

Report Object-Oriented Programming

Lab 03: Basic Object-Oriented Techniques

Student's name: Lê Tùng Lâm

Student's ID: 20235962

2. Working with method overloading

2.1. Overloading by differing types of parameter

```
public void addDigitalVideoDisc(DigitalVideoDisc disc) { 6 usages new *
    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = disc;
        qtyOrdered += 1;
        System.out.println("The disc has been added");
    } else System.out.println("The cart is almost full");
}

public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList) { 1 usage new *
    if (dvdList.length > MAX_NUMBERS_ORDERED) {
        System.out.println("The cart is almost full");
    } else {
        for (int i = 0; i < dvdList.length; i++) {
            itemsOrdered[qtyOrdered] = dvdList[i];
            qtyOrdered += 1;
            System.out.println(dvdList[i].getTitle() + " has been added");
        }
    }
}
```

2.2. Overloading by differing the number of parameters

```
public void addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2) {  
    DigitalVideoDisc [] dvdList = {dvd1, dvd2};  
    addDigitalVideoDisc(dvdList);  
}
```

3. Passing parameter

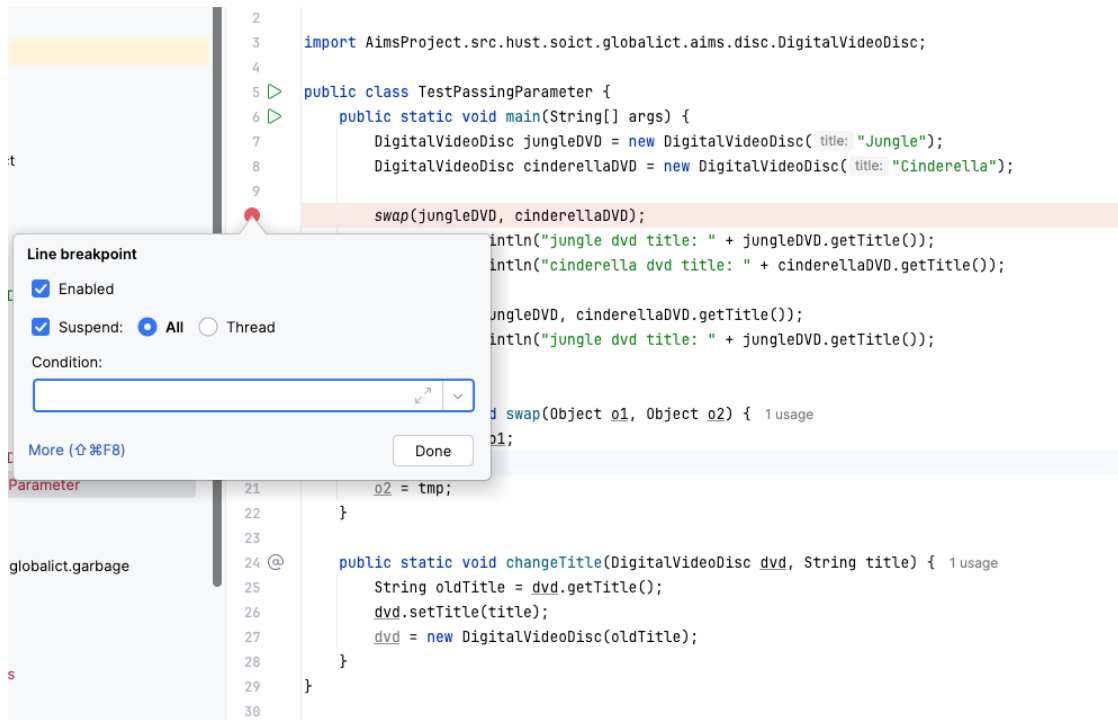
```
TestPassingParameter.java x  
1 package AimsProject.src.hust.soict.globalict.test.disc;  
2  
3 import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;  
4  
5 public class TestPassingParameter {  
6     public static void main(String[] args) {  
7         DigitalVideoDisc jungleDVD = new DigitalVideoDisc( title: "Jungle");  
8         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc( title: "Cinderella");  
9  
10        swap(jungleDVD, cinderellaDVD);  
11        System.out.println("jungle dvd title: " + jungleDVD.getTitle());  
12        System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());  
13  
14        changeTitle(jungleDVD, cinderellaDVD.getTitle());  
15        System.out.println("jungle dvd title: " + jungleDVD.getTitle());  
16    }  
17  
18    public static void swap(Object o1, Object o2) { 1 usage  
19        Object tmp = o1;  
20        o1 = o2;  
21        o2 = tmp;  
22    }  
23  
24    public static void changeTitle(DigitalVideoDisc dvd, String title) { 1 usage  
25        String oldTitle = dvd.getTitle();  
26        dvd.setTitle(title);  
27        dvd = new DigitalVideoDisc(oldTitle);  
28    }  
29 }
```

