

1. Working with method overloading

1.1 Overloading by differing types of parameter

```
public void addDigitalVideoDisc(DigitalVideoDisc [] dvdlist){ no usages
    for (DigitalVideoDisc dvd : dvdlist) {
        itemsOrdered.add(dvd);
    }
}
```

1.2 Overloading by differing the number of parameters

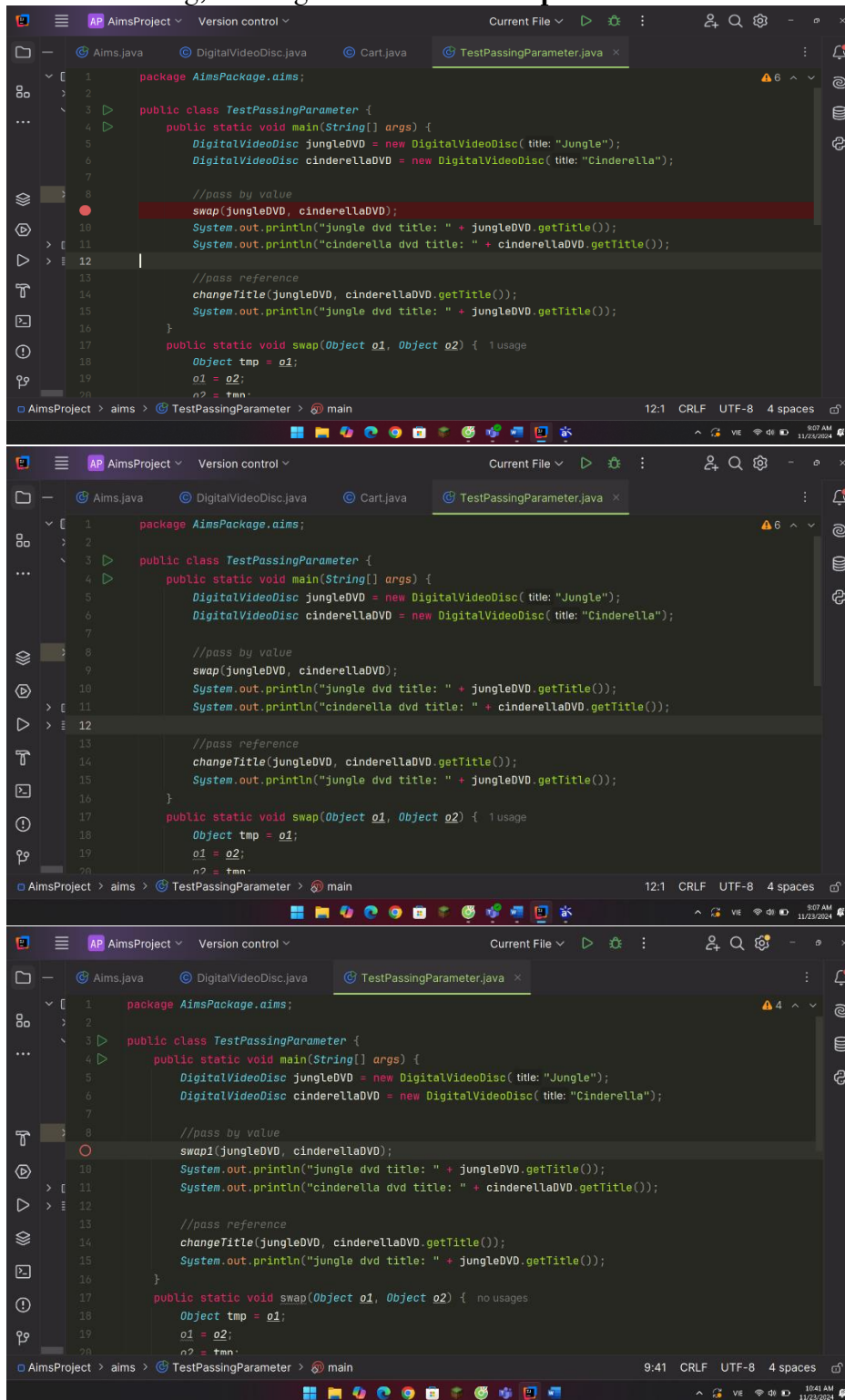
```
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2){ no usages
    addDigitalVideoDisc(dvd1);
    addDigitalVideoDisc(dvd2);
}
```

