(String[] args) {
3         // Tạo giỏ hàng
4         Cart_NTH cart = new Cart_NTH();
5
6         // Tạo các đối tượng DVD
7         DigitalVideoDisc_NTH dvd1 = new DigitalVideoDisc_NTH("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
8         DigitalVideoDisc_NTH dvd2 = new DigitalVideoDisc_NTH("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
9         DigitalVideoDisc_NTH dvd3 = new DigitalVideoDisc_NTH("Aladin", "Animation", 18.99f);
10
11        // Thêm DVD vào giỏ hàng
12        cart.addDigitalVideoDisc_NTH(dvd1);
13        cart.addDigitalVideoDisc_NTH(dvd2);
14        cart.addDigitalVideoDisc_NTH(dvd3);
15
16        // In giỏ hàng
17        System.out.println("\n--- Printing Cart ---");
18        cart.print();
19
20        // Tìm kiếm theo ID
21        System.out.println("\n--- Searching by ID ---");
22        cart.searchById(1); // Tìm kiếm ID 1
23        cart.searchById(5); // Tìm kiếm ID không tồn tại
24
25        // Tìm kiếm theo tiêu đề
26        System.out.println("\n--- Searching by Title ---");
27        cart.searchByTitle("Star Wars"); // Tiêu đề tồn tại
28        cart.searchByTitle("Avatar");   // Tiêu đề không tồn tại
29    }
30 }
31

```

Figure 17: CartTest class

Result:

```

terminated: CartTest [Java Application] C:\Users\PC\AppData\Local\Programs\org.eclipse.just4openjdk.
The disc "The Lion King" has been added to the cart.

The disc "Star Wars" has been added to the cart.

The disc "Aladin" has been added to the cart.

--- Printing Cart ---
*****CART*****
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.95 $
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95 $
3. DVD - Aladin - Animation - null - 0: 18.99 $
Total cost: 63.89 $
*****

--- Searching by ID ---
Found DVD with ID 1:
DVD - The Lion King - Animation - Roger Allers - 87: 19.95 $
No DVD found with ID 5