Result:

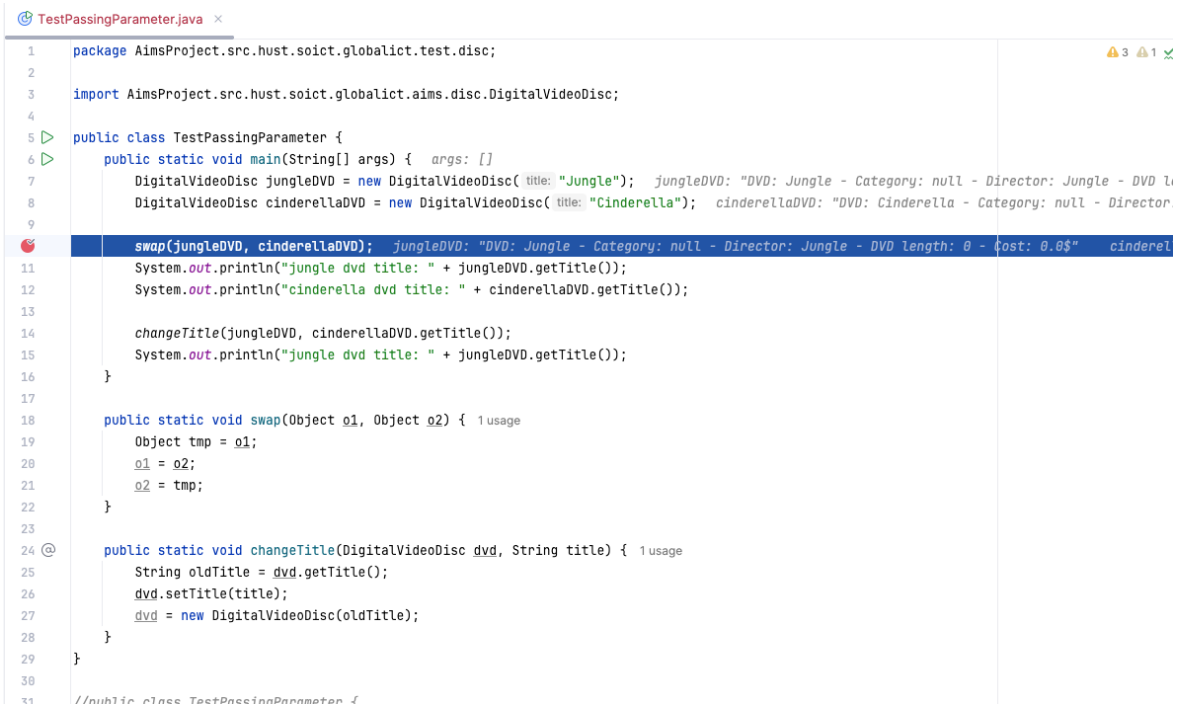
```
./TestPassingParameter.java:15: warning: no warning for parameter 'title' in method  
jungle dvd title: Jungle  
cinderella dvd title: Cinderella  
jungle dvd title: Jungle
```

