Lab03: Basic Object-Oriented Techniques

Section 1: Branch your repository

```
caoducanh@Caos-MacBook-Pro Lab03Workspace % git checkout -b refactor
Switched to a new branch 'refactor'
caoducanh@Caos-MacBook-Pro Lab03Workspace % git checkout -b feature
Switched to a new branch 'feature'
caoducanh@Caos-MacBook-Pro Lab03Workspace % git checkout -b bugfix
Switched to a new branch 'bugfix'
caoducanh@Caos-MacBook-Pro Lab03Workspace % git checkout -b hotfix
Switched to a new branch 'hotfix'
caoducanh@Caos-MacBook-Pro Lab03Workspace % git checkout -b release
Switched to a new branch 'release'
caoducanh@Caos-MacBook-Pro Lab03Workspace %
```

Figure 1. Creating common branches to local repo

```
Lab03Workspace — -zsh — 80×24
 create mode 100644 AimsProject/src/hust/soict/ite6/test/disc/TestPassingParamet
er.java
 create mode 100644 AimsProject/src/hust/soict/ite6/test/store/StoreTest.java
 create mode 100644 OtherProjects/.DS_Store
 create mode 100644 OtherProjects/.classpath
 create mode 100644 OtherProjects/.project
 create mode 100644 OtherProjects/.settings/org.eclipse.core.resources.prefs
 create mode 100644 OtherProjects/.settings/org.eclipse.jdt.core.prefs
 create mode 100644 OtherProjects/bin/hust/soict/ite6/garbage/ConcatenationInLoo
ps.class
 create mode 100644 OtherProjects/bin/hust/soict/ite6/garbage/GarbageCreator.cla
ss
 create mode 100644 OtherProjects/bin/hust/soict/ite6/garbage/NoGarbage.class
 create mode 100644 OtherProjects/src/.DS_Store
 create mode 100644 OtherProjects/src/garbagegen.txt
 create mode 100644 OtherProjects/src/hust/.DS_Store
 create mode 100644 OtherProjects/src/hust/soict/.DS_Store
 create mode 100644 OtherProjects/src/hust/soict/ite6/.DS_Store
 create mode 100644 OtherProjects/src/hust/soict/ite6/garbage/ConcatenationInLoo
ps.java
 create mode 100644 OtherProjects/src/hust/soict/ite6/garbage/GarbageCreator.jav
a
 create mode 100644 OtherProjects/src/hust/soict/ite6/garbage/NoGarbage.java
caoducanh@Caos-MacBook-Pro Lab03Workspace %
```

Figure 2. Initial commit to the 'master' branch on local repo

Section 2: Working with method overloading

2.1 Overloading by differing types of parameter

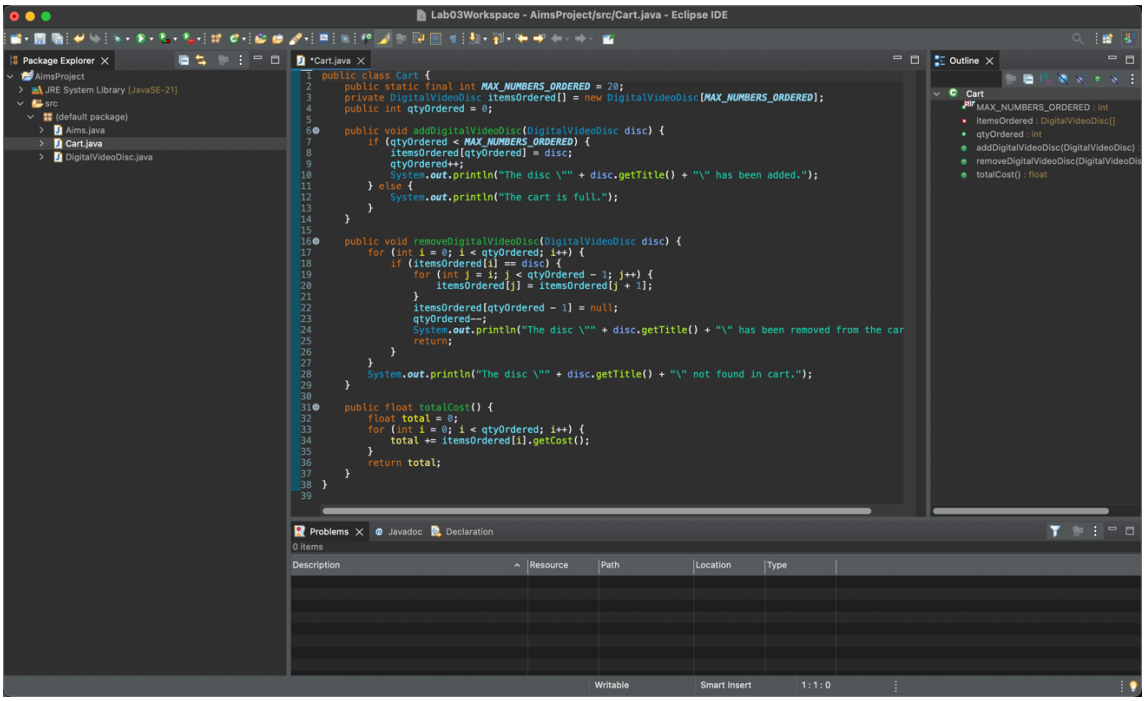

Figure 3. 'Cart' class from the previous lab

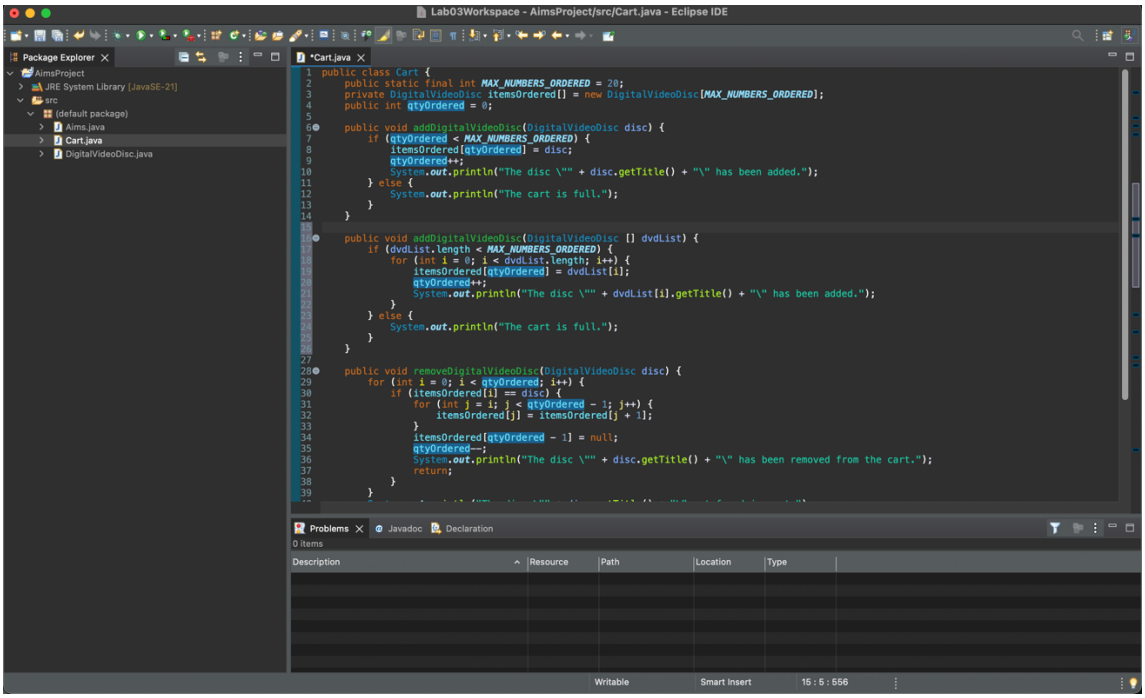

Figure 4. Overload method 'addDigitalVideoDisc' to add a list of DVDs to cart

2.2. Overloading by differing the number of parameters

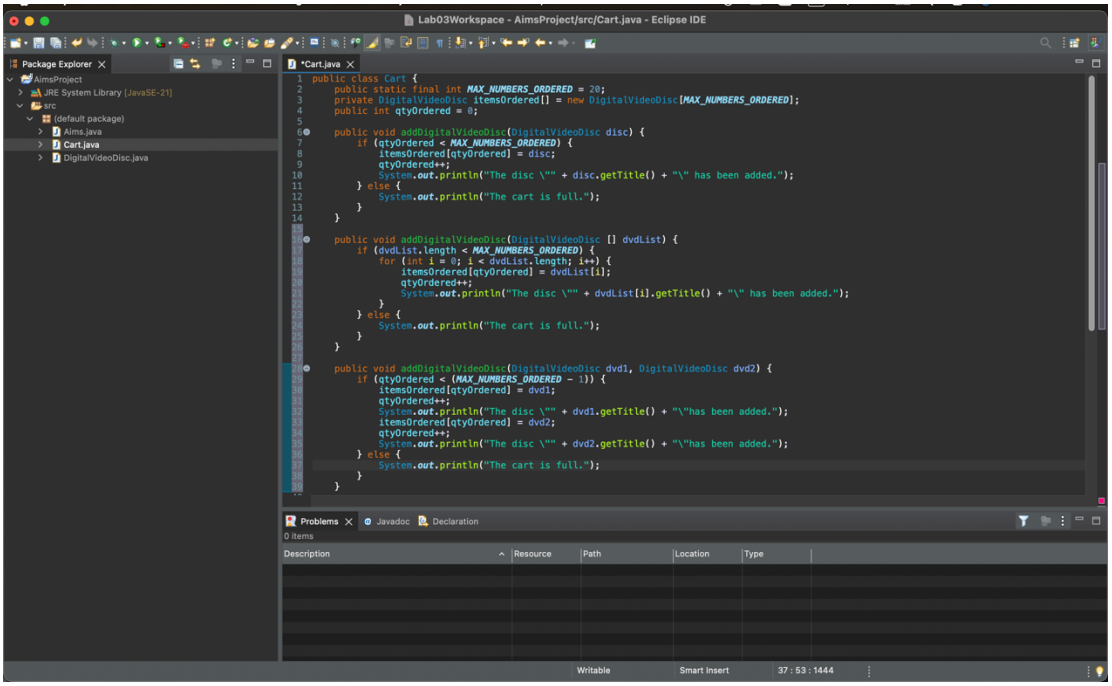

Figure 5. Another 'addDigitalVideoDisc' method to add 2 discs simultaniously to cart