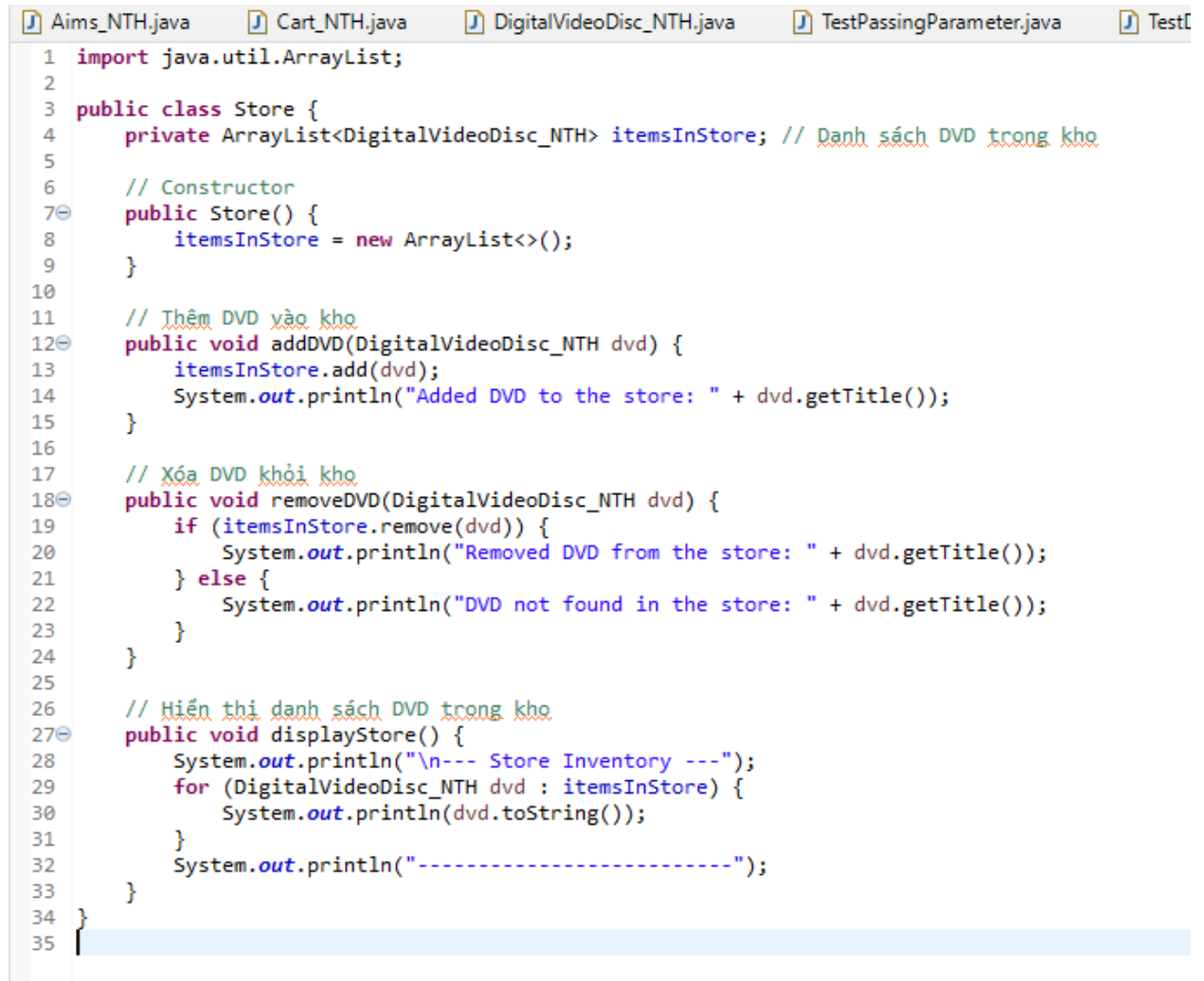
--- Searching by Title ---
Found DVD with title "Star Wars":
DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95 $
No DVD found with title "Avatar"

```

Figure 17: CartTest class

## 5. Implement the Store class

Code:

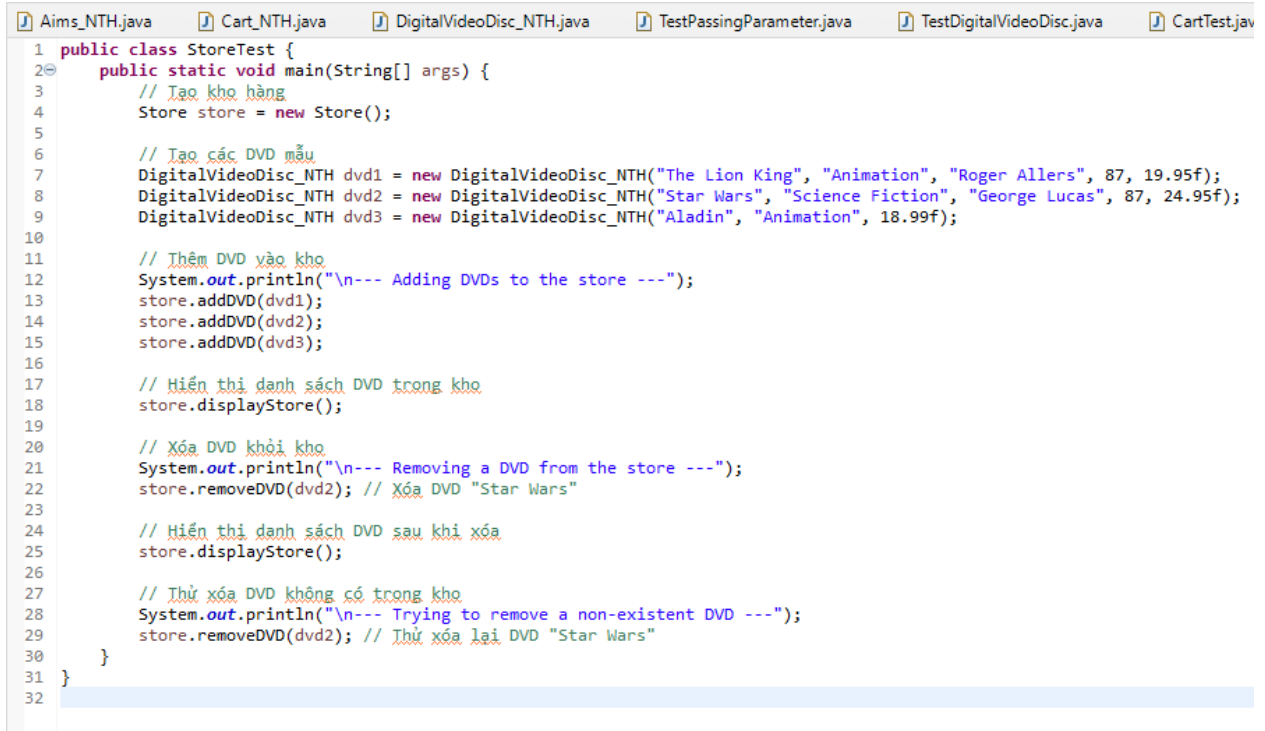