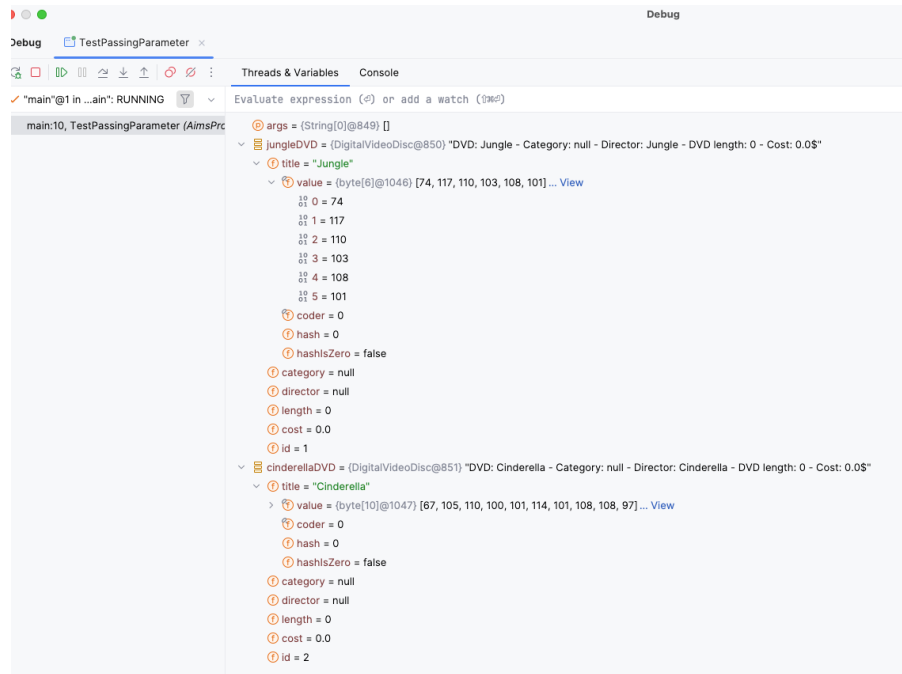
4. Use debug run

4.1. Debug run for the swap method of TestPassingParameter



4.2. Run in Debug mode:





5. Classifier Member and Instance Member

```

1 package AimsProject.src.hust.soict.globalict.aims.disc;
2
3 public class DigitalVideoDisc { 45 usages new *
4     private String title; 10 usages
5     private String category; 6 usages
6     private String director; 4 usages
7     private int length; 4 usages
8     private float cost; 6 usages
9     private static int nbDigitalVideoDiscs = 0; 4 usages
10    private int id; 5 usages
11
12    // Create a DVD object by title
13    public DigitalVideoDisc(String title) { 3 usages new *
14        this.title = title;
15        this.id = ++nbDigitalVideoDiscs;
16    }
17
18    // Create a DVD object by category, title and cost
19    public DigitalVideoDisc(String category, String title, float cost) { 3 usages new *
20        this.category = category;
21        this.title = title;
22        this.cost = cost;
23        this.id = ++nbDigitalVideoDiscs;
24    }
25
26    // Create a DVD object by director, category, title and cost
27    public DigitalVideoDisc(String director, String category, String title, float cost) { no usages new *
28        this.director = director;
29        this.category = category;
30        this.title = title;
31        this.cost = cost;
32        this.id = ++nbDigitalVideoDiscs;
33    }
34
35    // Create a DVD object by all attributes: title, category, director, length and cost
36    public DigitalVideoDisc(String title, String category, String director, int length, float cost) { 6 usages new *
37        this.title = title;
38        this.category = category;
39        this.director = director;
40        this.length = length;
41        this.cost = cost;
42        this.id = ++nbDigitalVideoDiscs;
43    }
44

```

6. Open the Cart class

```
---
114
115     public void print() { 1 usage new *
116         System.out.println("*****CART*****");
117         System.out.println("Ordered Items:");
118         for (int i = 0; i < qtyOrdered; i++) {
119             System.out.println(i+1 + ". " + itemsOrdered[i]);
120         }
121         System.out.println("Total cost: " + totalCost());
122         System.out.println("*****");
123     }
124 }
125

public void searchByID(int id) { 2 usages new *
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].getId() == id) {
            System.out.println(itemsOrdered[i].getTitle() + " has been found");
            found = true;
        }
    }
    if (!found) System.out.println("Not found the DVD with the ID " + id);
}

public void searchByTitle(String title) { 3 usages new *
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].isMatch(title)) {
            System.out.println(itemsOrdered[i].getTitle() + " has been found");
            found = true;
        }
    }
    if (!found) System.out.println("Not found the DVD with the title " + title);
}
```

```

    public boolean isMatch(String title) { 1 usage new *
        return this.title.toLowerCase().contains(title.toLowerCase());
    }

    @Override new *
    public String toString() {
        return "DVD: " + this.title +
            " - Category: " + this.category +
            " - Director: " + this.title +
            " - DVD length: " + this.length +
            " - Cost: " + this.cost + "$";
    }
}