2. Passing parameter

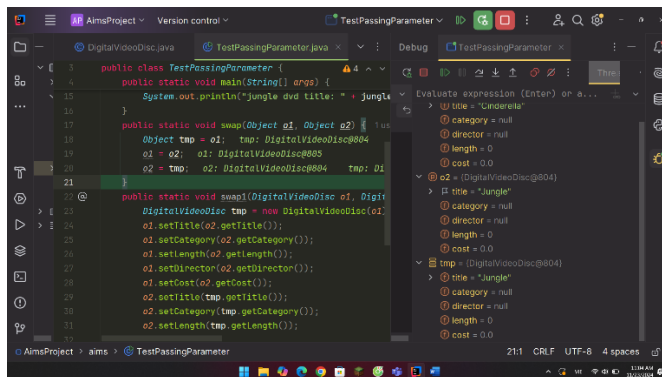
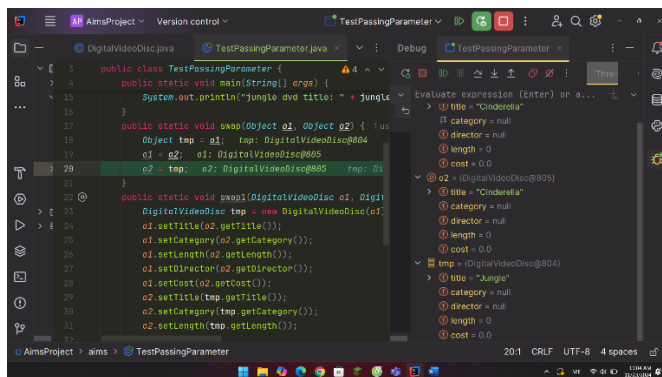
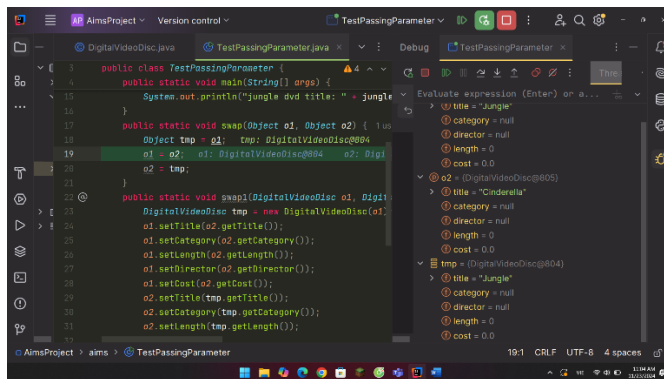
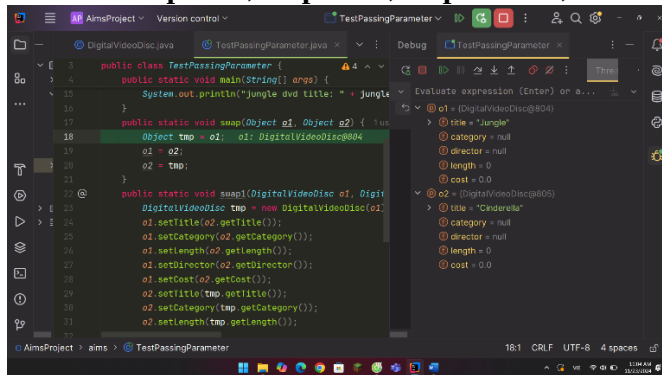
```
public static void swap1(DigitalVideoDisc o1, DigitalVideoDisc o2) { no usages
    DigitalVideoDisc tmp = new DigitalVideoDisc(o1);
    o1.setTitle(o2.getTitle());
    o1.setCategory(o2.getCategory());
    o1.setLength(o2.getLength());
    o1.setDirector(o2.getDirector());
    o1.setCost(o2.getCost());
    o2.setTitle(tmp.getTitle());
    o2.setCategory(tmp.getCategory());
    o2.setLength(tmp.getLength());
    o2.setDirector(tmp.getDirector());
    o2.setCost(tmp.getCost());
}
```