Section 3: Passing parameter

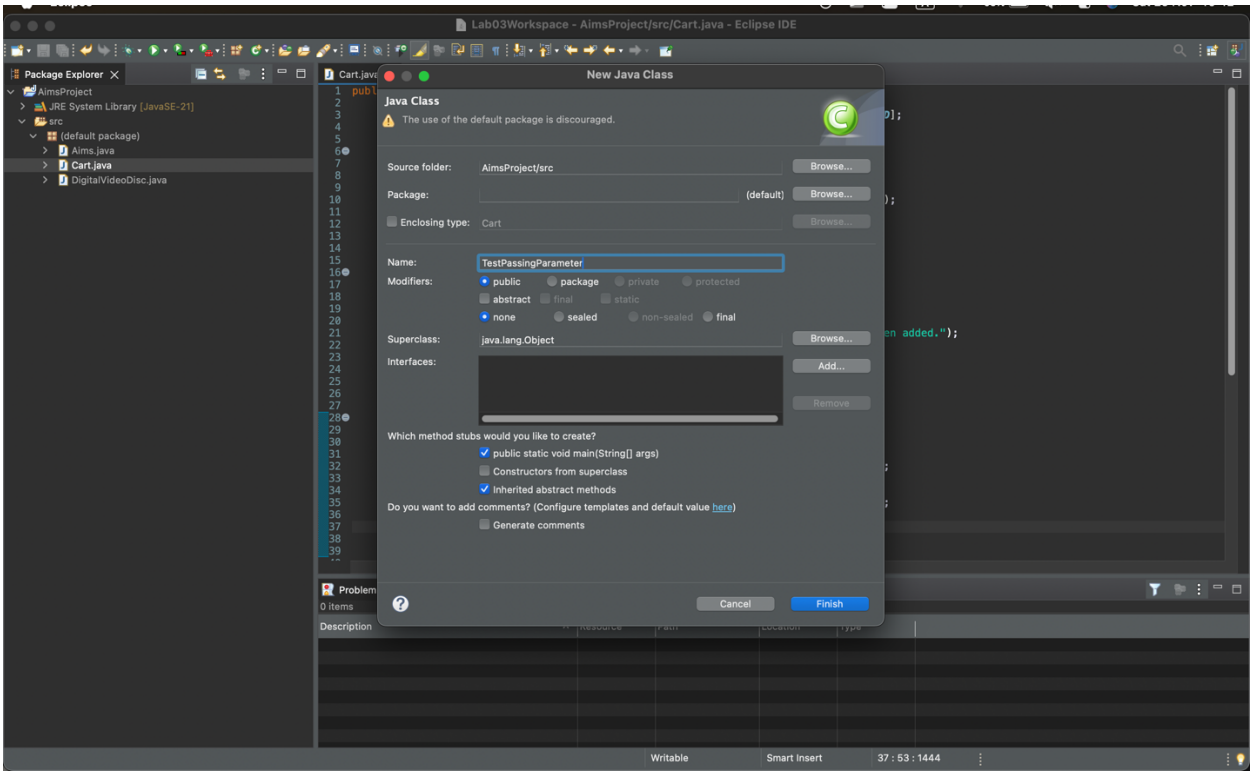

Figure 6. Create new class named 'TestPassingParameter' to test how Java passes parameters

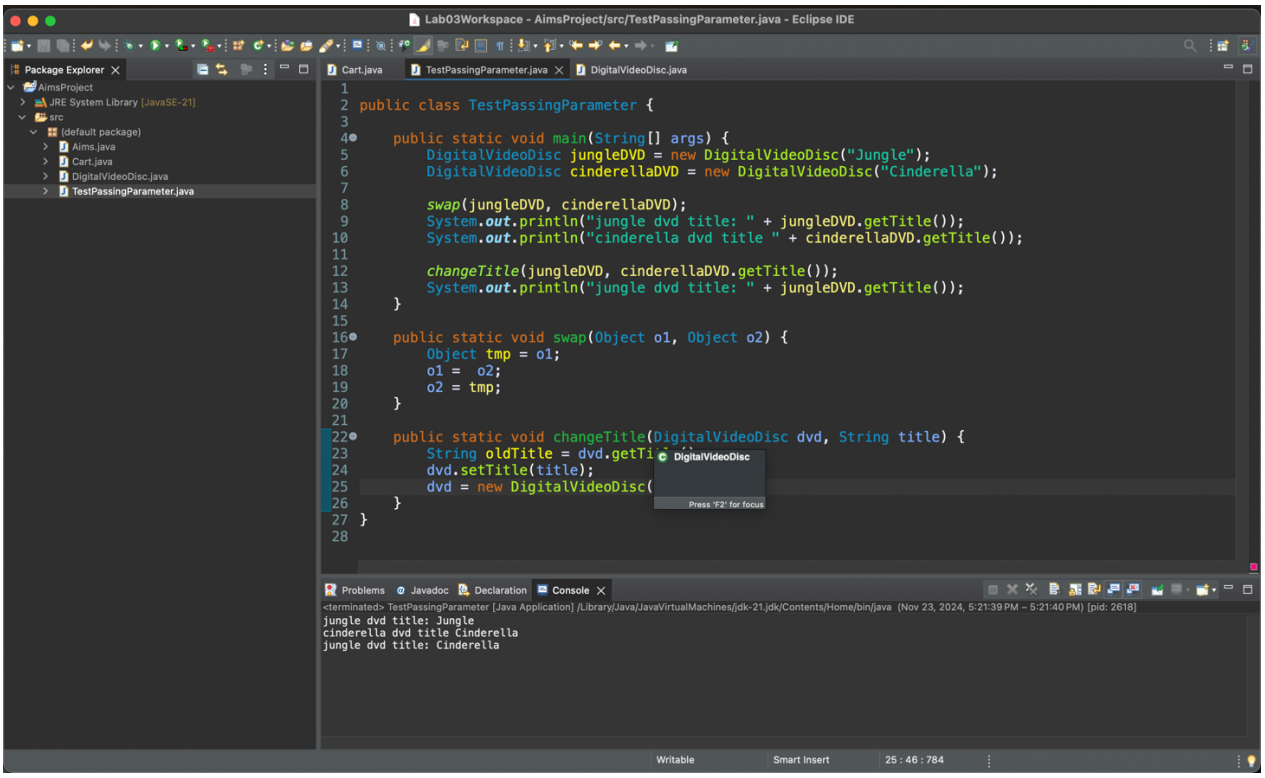

Figure 7. 'TestPassingParameter' class contents and run results

- The title of the two Objects remain unchanged after calling ‘swap(jungleDVD, cinderellaDVD)’ because Java is a pass by value language, including Object references. When passing the 2 DVDs Objects to the ‘swap’ method, the method receives copies of their references, thus only modifying these references, not the originals.
- When the ‘changeTitle’ method is called, a reference to the ‘jungleDVD’ object is passed to the method by value. This reference copy still points to the same object in memory. Thus, when ‘dvd.setTitle(title)’ is called, it directly modifies the title field of the ‘jungleDVD’ object.
- To rewrite ‘swap’ to make it works, we can use an array to hold the two objects then swap them:

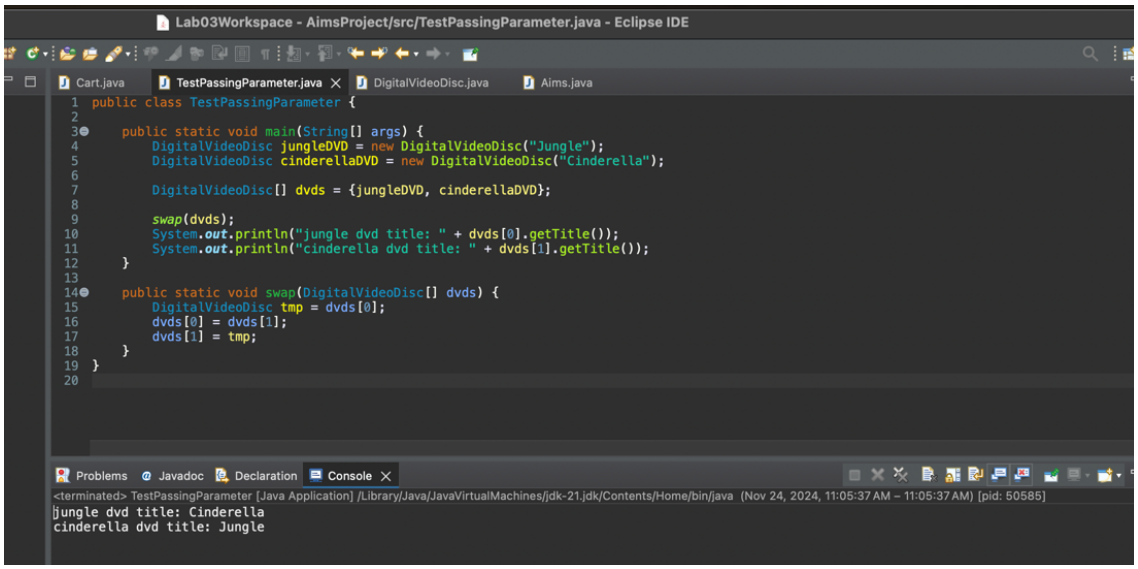

Figure 8. Rewritten 'swap' method to swap titles of 2 DVDs

Section 4: Use debug run

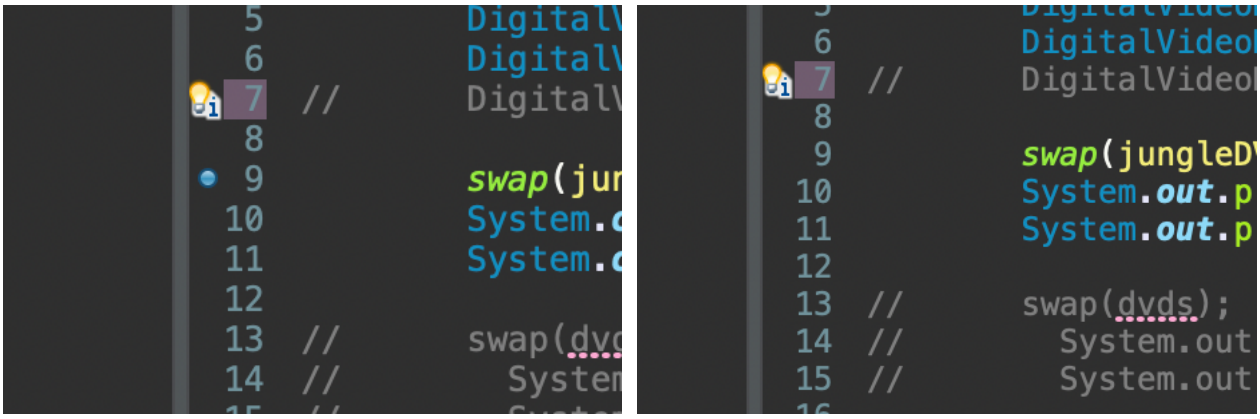

Figure 9. Toggling a breakpoint

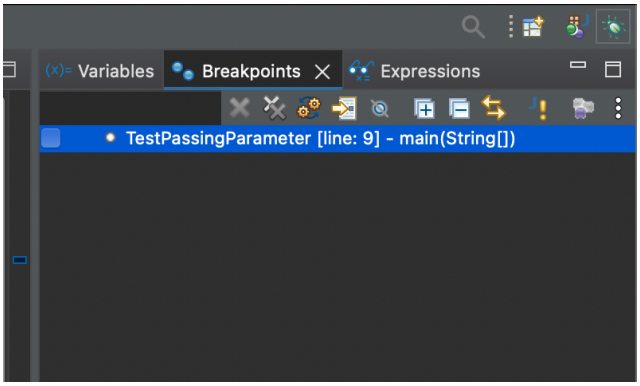

Figure 10. Enabling/Disabling breakpoints in the 'Breakpoints' window

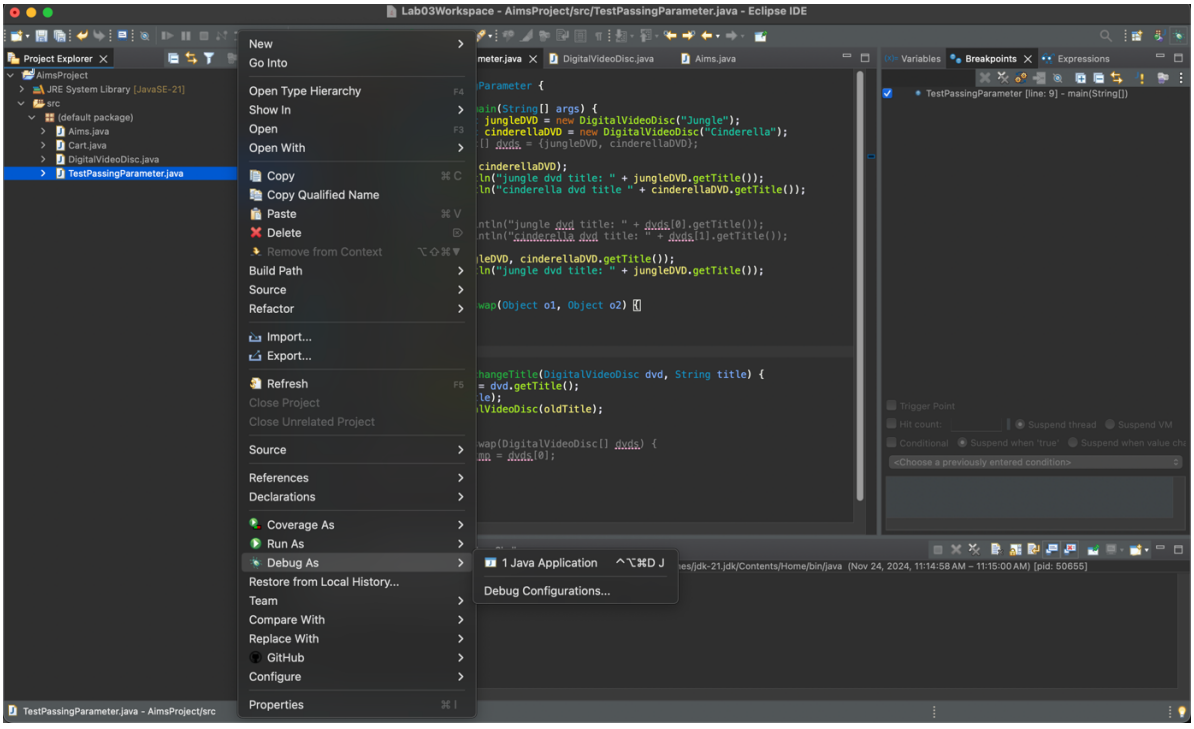

Figure 11. Steps to debug the class 'TestPassingParameter'

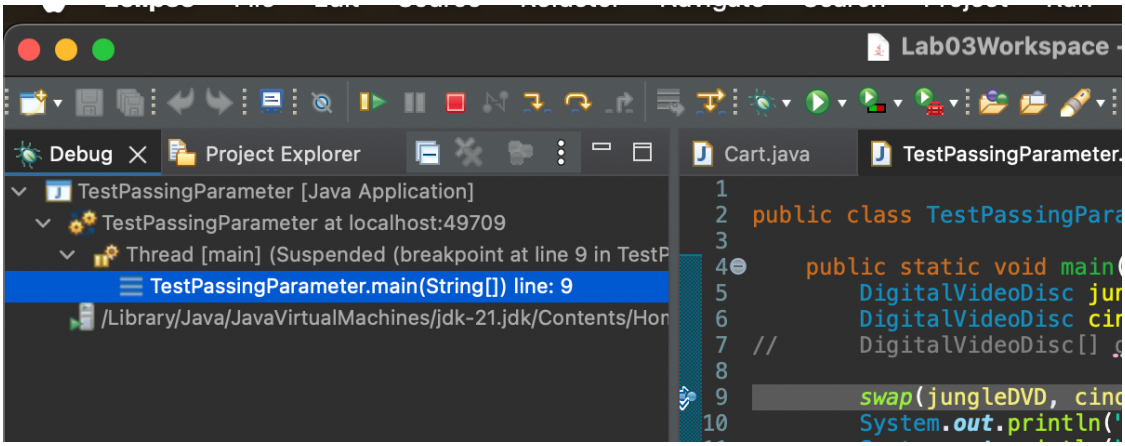

Figure 12. The toolbar in Debug perspective

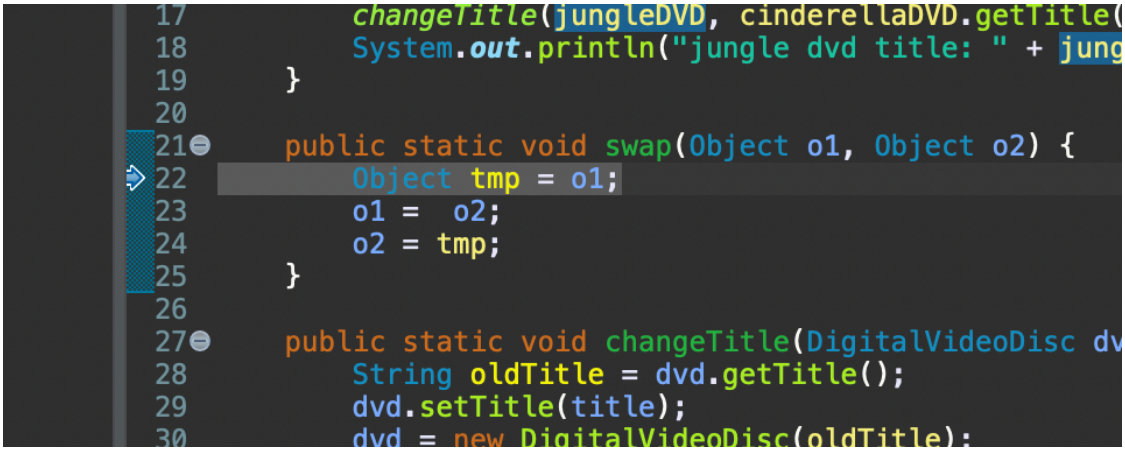

Figure 13. Stepping into the 'swap' method

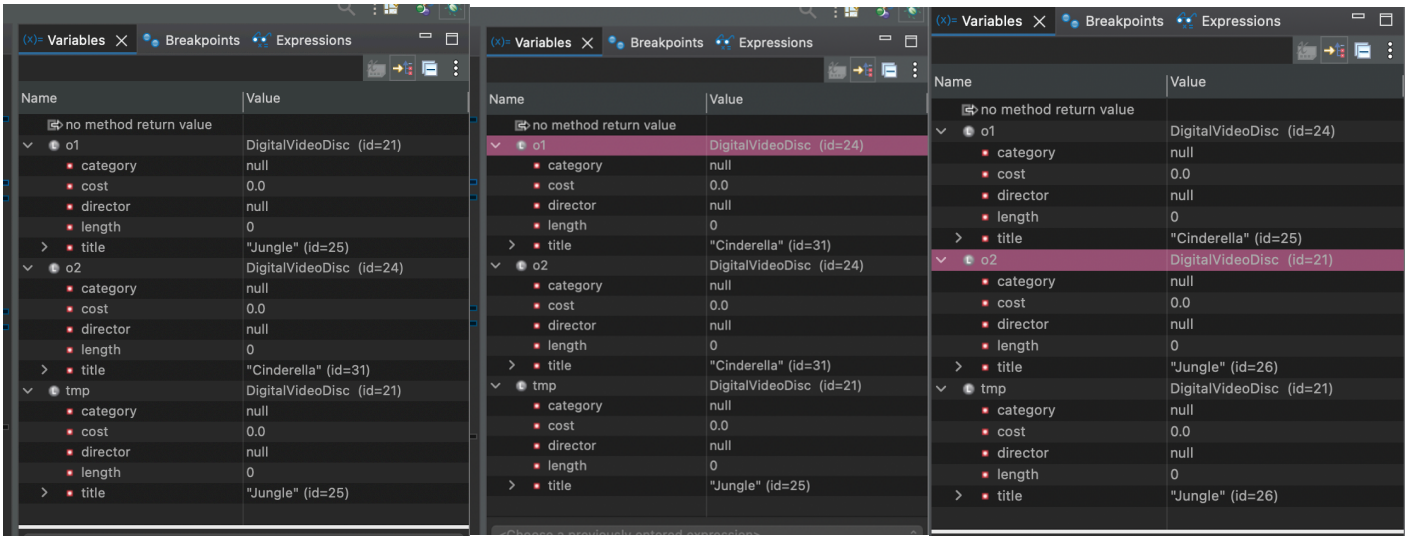

Figure 14. Observing variable 'o1', 'o2', 'tmp' change in 'swap' method

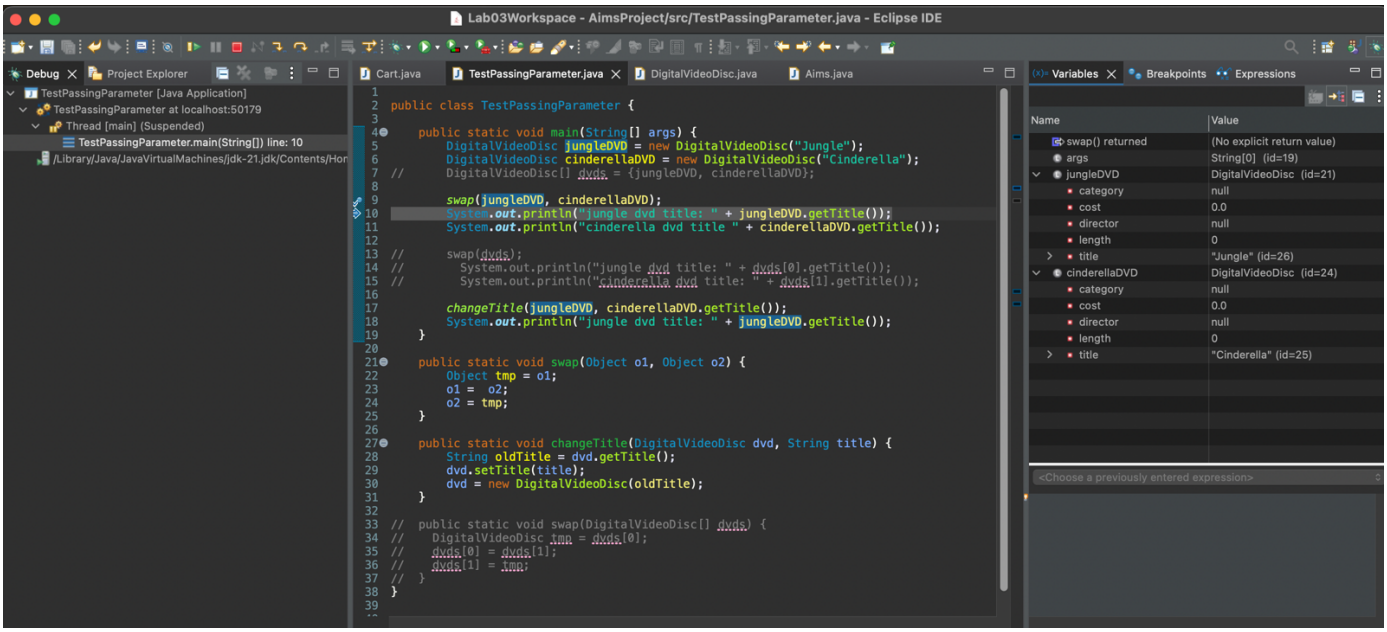

Figure 15. Using Step Return to return to the line after calling 'swap'

