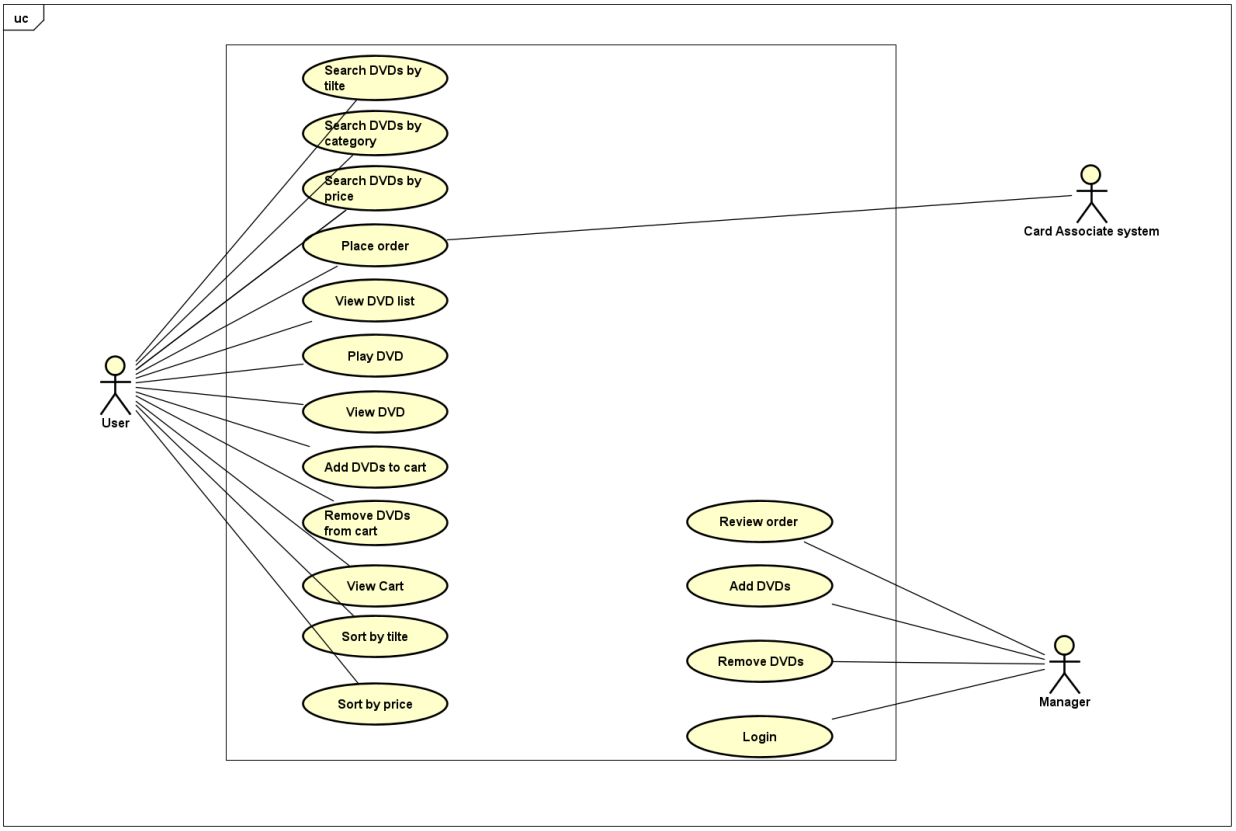
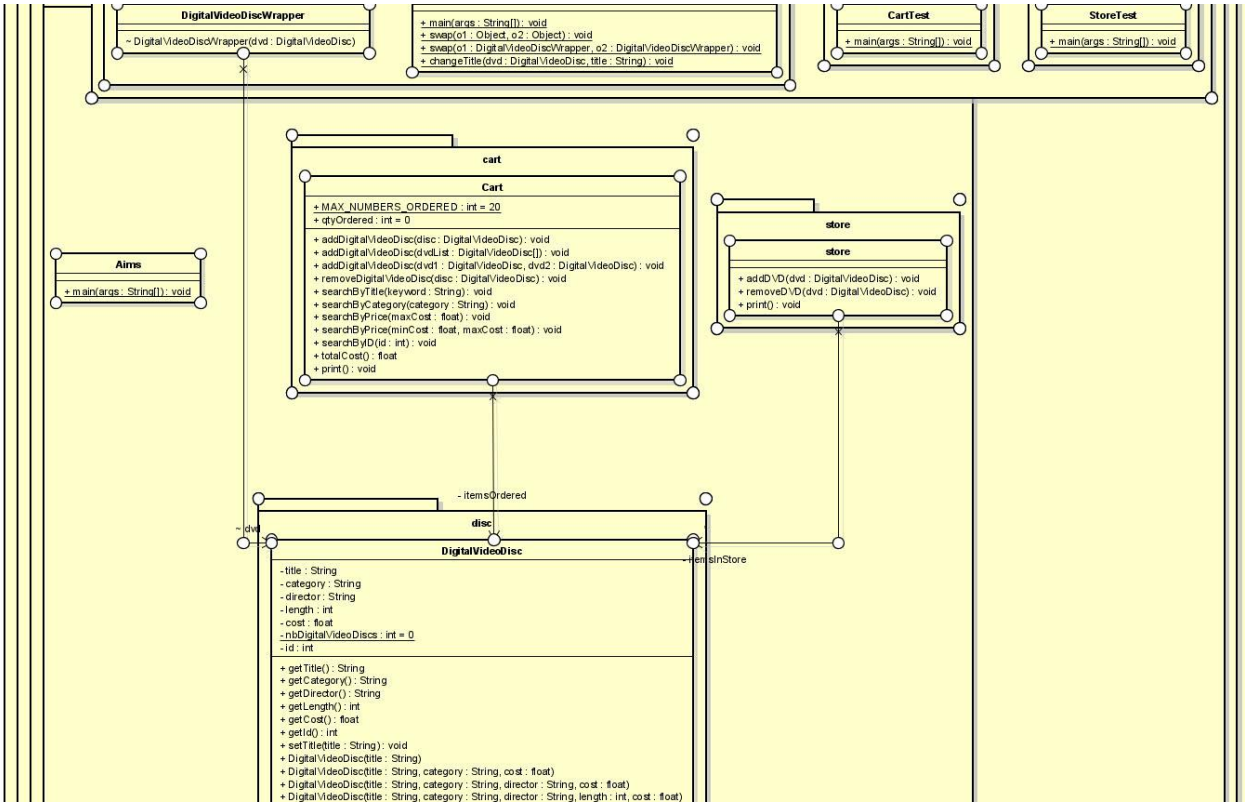
Phạm Quang Minh – 20235606