```

1  import java.util.ArrayList;
2
3  public class Store {
4      private ArrayList<DigitalVideoDisc_NTH> itemsInStore; // Danh sách DVD trong kho
5
6      // Constructor
7      public Store() {
8          itemsInStore = new ArrayList<>();
9      }
10
11     // Thêm DVD vào kho
12     public void addDVD(DigitalVideoDisc_NTH dvd) {
13         itemsInStore.add(dvd);
14         System.out.println("Added DVD to the store: " + dvd.getTitle());
15     }
16
17     // Xóa DVD khỏi kho
18     public void removeDVD(DigitalVideoDisc_NTH dvd) {
19         if (itemsInStore.remove(dvd)) {
20             System.out.println("Removed DVD from the store: " + dvd.getTitle());
21         } else {
22             System.out.println("DVD not found in the store: " + dvd.getTitle());
23         }
24     }
25
26     // Hiển thị danh sách DVD trong kho
27     public void displayStore() {
28         System.out.println("\n--- Store Inventory ---");
29         for (DigitalVideoDisc_NTH dvd : itemsInStore) {
30             System.out.println(dvd.toString());
31         }
32         System.out.println("-----");
33     }
34 }
35

```

Figure 18: Store class



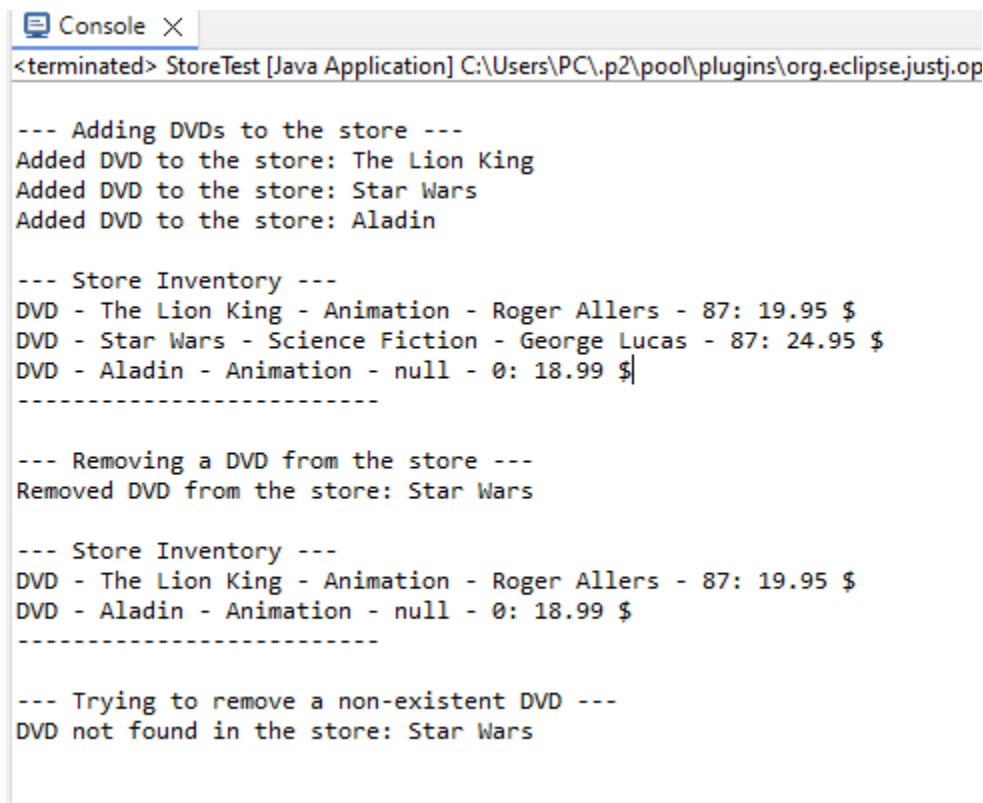
```