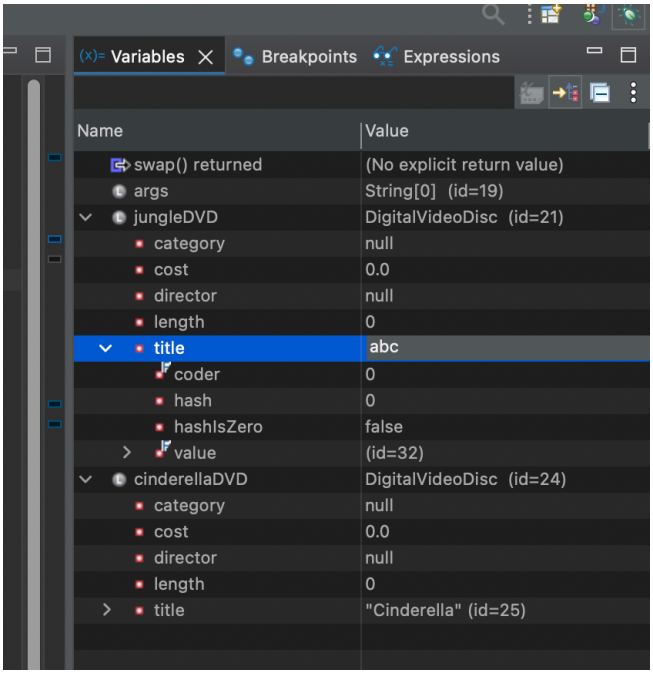

Figure 16. Changing the title attriute of 'jungleDVD'

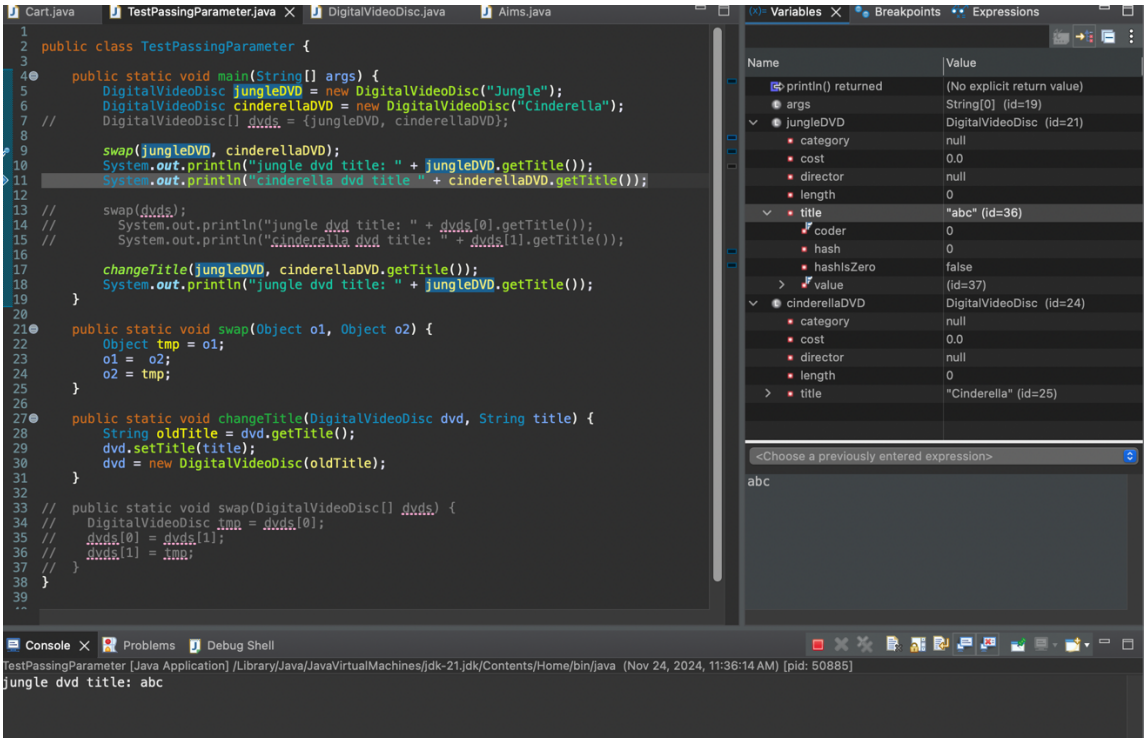

Figure 17. Run result after modifying the variable

Section 5: Classifier Member and Instance Member

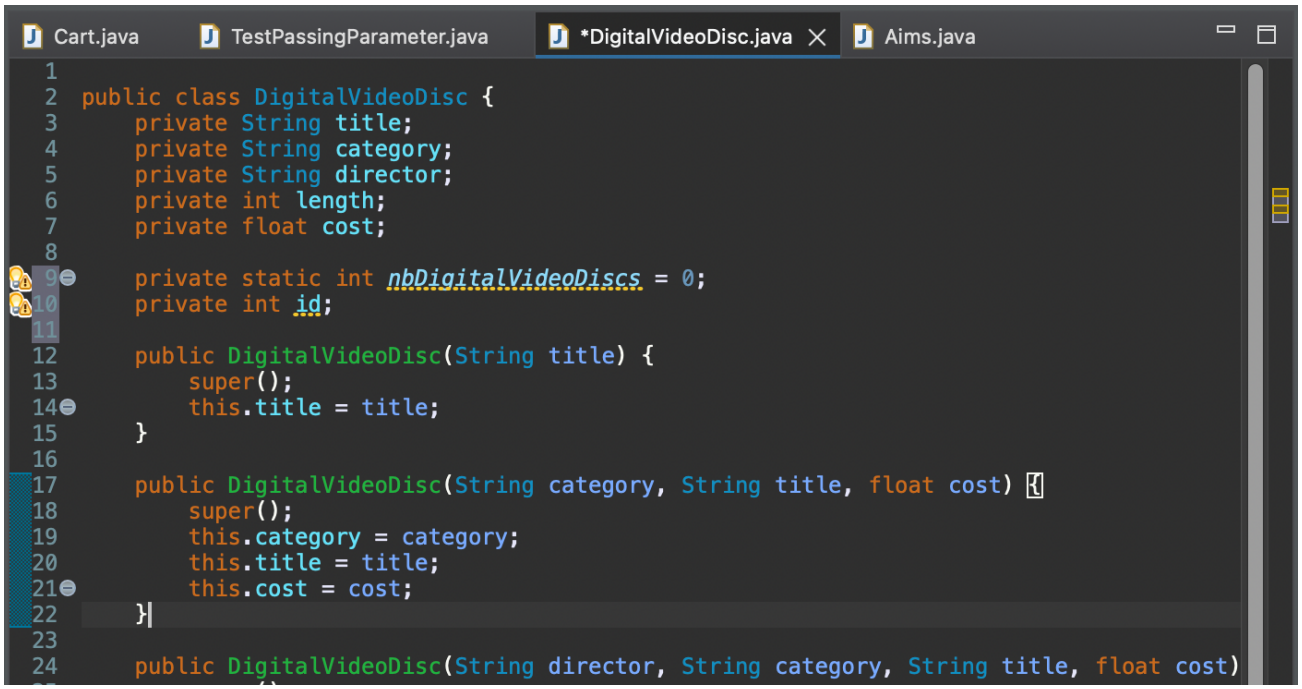

Figure 18. Create a class attribute named 'nbDigitalVideoDiscs' and an instance attributre 'id'

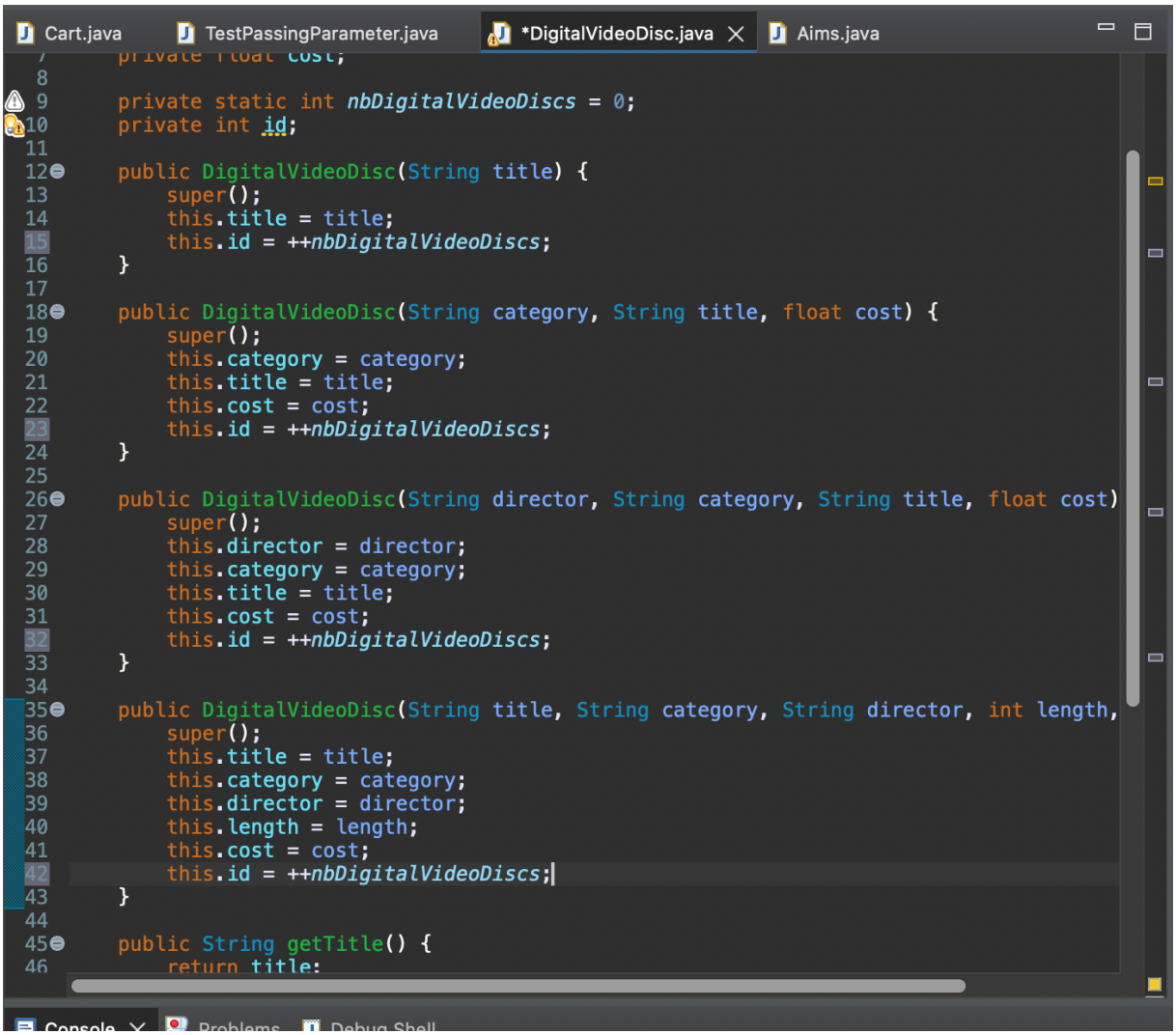

Figure 19. Update 'nbDigitalVideoDiscs' inside every constructor methods and assign the value to 'id'