3. Use debug run:

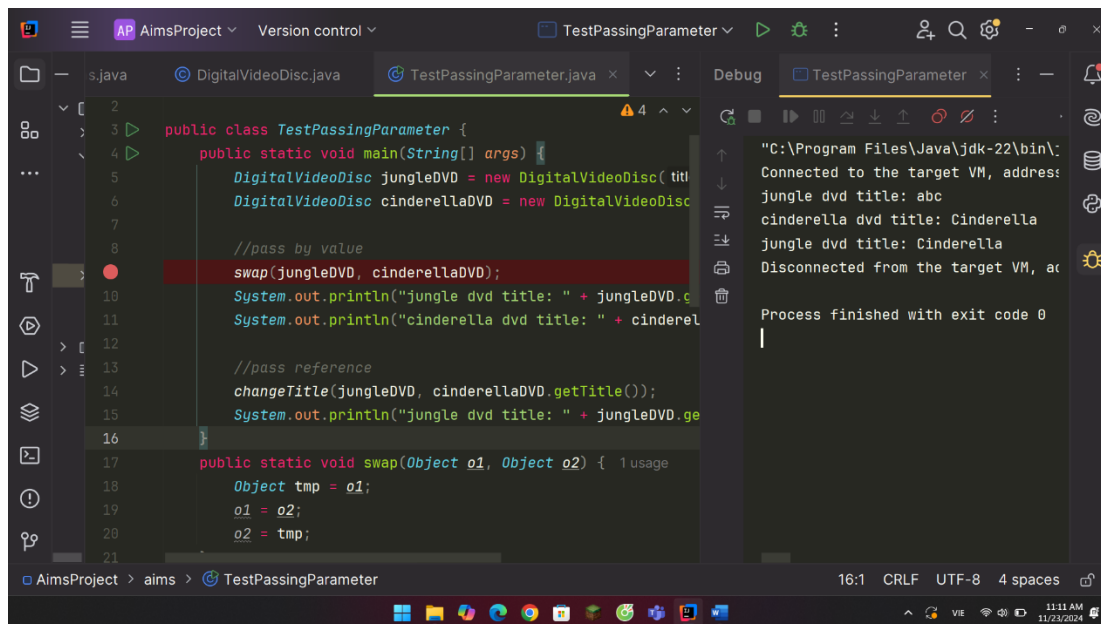
3.1 Setting, deleting & deactivate breakpoints:



3.2 Step Into, Step Over, Step Return, Resume:



3.3 Change value of variables:



4. Classifier Member and Instance Member

```
private static int nbDigitalVideoDiscs=0; 12 usages
private int id; 6 usages
```

```
nbDigitalVideoDiscs++;  
this.id = nbDigitalVideoDiscs;
```

5. Open the **Cart** class

```
public String toString() {  
    return "DVD-"+  
        this.title+  
        "-"+this.category+"-"+  
        this.director+"-"+  
        this.length+": "+  
        this.cost+"$";  
}  
  
public void print() { no usages  
    System.out.println("*****CART*****");  
    System.out.println("Ordered Items:");  
    for(DigitalVideoDisc item : itemsOrdered){  
        System.out.println(item.toString());  
    }  
    System.out.println("Total Cost: "+totalCost());  
    System.out.println("*****");  
}
```

```
public boolean isMatch(String title){ no usages  
    if(this.title.compareTo(title)==0) return true;  
    return false;  
}
```