1 public class StoreTest {
2     public static void main(String[] args) {
3         // Tạo kho hàng
4         Store store = new Store();
5
6         // Tạo các DVD mẫu
7         DigitalVideoDisc_NTH dvd1 = new DigitalVideoDisc_NTH("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
8         DigitalVideoDisc_NTH dvd2 = new DigitalVideoDisc_NTH("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
9         DigitalVideoDisc_NTH dvd3 = new DigitalVideoDisc_NTH("Aladin", "Animation", 18.99f);
10
11        // Thêm DVD vào kho
12        System.out.println("\n--- Adding DVDs to the store ---");
13        store.addDVD(dvd1);
14        store.addDVD(dvd2);
15        store.addDVD(dvd3);
16
17        // Hiển thị danh sách DVD trong kho
18        store.displayStore();
19
20        // Xóa DVD khỏi kho
21        System.out.println("\n--- Removing a DVD from the store ---");
22        store.removeDVD(dvd2); // Xóa DVD "Star Wars"
23
24        // Hiển thị danh sách DVD sau khi xóa
25        store.displayStore();
26
27        // Thử xóa DVD không có trong kho
28        System.out.println("\n--- Trying to remove a non-existent DVD ---");
29        store.removeDVD(dvd2); // Thử xóa lại DVD "Star Wars"
30    }
31 }
32

```

Figure 19: StoreTest class

Result:



```

<terminated> StoreTest [Java Application] C:\Users\PC\.p2\pool\plugins\org.eclipse.justj.op

--- Adding DVDs to the store ---
Added DVD to the store: The Lion King
Added DVD to the store: Star Wars
Added DVD to the store: Aladin

--- Store Inventory ---
DVD - The Lion King - Animation - Roger Allers - 87: 19.95 $
DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95 $
DVD - Aladin - Animation - null - 0: 18.99 $
-----

--- Removing a DVD from the store ---
Removed DVD from the store: Star Wars

--- Store Inventory ---
DVD - The Lion King - Animation - Roger Allers - 87: 19.95 $
DVD - Aladin - Animation - null - 0: 18.99 $
-----