Section 6: Update the 'Cart' class

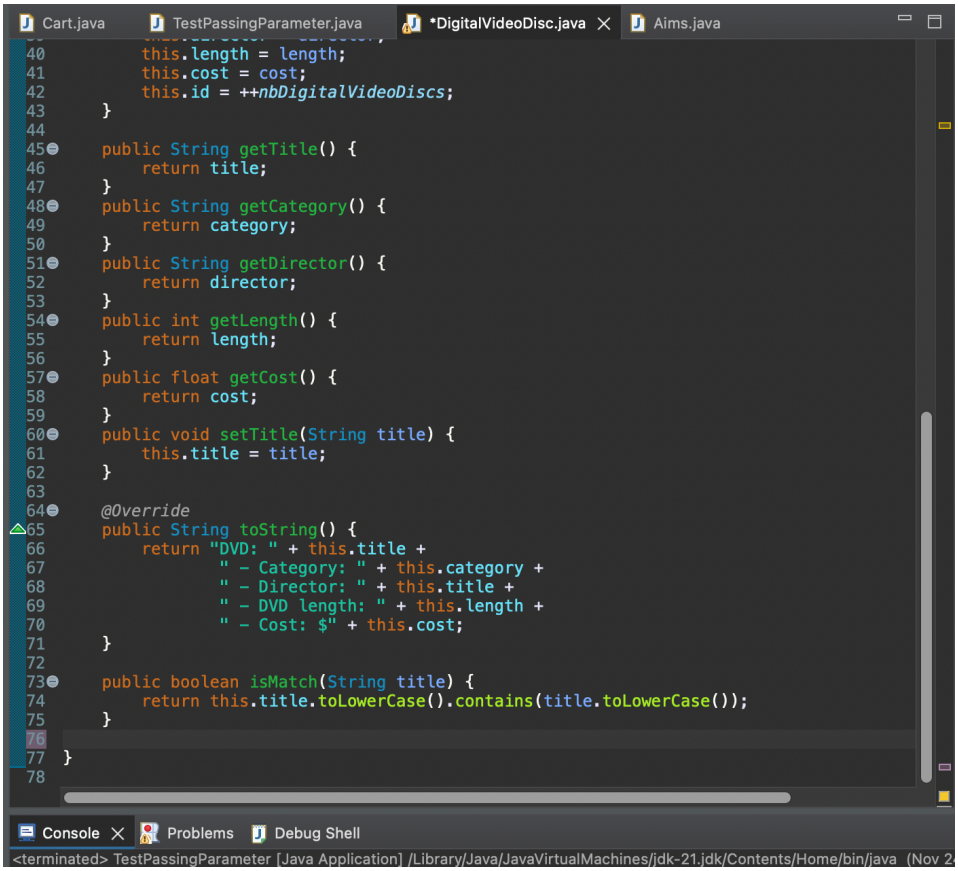

Figure 20. Write a new 'toString' method to return a DVD's info by a format and a 'isMatch' method to check for DVD's title match

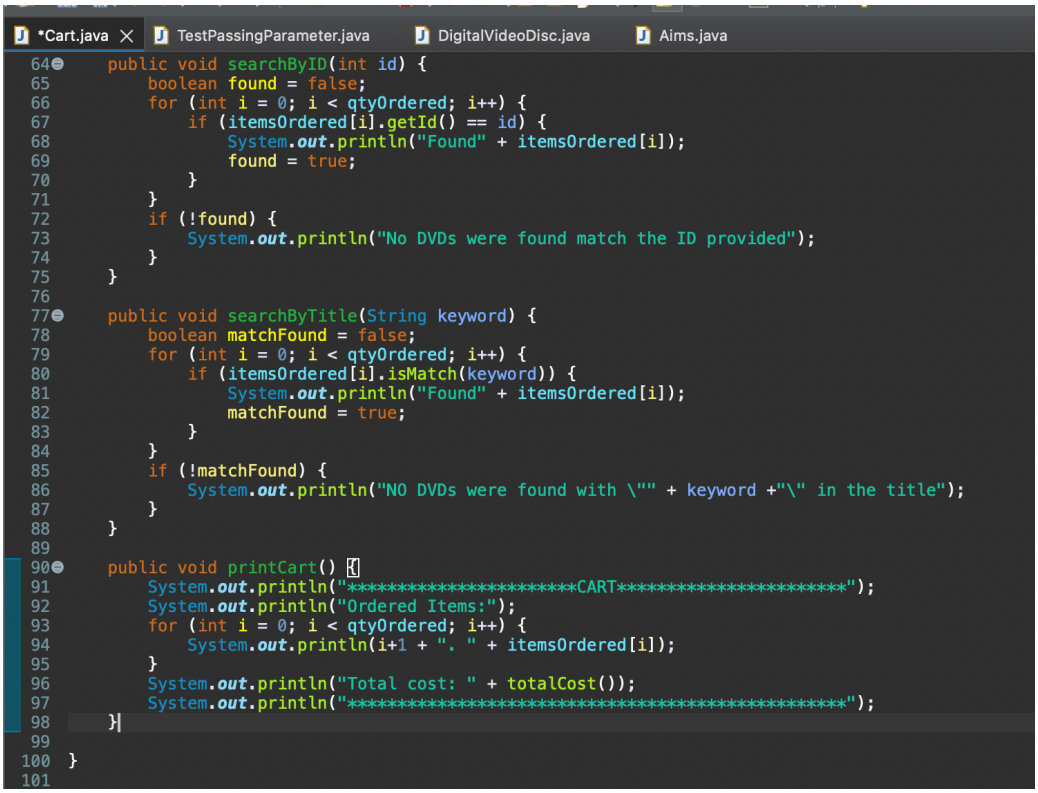

Figure 21. New methods in 'Cart' to print out cart contents, search DVDs by titles and IDs

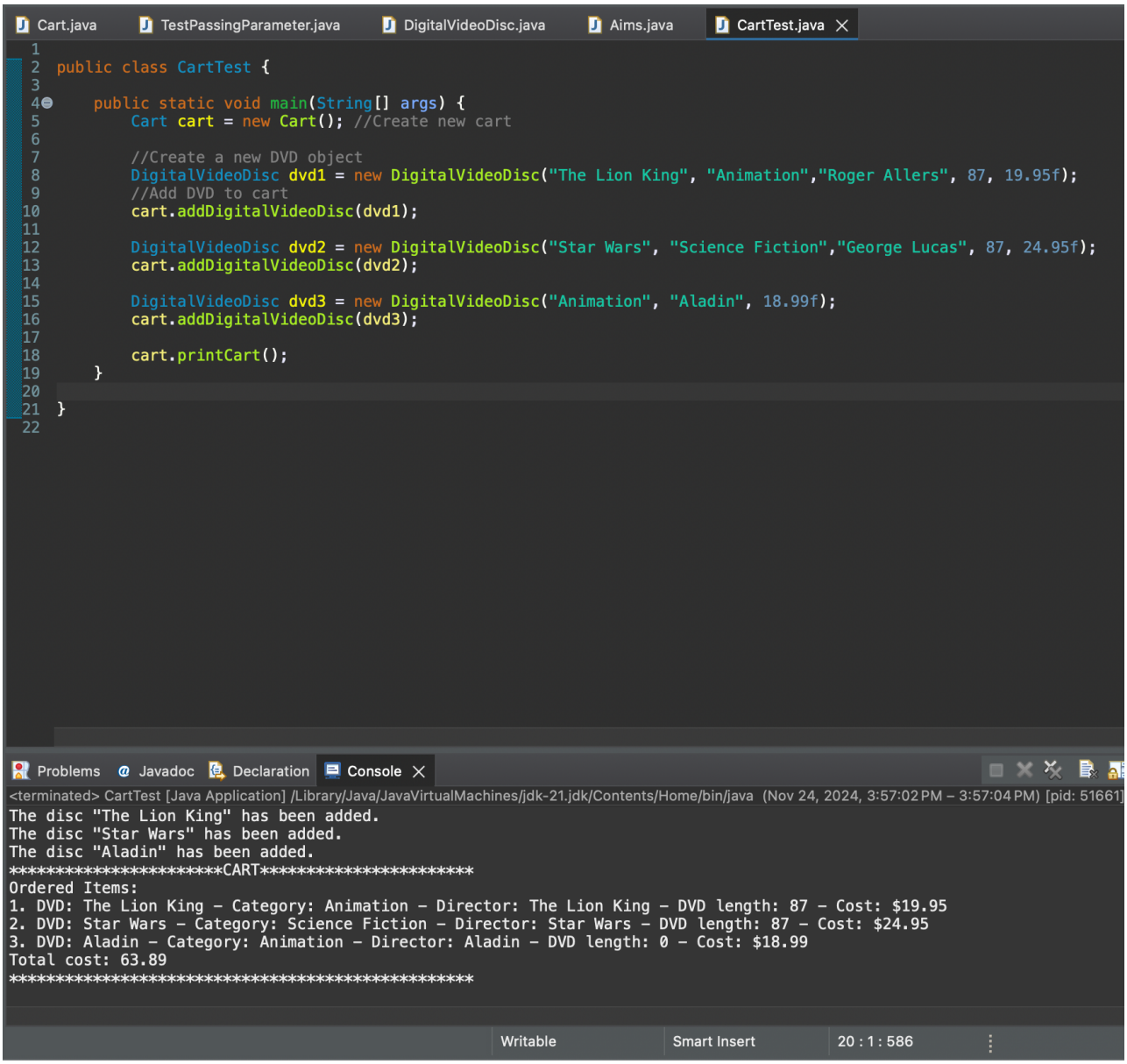

Figure 22. Testing the new methods in a new class 'CartTest'