DigitalVideoDiscWrapper
~ DigitalVideoDiscWrapper(dvd : DigitalVideoDisc)
+ main(args : String[]) : void
+ swap(o1 : Object, o2 : Object) : void
+ swap(o1 : DigitalVideoDiscWrapper, o2 : DigitalVideoDiscWrapper) : void
+ changeTitle(dvd : DigitalVideoDisc, title : String) : void
CartTest
+ main(args : String[]) : void
StoreTest
+ main(args : String[]) : void
cart
Cart
+ MAX_NUMBERS_ORDERED : int = 20
+ qtyOrdered : int = 0
+ addDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ addDigitalVideoDisc(dvdList : DigitalVideoDisc[]) : void
+ addDigitalVideoDisc(dvd1 : DigitalVideoDisc, dvd2 : DigitalVideoDisc) : void
+ removeDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ searchByTitle(keyword : String) : void
+ searchByCategory(category : String) : void
+ searchByPrice(maxCost : float) : void
+ searchByPrice(minCost : float, maxCost : float) : void
+ searchByID(id : int) : void
+ totalCost() : float
+ print() : void
store
store
+ addDVD(dvd : DigitalVideoDisc) : void
+ removeDVD(dvd : DigitalVideoDisc) : void
+ print() : void
Aims
+ main(args : String[]) : void
- itemsOrdered
disc
DigitalVideoDisc
- title : String
- category : String
- director : String
- length : int
- cost : float
- nbDigitalVideoDiscs : int = 0
- id : int
+ getTitle() : String
+ getCategory() : String
+ getDirector() : String
+ getLength() : int
+ getCost() : float
+ getId() : int
+ setTitle(title : String) : void
+ DigitalVideoDisc(title : String)
+ DigitalVideoDisc(title : String, category : String, cost : float)
+ DigitalVideoDisc(title : String, category : String, director : String, cost : float)
+ DigitalVideoDisc(title : String, category : String, director : String, length : int, cost : float)
- dvd
- itemsInStore