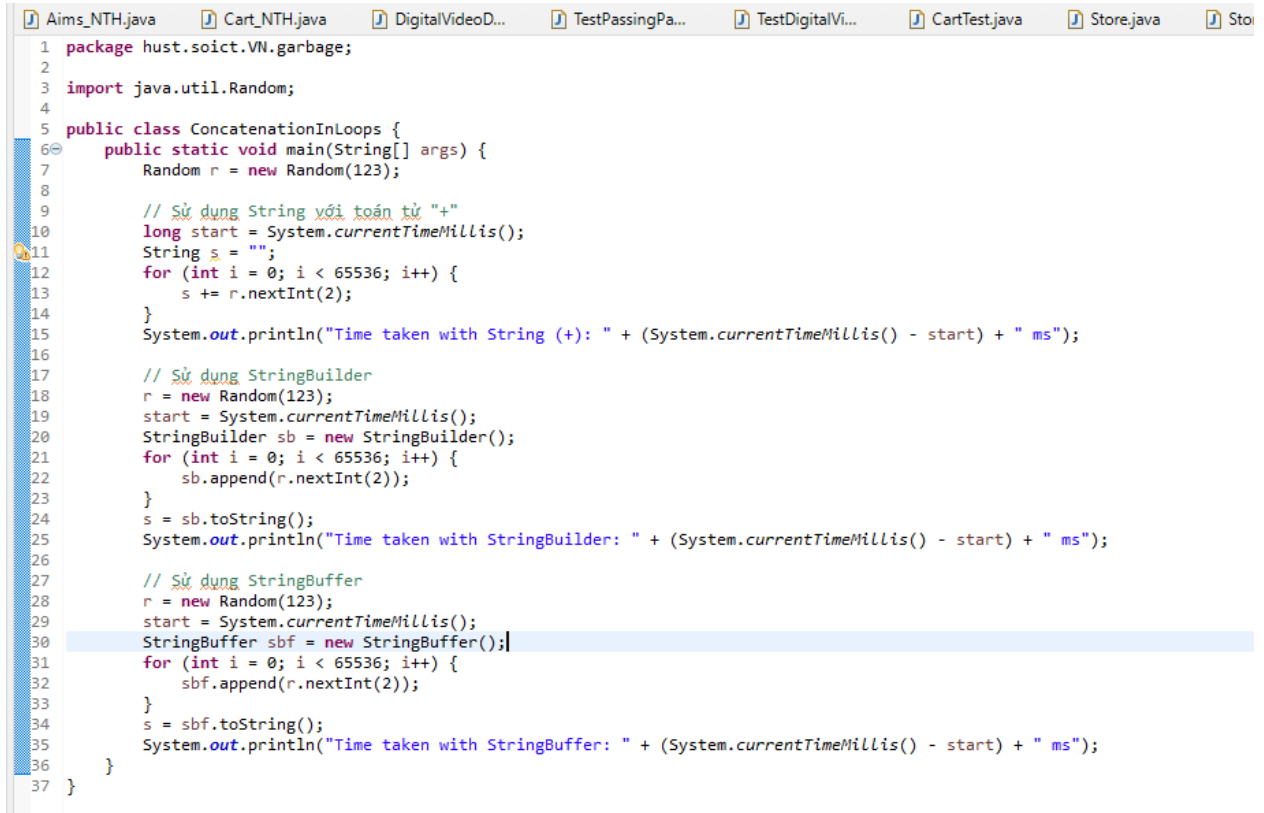
--- Trying to remove a non-existent DVD ---
DVD not found in the store: Star Wars

```

Figure 20: Result of Store

## 6. String, StringBuilder and StringBuffer

### Code: ConcatenationInLoops



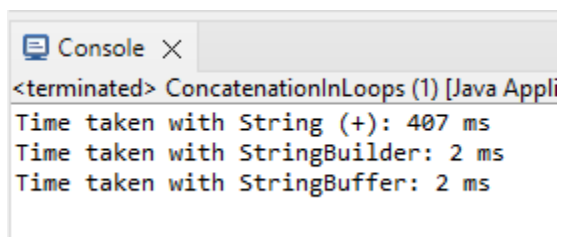
```