Section 7: Implement the 'Store' class

Cart.java | TestPassingParameter.java | DigitalVideoDisc.java | Aims.java | CartTest.java | *Store.java

```java
import java.util.ArrayList;
import java.util.List;


public class Store {
    private List<DigitalVideoDisc> itemsInStore = new ArrayList<DigitalVideoDisc>();

    public void addDVD(DigitalVideoDisc dvd) {
        itemsInStore.add(dvd);
        System.out.println("DVD: " + dvd.getTitle() + " has been added to the store.");
    }
    public void removeDVD(DigitalVideoDisc dvd) {
        boolean removed = itemsInStore.remove(dvd);
        if (removed) {
            System.out.println("DVD: " + dvd.getTitle() + " has been removed from the store.");
        } else {
            System.out.println("DVD: " + dvd.getTitle() + " is unavailable in the store.");
        }
    }

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

```

Figure 23. New 'Store' class with single attribute 'itemsInStore[]' and 3 methods 'addDVD', 'removeDVD', and 'print'

Cart.java | TestPassingParameter.java | DigitalVideoDisc.java | Aims.java | CartTest.java | Store.java | StoreTest.java

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

        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        store.addDVD(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star War", "Science Fiction", "George Lucas", 87, 24.95f);
        store.addDVD(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
        store.addDVD(dvd3);

        store.print();
        store.removeDVD(dvd2);
        store.print();
    }
}
```

Problems | Javadoc | Declaration | Console

```
<terminated> StoreTest [Java Application] /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java (Nov 24, 2024, 4:23:13 PM – 4:23:13 PM) [pid: 51799]
DVD: The Lion King has been added to the store.
DVD: Star War has been added to the store.
DVD: Animation has been added to the store.
1. DVD: The Lion King - Category: Animation - Director: The Lion King - DVD length: 87 - Cost: $19.95
2. DVD: Star War - Category: Science Fiction - Director: Star War - DVD length: 87 - Cost: $24.95
3. DVD: Animation - Category: Aladin - Director: Animation - DVD length: 0 - Cost: $18.99
DVD: Star War has been removed from the store.
1. DVD: The Lion King - Category: Animation - Director: The Lion King - DVD length: 87 - Cost: $19.95
2. DVD: Animation - Category: Aladin - Director: Animation - DVD length: 0 - Cost: $18.99
```

Figure 24. New 'StoreTest' to test methods in 'Store' clas

Section 8: Re-organize your projects

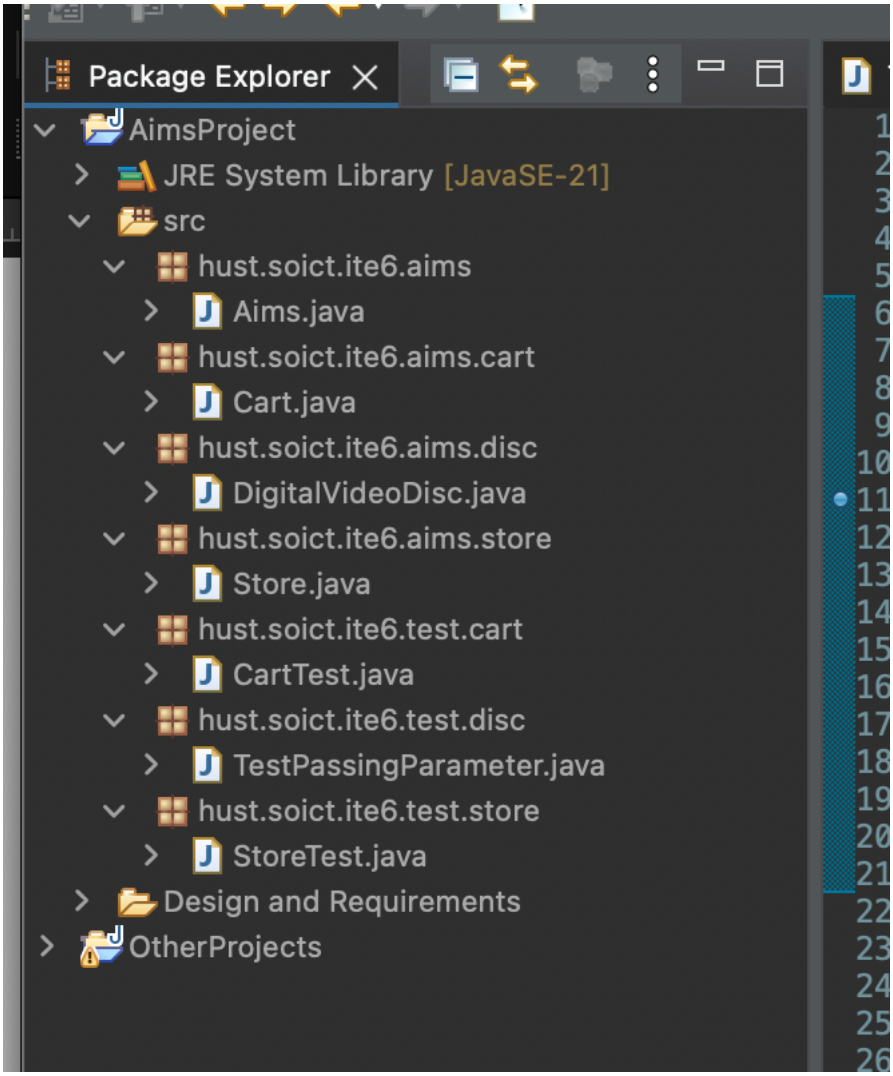

Figure 25. Reorganized classes inside packages

Section 9: String, StringBuilder and StringBuffer

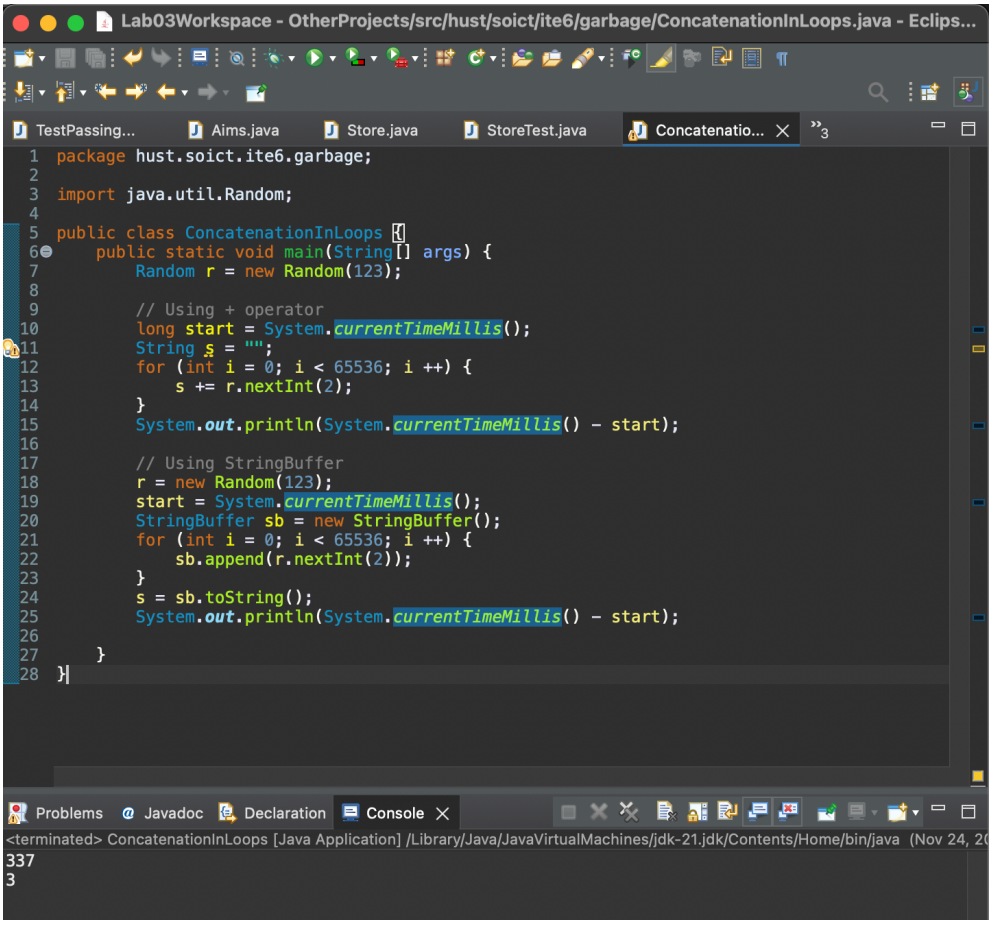

Figure 26. This shows string concatenation using StringBuffer yields better performance than using the + operator

Another demonstration:

Figure 27. A text file with extreme long string

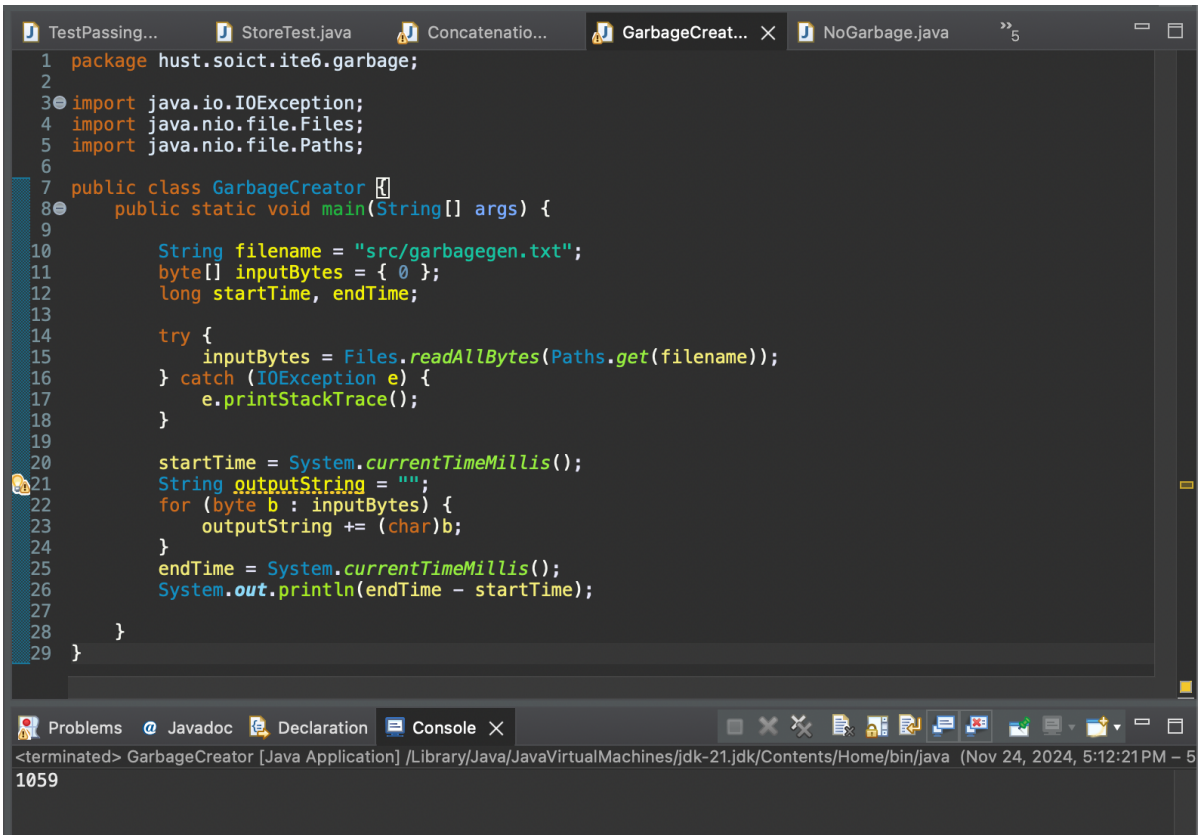

Figure 28. The 'GarbageCreator' reads the above file and process it using the + operator take about 1sec of runtime

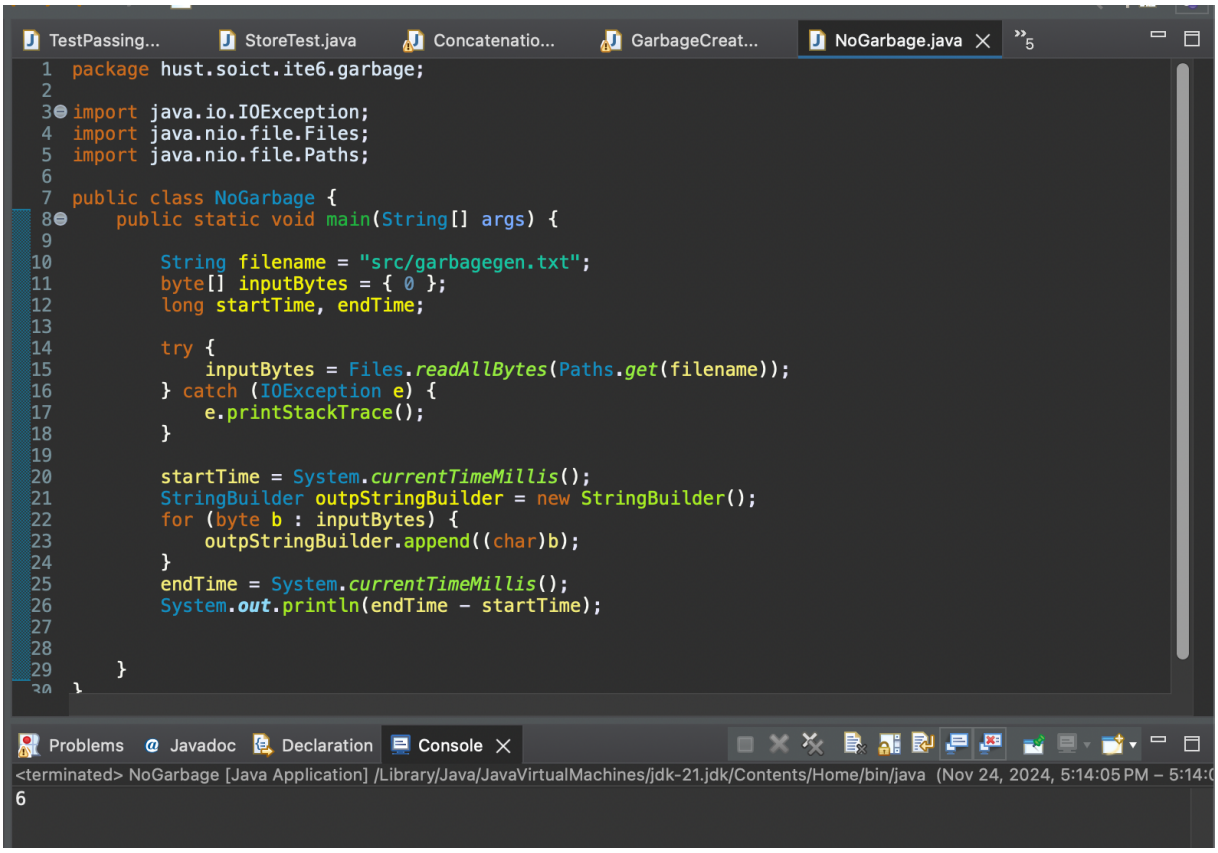

Figure 29. Another implementation using StringBuilder shortens the runtime to only 6ms

Section 10: Release flow demonstration