```

CartTest:

```

CartTest.java x
1 package AimsProject.src.hust.soict.globalict.test.cart;
2
3 import AimsProject.src.hust.soict.globalict.aims.cart.Cart;
4 import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;
5
6 public class CartTest {
7     public static void main(String[] args) {
8         Cart cart = new Cart();
9
10        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation",
11            director: "Roger Allers", length: 87, cost: 19.95f);
12        cart.addDigitalVideoDisc(dvd1);
13
14        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction",
15            director: "George Lucas", length: 87, cost: 24.95f);
16        cart.addDigitalVideoDisc(dvd2);
17
18        DigitalVideoDisc dvd3 = new DigitalVideoDisc( category: "Aladdin", title: "Animation", cost: 18.99f);
19        cart.addDigitalVideoDisc(dvd3);
20
21        cart.print();
22
23        cart.searchByID(1);
24        cart.searchByTitle("Star Wars");
25        cart.searchByID(5);
26        cart.searchByTitle("Sherlock Holmes");
27        cart.searchByTitle("Lion");
28    }
29 }

```

Result:

```
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java -jar target/applications/interactio
The disc has been added
The disc has been added
The disc has been added
*****CART*****
Ordered Items:
1. DVD: The Lion King - Category: Animation - Director: The Lion King - DVD length: 87 - Cost: 19.95$
2. DVD: Star Wars - Category: Science Fiction - Director: Star Wars - DVD length: 87 - Cost: 24.95$
3. DVD: Animation - Category: Aladin - Director: Animation - DVD length: 0 - Cost: 18.99$
Total cost: 63.89
*****
The Lion King has been found
Star Wars has been found
Not found the DVD with the ID 5
Not found the DVD with the title Sherlock Holmes
The Lion King has been found
```