1 package hust.soict.VN.garbage;
2
3 import java.util.Random;
4
5 public class ConcatenationInLoops {
6     public static void main(String[] args) {
7         Random r = new Random(123);
8
9         // Sử dụng String với toán tử "+"
10        long start = System.currentTimeMillis();
11        String s = "";
12        for (int i = 0; i < 65536; i++) {
13            s += r.nextInt(2);
14        }
15        System.out.println("Time taken with String (+): " + (System.currentTimeMillis() - start) + " ms");
16
17        // Sử dụng StringBuilder
18        r = new Random(123);
19        start = System.currentTimeMillis();
20        StringBuilder sb = new StringBuilder();
21        for (int i = 0; i < 65536; i++) {
22            sb.append(r.nextInt(2));
23        }
24        s = sb.toString();
25        System.out.println("Time taken with StringBuilder: " + (System.currentTimeMillis() - start) + " ms");
26
27        // Sử dụng StringBuffer
28        r = new Random(123);
29        start = System.currentTimeMillis();
30        StringBuffer sbf = new StringBuffer();
31        for (int i = 0; i < 65536; i++) {
32            sbf.append(r.nextInt(2));
33        }
34        s = sbf.toString();
35        System.out.println("Time taken with StringBuffer: " + (System.currentTimeMillis() - start) + " ms");
36    }
37 }

```

Figure 21: ConcatenationInLoops Class

Result:



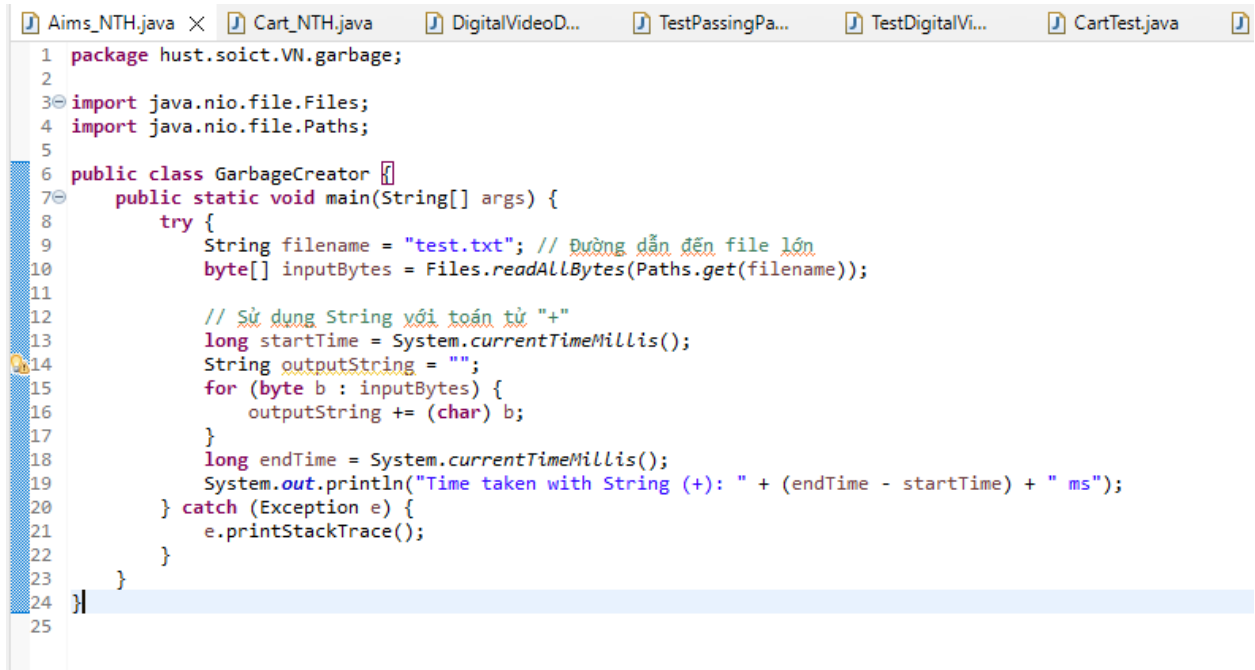
```