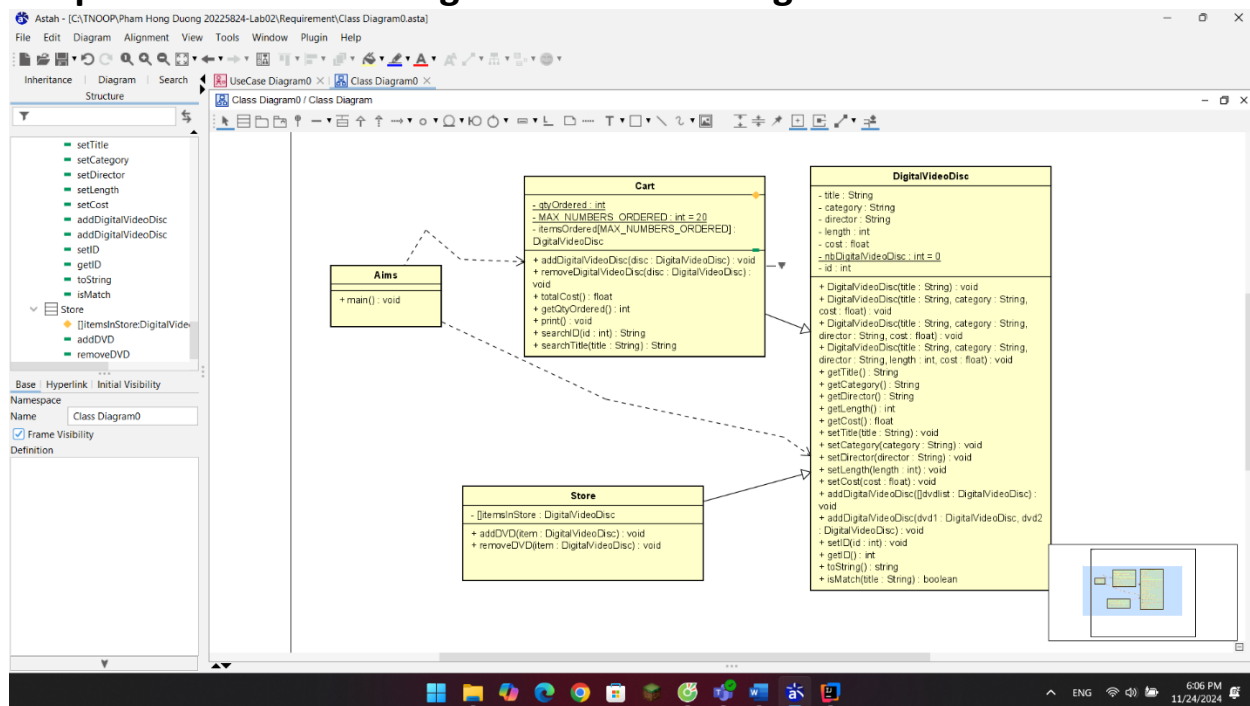
```
public String searchID(int ID){ no usages  
    for(DigitalVideoDisc item : itemsOrdered){  
        if(item.getId()==ID) return item.toString();  
    }  
    return "not found";  
}  
  
public String searchTitle(String title){ no usages  
    for(DigitalVideoDisc item : itemsOrdered){  
        if(item.isMatch(title)) return item.toString();  
    }  
    return "not found";  
}
```