```
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
caoducanh@Caos-MacBook-Pro Lab03Workspace % git checkout -b feature/demonstrate-release-flow
Switched to a new branch 'feature/demonstrate-release-flow'
caoducanh@Caos-MacBook-Pro Lab03Workspace % git commit -m “Add a feature for demonstration”
```

Figure 30. Create a new branch in the local repo

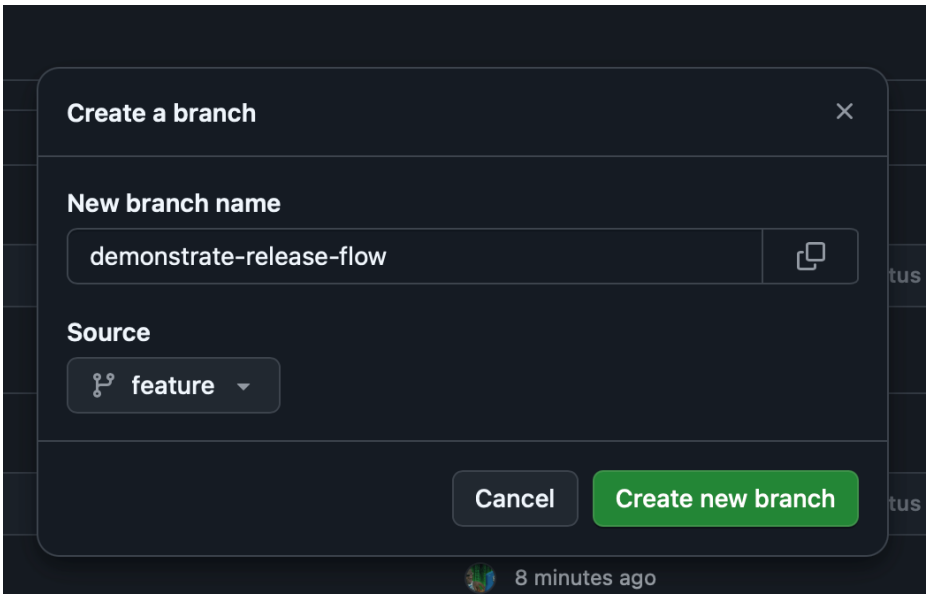

Figure 31. Creating remote branch

```
caoducanh@Caos-MacBook-Pro Lab03Workspace % git commit -m "Add feature for demo"
On branch feature/demonstrate-release-flow
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    AimsProject/Design and Requirements/AimsUML.asta.lock

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store
        .metadata/

no changes added to commit (use "git add" and/or "git commit -a")
caoducanh@Caos-MacBook-Pro Lab03Workspace %
```

Figure 32. Commit changes to the branch 'feature/demonstrate-release-flow' on local repo