<terminated> ConcatenationInLoops (1) [Java Appli
Time taken with String (+): 407 ms
Time taken with StringBuilder: 2 ms
Time taken with StringBuffer: 2 ms

```

Figure 22: Result of ConcatenationInLoops Class

Code: GarbageCreator: tạo "rác" bằng cách đọc một file lớn và sử dụng *String* với toán tử + để nối chuỗi.



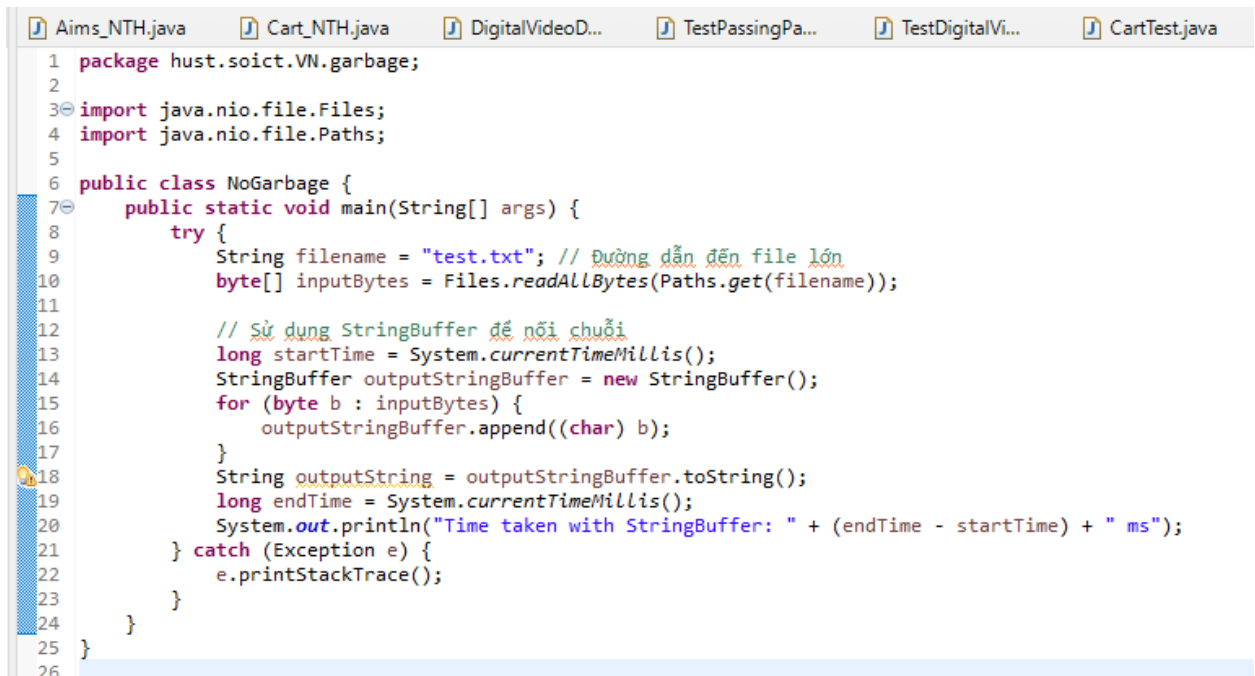
```

1 package hust.soict.VN.garbage;
2
3 import java.nio.file.Files;
4 import java.nio.file.Paths;
5
6 public class GarbageCreator {
7     public static void main(String[] args) {
8         try {
9             String filename = "test.txt"; // Đường dẫn đến file lớn
10            byte[] inputBytes = Files.readAllBytes(Paths.get(filename));
11
12            // Sử dụng String với toán tử "+"
13            long startTime = System.currentTimeMillis();
14            String outputString = "";
15            for (byte b : inputBytes) {
16                outputString += (char) b;
17            }
18            long endTime = System.currentTimeMillis();
19            System.out.println("Time taken with String (+): " + (endTime - startTime) + " ms");
20        } catch (Exception e) {
21            e.printStackTrace();
22        }
23    }
24 }
25

```

Figure 23: GarbageCreator Class

Code: NoGarbage: sử dụng *StringBuffer* thay vì *String* với toán tử + để cải thiện hiệu năng