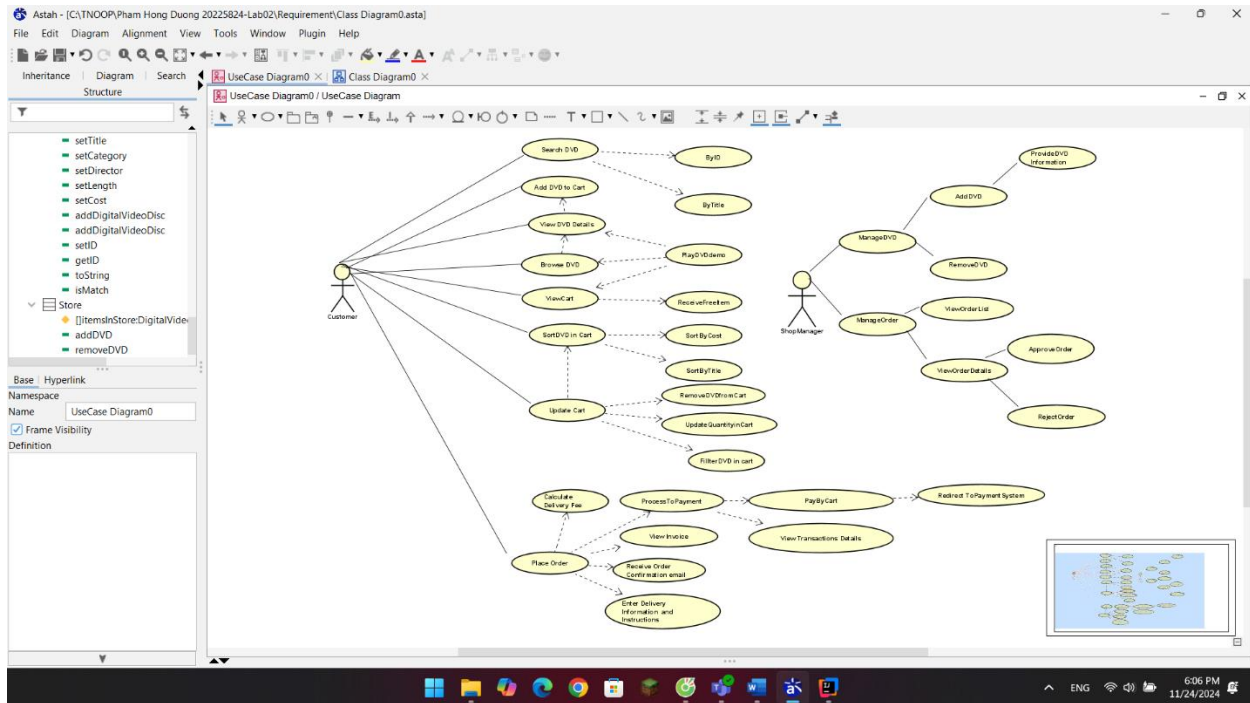
6. Implement the **Store** class

```
public void removeDVD(DigitalVideoDisc item){ no usages
    itemsInStore.remove(item);
    System.out.println("Removed DVD");
}
```

```
public void addDVD(DigitalVideoDisc item){ no usages
    itemsInStore.add(item);
    System.out.println("Added DVD");
}
```

7. Updated Use-Case Diagram and Class Diagram





8. Answering:

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

Java là **pass-by-value**.

- Với **kiểu nguyên thủy** (int, double, v.v.): Giá trị được sao chép. Thay đổi trong hàm **không ảnh hưởng** đến biến gốc.
- Với **đối tượng**: Tham chiếu (reference) tới đối tượng được sao chép.
 - Thay đổi nội dung đối tượng **ảnh hưởng** đến biến gốc.
 - Nhưng gán lại tham chiếu trong hàm **không ảnh hưởng** đến tham chiếu gốc.

Java **luôn là pass-by-value**, dù đôi khi làm việc với đối tượng có vẻ giống pass-by-reference.

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

Tiêu đề của hai đối tượng vẫn giữ nguyên vì trong Java, tham số được truyền vào phương thức là giá trị của đối tượng, không phải là tham chiếu đến đối tượng. Khi ta thay đổi giá trị của tham số bên trong phương thức (như việc đổi chỗ giữa o1 và o2), sự thay đổi này không ảnh hưởng đến giá trị của các đối tượng gốc.

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

Tiêu đề của jungleDVD bị thay đổi vì trong phương thức changeTitle, ta thực hiện thay đổi trực tiếp trên đối tượng dvd (được truyền vào phương thức) bằng cách gọi dvd.setTitle(title). Điều này ảnh hưởng trực tiếp đến đối tượng gốc được truyền vào phương thức.