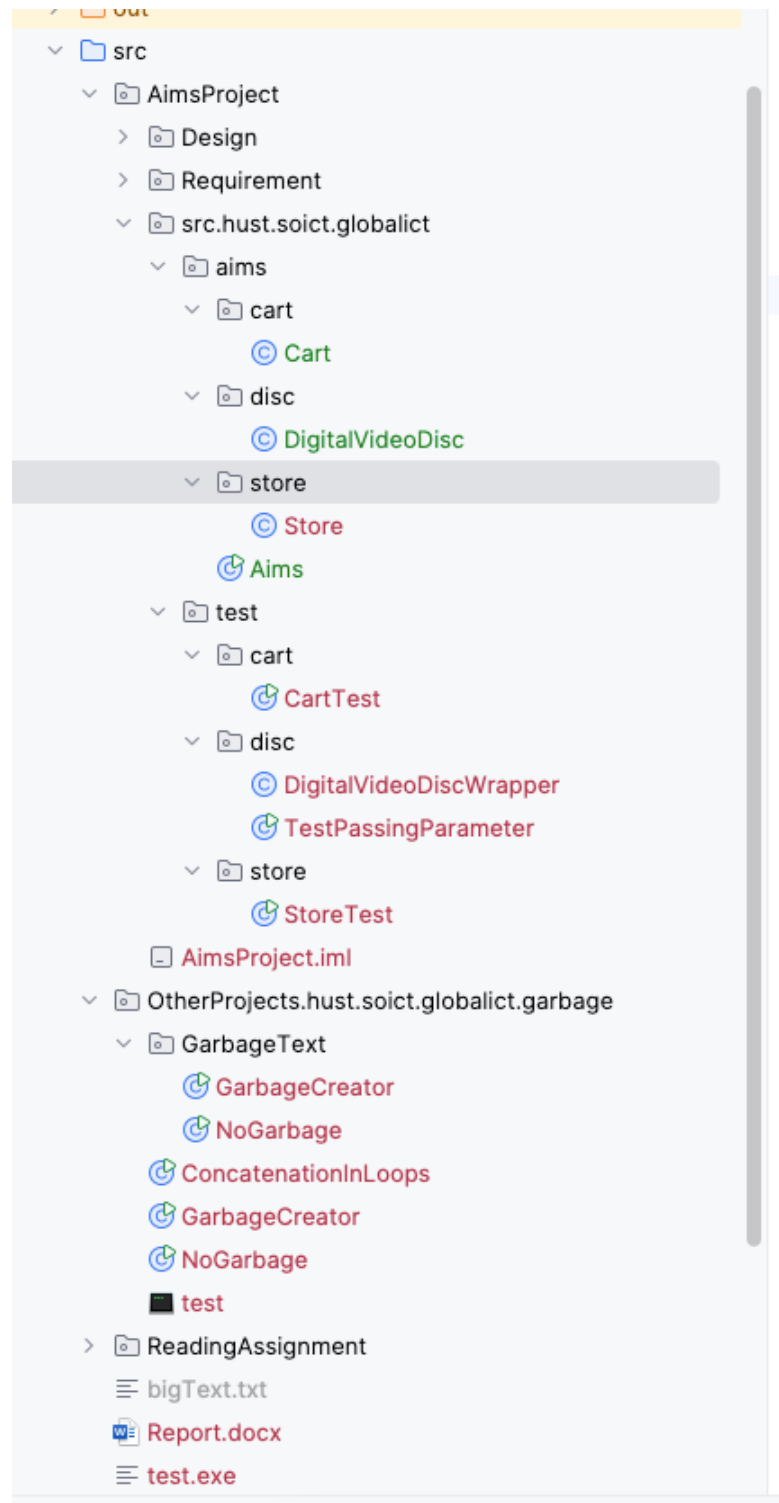
7. Implement the Store class

```
CartTest.java  Store.java x
1 package AimsProject.src.hust.soict.globalict.aims.store;
2
3 import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;
4
5 import java.util.ArrayList;
6 import java.util.List;
7
8 public class Store { 3 usages
9     private List<DigitalVideoDisc> listItems = new ArrayList<DigitalVideoDisc>(); 4 usages
10
11     public void addDVD(DigitalVideoDisc dvd) { 3 usages
12         listItems.add(dvd);
13         System.out.println(dvd.getTitle() + " has been added to the store.");
14     }
15
16     public void removeDVD(DigitalVideoDisc dvd) { 1 usage
17         if (listItems.remove(dvd)) {
18             System.out.println(dvd.getTitle() + " has been removed");
19         } else {
20             System.out.println(" Not found the disc with title " + dvd.getTitle());
21         }
22     }
23
24     public void print() { 2 usages
25         System.out.println("List of DVDs: ");
26         for (int i = 0; i < listItems.size(); i++) {
27             System.out.println(i+1 + ". " + listItems.get(i));
28         }
29     }
30 }
31
```