2. Working with overloading

```java
public void addDigitalVideoDisc(DigitalVideoDisc[] dvdArray) {
    for (DigitalVideoDisc disc : dvdArray) {
        if (qtyOrdered == MAX_NUMBERS_ORDERED) {
            System.out.println(x:"The cart is almost full. Can't add more discs");
            break;
        } else {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The DVD " + '\"' + disc.getTitle() + '\"' + " has been added!");
        }
    }
}

public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    if (qtyOrdered + 1 >= MAX_NUMBERS_ORDERED) {
        System.out.println(x:"The cart is almost full. Can't add more discs");

    } else {
        itemsOrdered[qtyOrdered] = dvd1;
        qtyOrdered++;
        System.out.println("The DVD " + '\"' + dvd1.getTitle() + '\"' + " has been added!");

        itemsOrdered[qtyOrdered] = dvd2;
        qtyOrdered++;
        System.out.println("The DVD " + '\"' + dvd2.getTitle() + '\"' + " has been added!");

    }
}
```

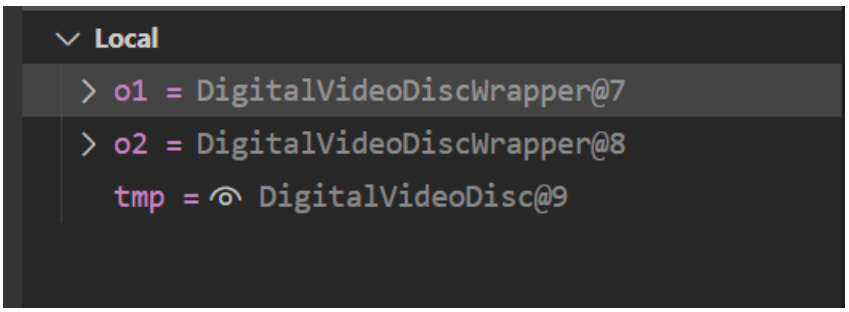
3

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

    DigitalVideoDiscWrapper wjungleDVD = new DigitalVideoDiscWrapper(jungleDVD);
    DigitalVideoDiscWrapper wcinderellaDVD = new DigitalVideoDiscWrapper(cinderellaDVD);

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

    // Correct swap() function
    swap(wjungleDVD, wcinderellaDVD);
    System.out.println("Correct swapped jungle dvd title: " + wjungleDVD.dvd.getTitle());
    System.out.println("Correct swapped cinderella dvd title: " + wcinderellaDVD.dvd.getTitle());
}
// Wrong swap() function
public static void swap(Object o1, Object o2) {
    Object tmp = o1;
    o1 = o2;
    o2 = tmp;
}

// Correct swap() function
public static void swap(DigitalVideoDiscWrapper o1, DigitalVideoDiscWrapper o2) {
    DigitalVideoDisc tmp = o1.dvd;
    o1.dvd = o2.dvd;
    o2.dvd = tmp;
}
```

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

- Because the swap() method is exchanging the values of the title fields between the two objects, but it's not changing the object references themselves

4.