```

1 package hust.soict.VN.garbage;
2
3 import java.nio.file.Files;
4 import java.nio.file.Paths;
5
6 public class NoGarbage {
7     public static void main(String[] args) {
8         try {
9             String filename = "test.txt"; // Đường dẫn đến file lớn
10            byte[] inputBytes = Files.readAllBytes(Paths.get(filename));
11
12            // Sử dụng StringBuffer để nối chuỗi
13            long startTime = System.currentTimeMillis();
14            StringBuffer outputStringBuffer = new StringBuffer();
15            for (byte b : inputBytes) {
16                outputStringBuffer.append((char) b);
17            }
18            String outputString = outputStringBuffer.toString();
19            long endTime = System.currentTimeMillis();
20            System.out.println("Time taken with StringBuffer: " + (endTime - startTime) + " ms");
21        } catch (Exception e) {
22            e.printStackTrace();
23        }
24    }
25 }
26

```

Figure 24: NoGarbage Class



Result:

Conclusion:

- So sánh:
  - *String* với toán tử + tạo nhiều đối tượng tạm thời, gây giảm hiệu năng.
  - *StringBuilder* và *StringBuffer* hiệu quả hơn, trong đó *StringBuilder* nhanh hơn do không hỗ trợ đồng bộ (*non-synchronized*).
- Ứng dụng:
  - Sử dụng *StringBuilder* trong hầu hết các trường hợp khi cần xử lý chuỗi lớn
  - Sử dụng *StringBuffer* nếu cần đồng bộ trong môi trường đa luồng.

## 7. Class Diagram

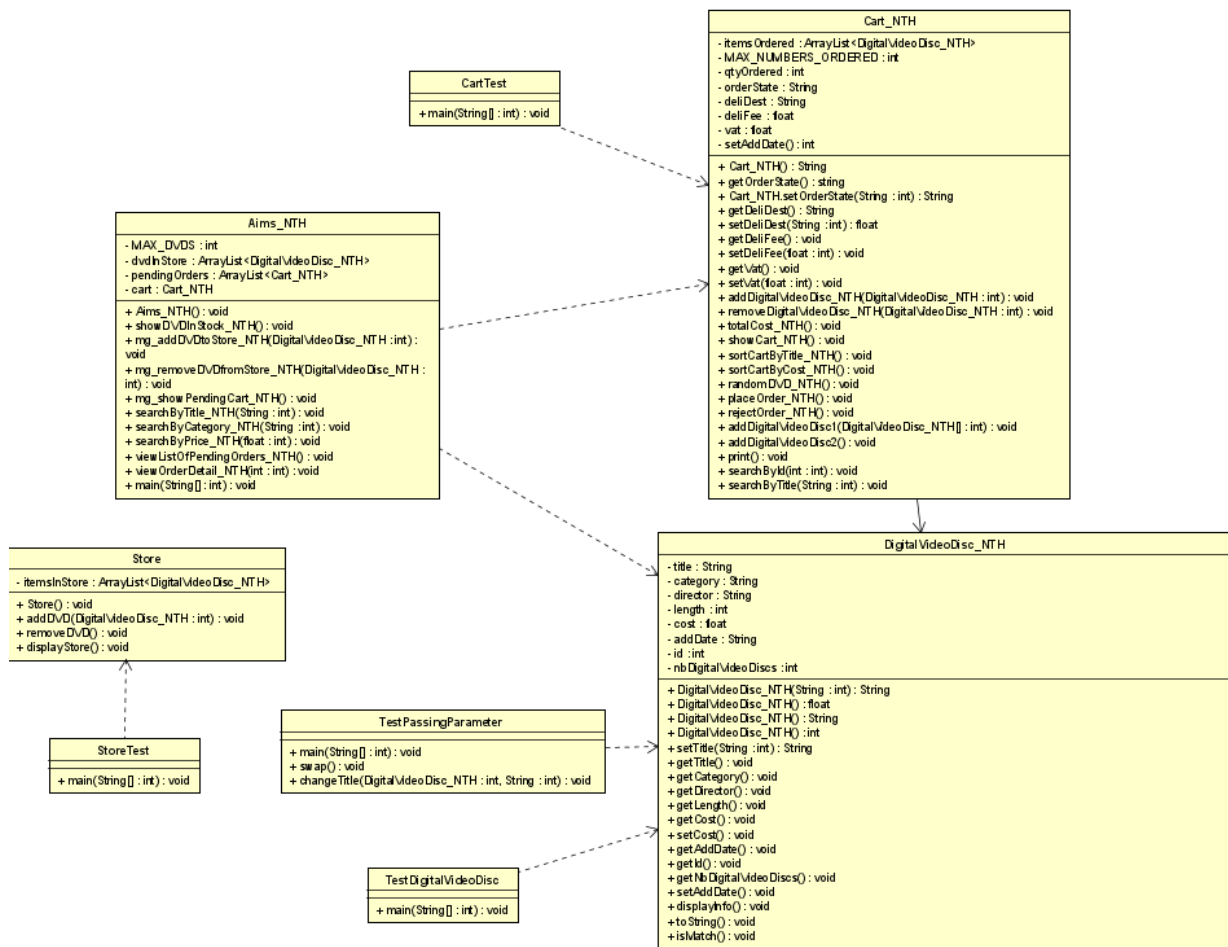


Figure 24: Class Diagram