```
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
caoducanh@Caos-MacBook-Pro Lab03Workspace % git push origin feature/demonstrate-release-flow
Enumerating objects: 138, done.
Counting objects: 100% (138/138), done.
Delta compression using up to 8 threads
Compressing objects: 100% (104/104), done.
Writing objects: 100% (138/138), 194.67 KiB | 7.49 MiB/s, done.
Total 138 (delta 14), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (14/14), done.
remote:
remote: Create a pull request for 'feature/demonstrate-release-flow' on GitHub by visiting:
remote:      https://github.com/KazooBoye/ObjectOrientedProgrammingLab744521/pull/new/feature/
demonstrate-release-flow
remote:
To https://github.com/KazooBoye/ObjectOrientedProgrammingLab744521.git
 * [new branch]      feature/demonstrate-release-flow -> feature/demonstrate-release-flow
caoducanh@Caos-MacBook-Pro Lab03Workspace %
```

Figure 33. Adding remote repo as reference the local repo then push changes to the remote repo

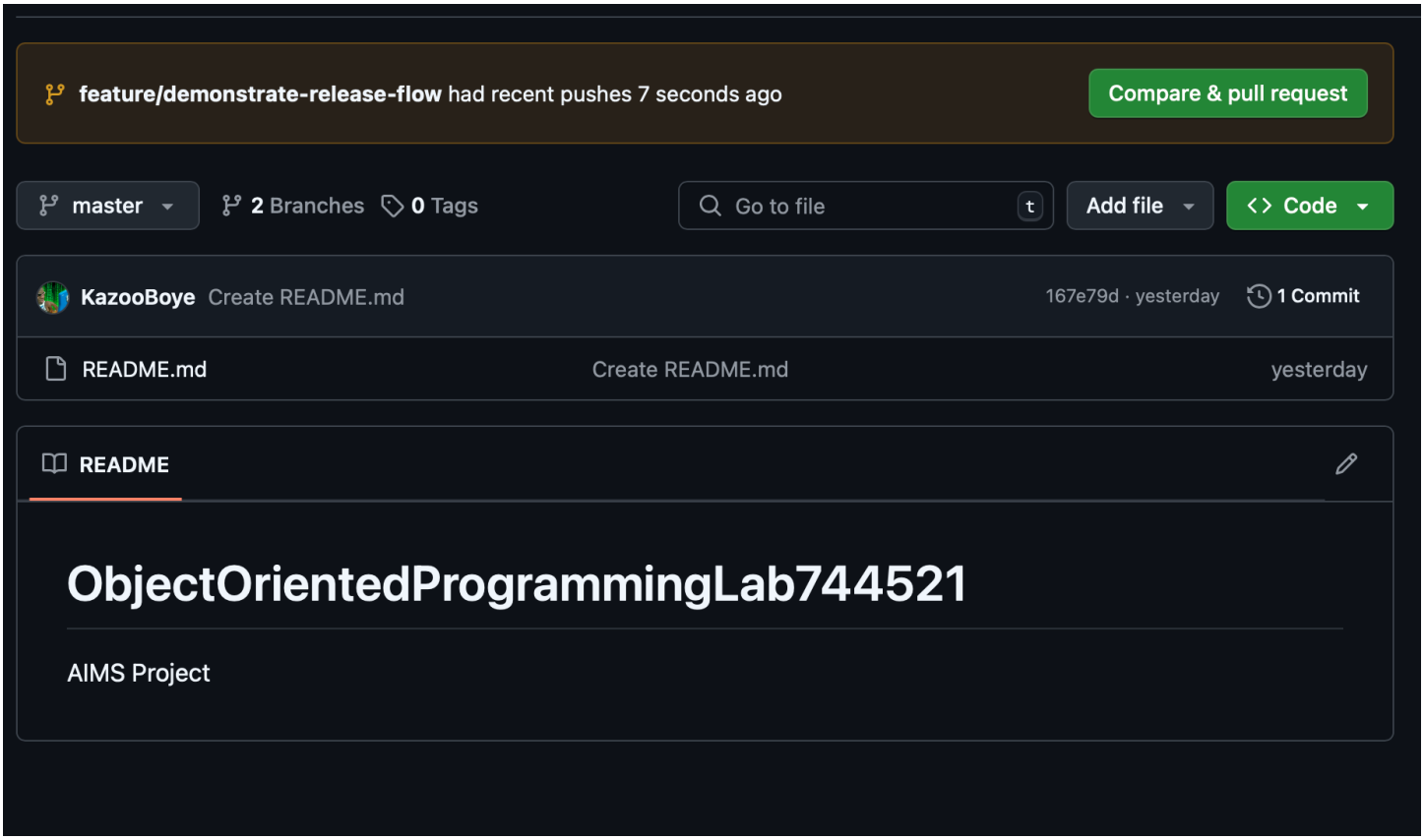

Figure 34. New pull request has been created on the remote repo

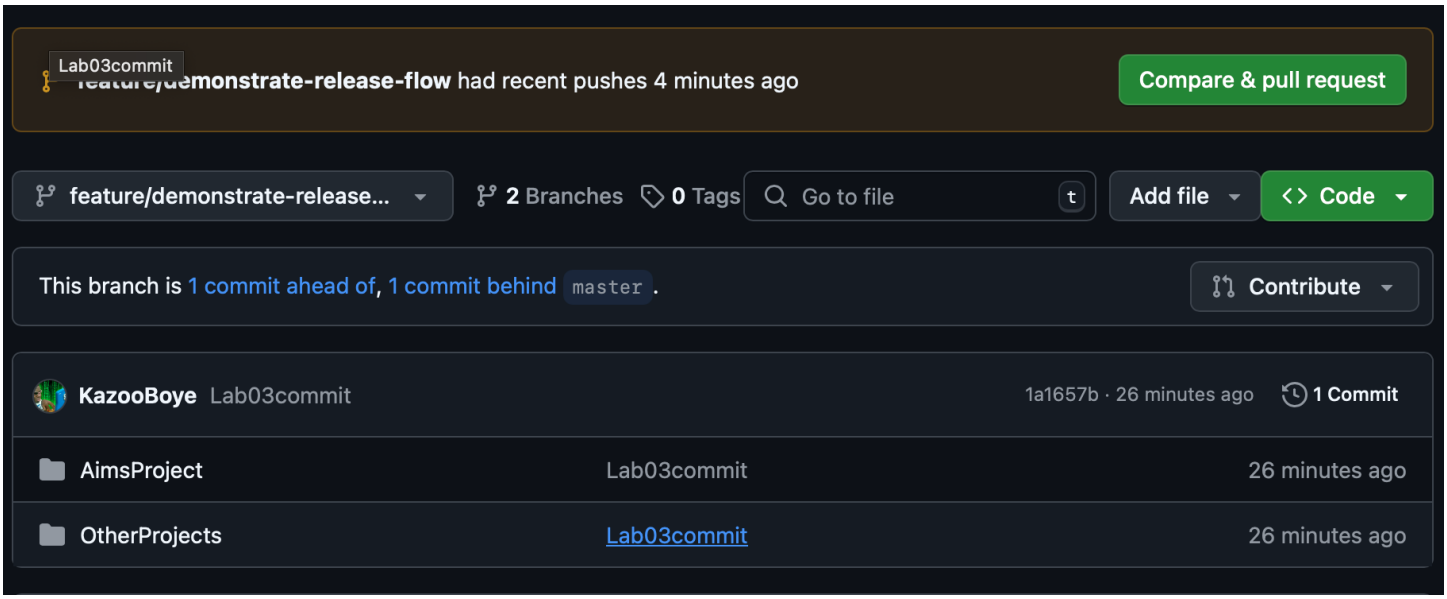

Figure 35. Switch to the new branch and compare changes

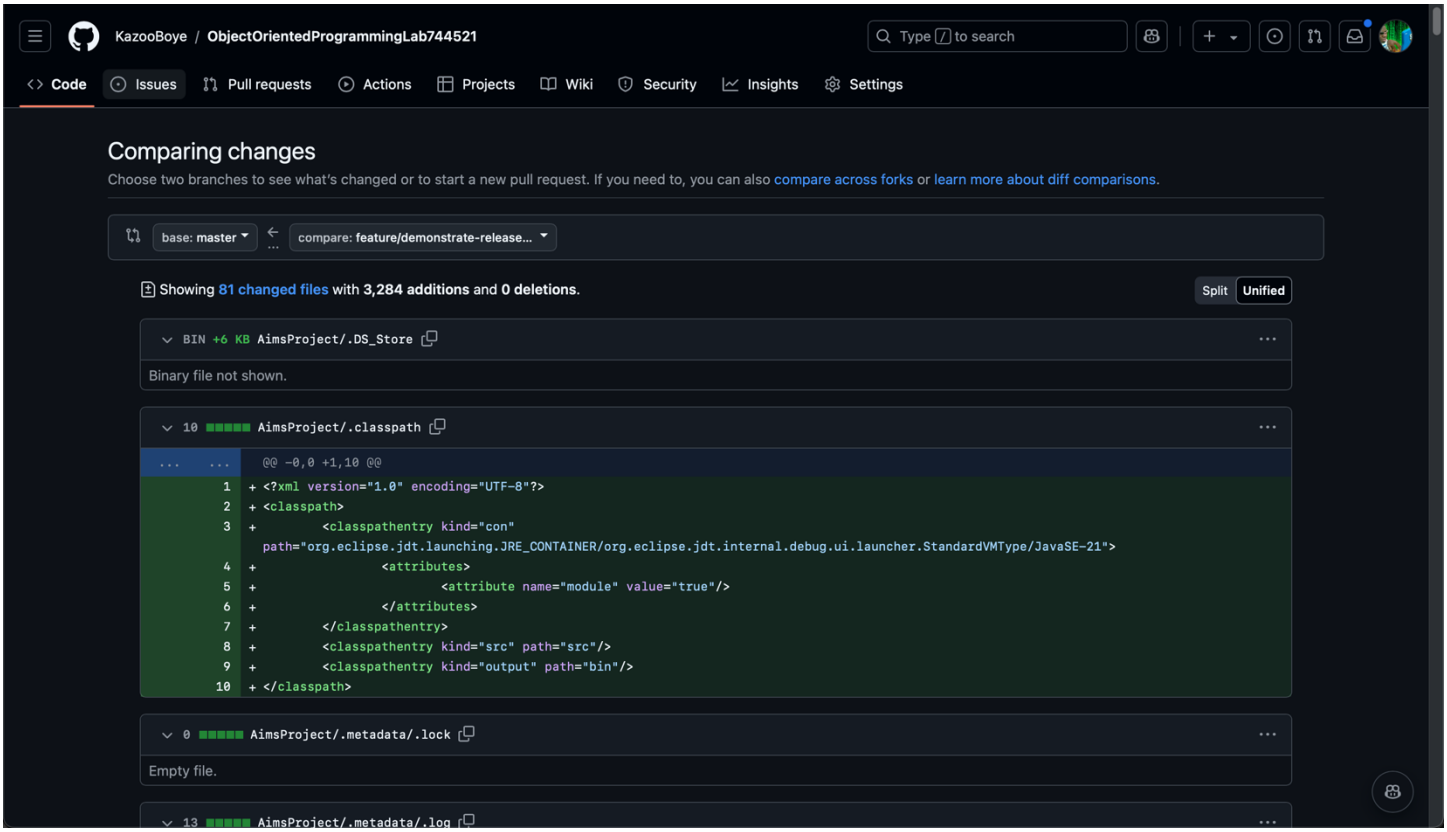

Figure 36. Creating a pull request to merge the current branch to 'master'

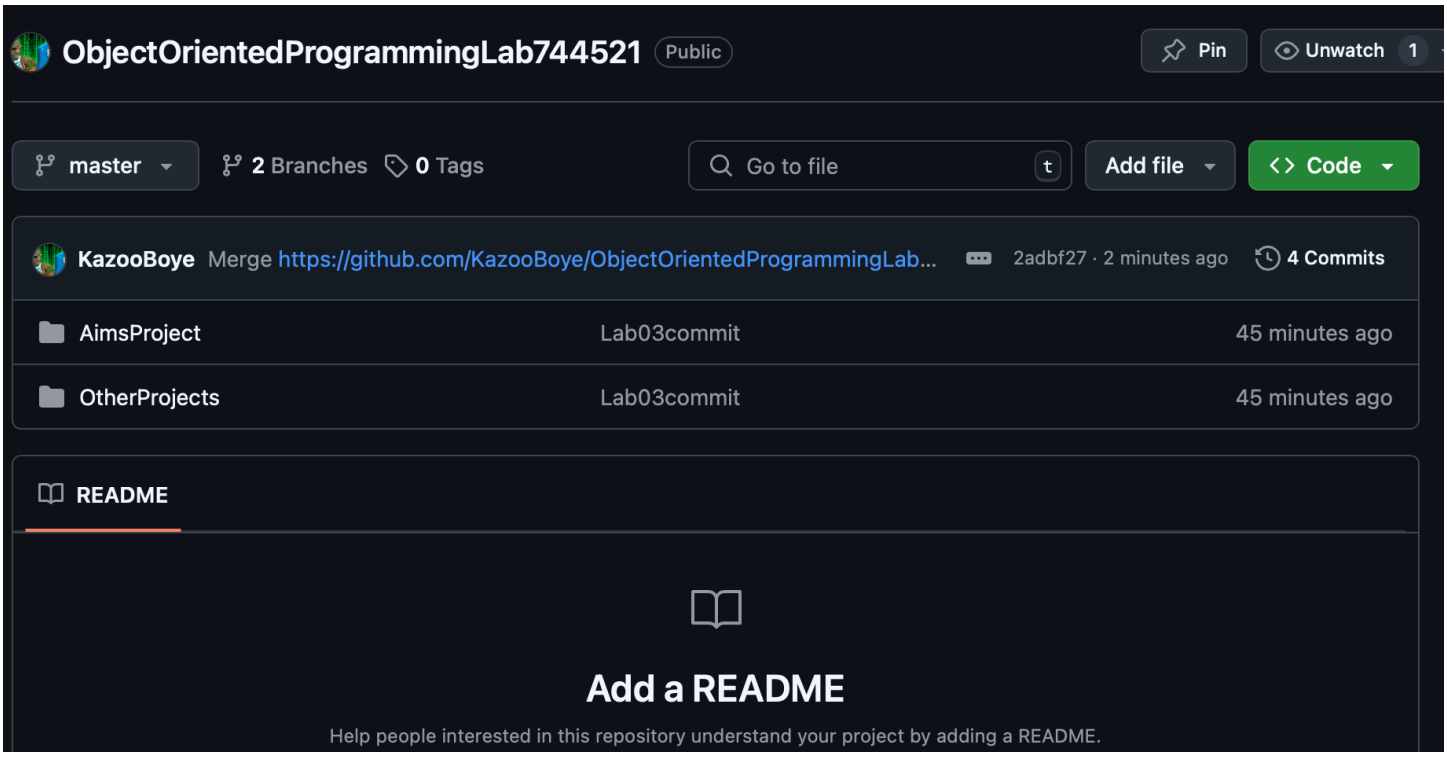

Figure 37. Confirm changes have been made to 'master'