```
args = String[0]@7
jungleDVD = DigitalVideoDisc@8
cinderellaDVD = DigitalVideoDisc@9
wjungleDVD = DigitalVideoDiscWrapper@10
  dvd = DigitalVideoDisc@9
wcinderellaDVD = DigitalVideoDiscWrapper@11
  dvd = DigitalVideoDisc@8
```

5. Classifier member and instance member

```java
package hust.soict.dsai.test.disc;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

public class DigitalVideoDiscWrapper {
    DigitalVideoDisc dvd;
    DigitalVideoDiscWrapper(DigitalVideoDisc dvd) {
        super();

        this.dvd = dvd;
    }
}
```

```java
package hust.soict.dsai.test.disc;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

public class TestPassingParameter {

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

        DigitalVideoDiscWrapper wjungleDVD = new DigitalVideoDiscWrapper(jungleDVD);
        DigitalVideoDiscWrapper wcinderellaDVD = new DigitalVideoDiscWrapper(cinderellaDVD);

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

        // Correct swap() function
        swap(wjungleDVD, wcinderellaDVD);
        System.out.println("Correct swapped jungle dvd title: " + wjungleDVD.dvd.getTitle());
        System.out.println("Correct swapped cinderella dvd title: " + wcinderellaDVD.dvd.getTitle());
    }
    // Wrong swap() function
    public static void swap(Object o1, Object o2) {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

    // Correct swap() function
    public static void swap(DigitalVideoDiscWrapper o1, DigitalVideoDiscWrapper o2) {
        DigitalVideoDisc tmp = o1.dvd;
        o1.dvd = o2.dvd;
        o2.dvd = tmp;
    }

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

6. Open Cart class

```java
package hust.soict.dsai.test.cart;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

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

        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger All
        cart.addDigitalVideoDisc(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George
        cart.addDigitalVideoDisc(dvd2);

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

        cart.print();
    }
}
```

7. Implement Store class

```java
package hust.soict.dsai.aims.store;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;
import java.util.List;
import java.util.ArrayList;

public class Store {

    private List<DigitalVideoDisc> itemsInStore = new ArrayList<DigitalVideoDisc>();

    public void addDVD(DigitalVideoDisc dvd) {
        itemsInStore.add(dvd);
        System.out.println(dvd.getTitle() + " has been added to the store.");
    }
    public void removeDVD(DigitalVideoDisc dvd) {
        boolean removed = itemsInStore.remove(dvd);
        if (removed) {
            System.out.println(dvd.getTitle() + " has been removed from the store.");
        } else {
            System.out.println(dvd.getTitle() + " is not found in the store.");
        }
    }

    public void printStore() {
        for (int i = 0; i < itemsInStore.size(); i++) {
            System.out.println(i+1 + ". " + itemsInStore.get(i));
        }
    }
}
```

9. String Concatenation

```java
public class ConcatenationInLoops {
    Run | Debug
    public static void main(String[] args) {
        Random r = new Random(seed:123);
        long start = System.currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i ++) {
            s += r.nextInt(bound:2);
        }
        System.out.println("Using + operator: " + (System.currentTimeMillis() - start) + "ms");

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

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

    }
```

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

public class NoGarbage {
    Run | Debug
    public static void main(String[] args) {
        StringBuilder garbageBuilder = new StringBuilder();
        long start = System.currentTimeMillis();

        for (int i = 0; i < Integer.MAX_VALUE; i++) {
            garbageBuilder.append(i); // Sử dụng StringBuilder
            if (i % 100000 == 0) { // Giới hạn bộ nhớ khi lặp
                garbageBuilder.setLength(newLength:0);
            }
        }

        long end = System.currentTimeMillis();
        System.out.println("Execution time: " + (end - start) + " ms");
    }
}
```