CartTest.java Store.java StoreTest.java x

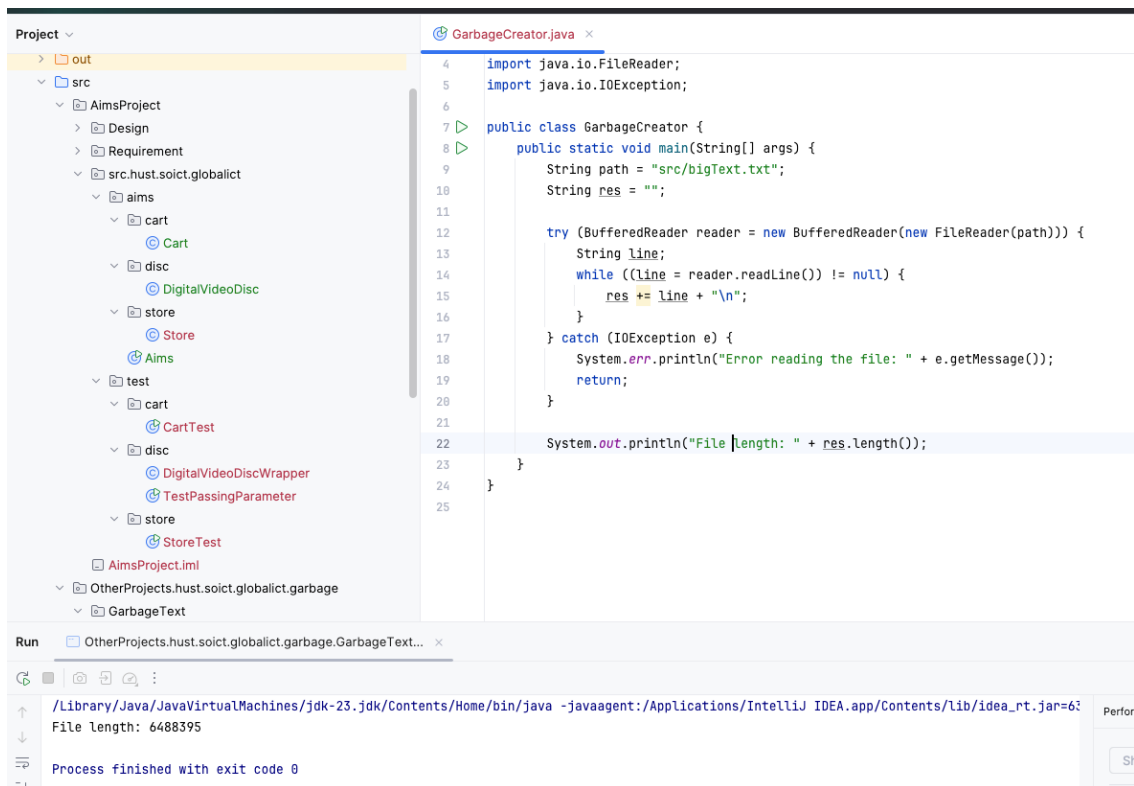
```
1 package AimsProject.src.hust.soict.globalict.test.store;
2
3 import AimsProject.src.hust.soict.globalict.aims.disc.DigitalVideoDisc;
4 import AimsProject.src.hust.soict.globalict.aims.store.Store;
5
6 public class StoreTest {
7     public static void main(String[] args) {
8         Store store = new Store();
9
10        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation",
11            director: "Roger Allers", length: 87, cost: 19.95f);
12        store.addDVD(dvd1);
13
14        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star War", category: "Science Fiction",
15            director: "George Lucas", length: 87, cost: 24.95f);
16        store.addDVD(dvd2);
17
18        DigitalVideoDisc dvd3 = new DigitalVideoDisc( category: "Aladin", title: "Animation", cost: 18.99f);
19        store.addDVD(dvd3);
20
21        store.print();
22
23        store.removeDVD(dvd1);
24
25        store.print();
26    }
27 }
28
```


8. Re-organize your projects



9. String, StringBuilder and StringBuffer

GarbageCreator:

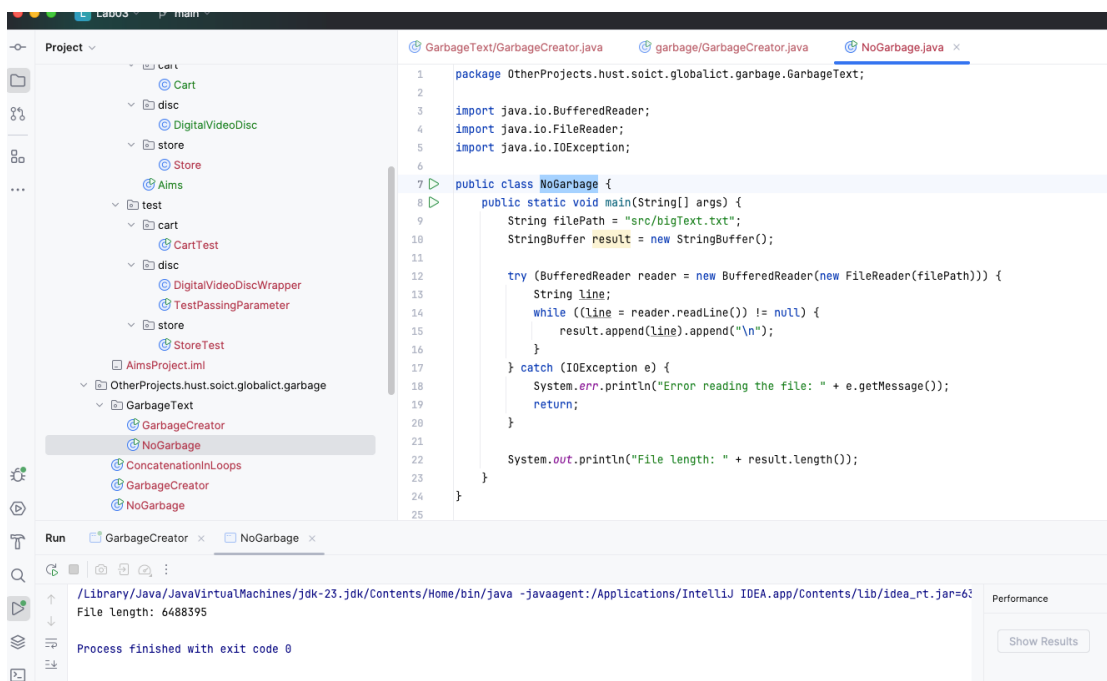


The screenshot shows the IntelliJ IDEA IDE with the `GarbageCreator.java` file open. The project structure on the left includes a `src` directory with `src.hust.soict.globalict` and `src.hust.soict.globalict.garbage` packages. The `GarbageCreator.java` file contains the following code:

```
4 import java.io.FileReader;
5 import java.io.IOException;
6
7 public class GarbageCreator {
8     public static void main(String[] args) {
9         String path = "src/bigText.txt";
10        String res = "";
11
12        try (BufferedReader reader = new BufferedReader(new FileReader(path))) {
13            String line;
14            while ((line = reader.readLine()) != null) {
15                res += line + "\n";
16            }
17        } catch (IOException e) {
18            System.err.println("Error reading the file: " + e.getMessage());
19            return;
20        }
21
22        System.out.println("File length: " + res.length());
23    }
24 }
25
```

The Run tab at the bottom shows the execution of the `GarbageCreator` class, with the output: `File length: 6488395`. The process finished with exit code 0.

NoGarbage:



The screenshot shows the IntelliJ IDEA IDE with the `NoGarbage.java` file open. The project structure on the left includes a `src` directory with `src.hust.soict.globalict.garbage` package. The `NoGarbage.java` file contains the following code:

```
1 package OtherProjects.hust.soict.globalict.garbage.NoGarbage;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.io.IOException;
6
7 public class NoGarbage {
8     public static void main(String[] args) {
9         String filePath = "src/bigText.txt";
10        StringBuffer result = new StringBuffer();
11
12        try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
13            String line;
14            while ((line = reader.readLine()) != null) {
15                result.append(line).append("\n");
16            }
17        } catch (IOException e) {
18            System.err.println("Error reading the file: " + e.getMessage());
19            return;
20        }
21
22        System.out.println("File length: " + result.length());
23    }
24 }
25
```

The Run tab at the bottom shows the execution of the `NoGarbage` class, with the output: `File length: 6488395`. The process finished with exit code 0.

ConcatenationInLoops:

```
ConcatenationInLoops.java x
6 import java.util.Random;
7
8 public class ConcatenationInLoops {
9     public static void main(String[] args) throws IOException {
10         Random r = new Random( seed: 123);
11
12         // + operator
13         long start = System.currentTimeMillis();
14         String s = "";
15         for (int i = 0; i < 65536; i++) {
16             s += r.nextInt( bound: 2);
17         }
18         System.out.println(System.currentTimeMillis() - start);
19
20         // StringBuffer
21         r = new Random( seed: 123);
22         start = System.currentTimeMillis();
23         StringBuffer sb = new StringBuffer();
24         for (int i = 0; i < 65536; i++) {
25             sb.append(r.nextInt( bound: 2));
26         }
27         s = sb.toString();
28         System.out.println(System.currentTimeMillis() - start);
29     }
}
```

```
30 // StringBuilder
31 r = new Random( seed: 123);
32 start = System.currentTimeMillis();
33 StringBuilder sb2 = new StringBuilder();
34 for (int i = 0; i < 65536; i++) {
35     sb2.append(r.nextInt( bound: 2));
36 }
37 s = sb2.toString();
38 System.out.println(System.currentTimeMillis() - start);
39
40 String filename = "src/test.exe";
41 byte[] inputBytes = { 0 };
42 long startTime, endTime;
43
44 inputBytes = Files.readAllBytes(Paths.get(filename));
45 startTime = System.currentTimeMillis();
46 // String outputString = "";
47 // for (byte b : inputBytes) {
48 //     outputString += b;
49 // }
50 StringBuilder outputStringBuilder = new StringBuilder();
51 for (byte b : inputBytes) {
52     outputStringBuilder.append((char) b);
53 }
54 endTime = System.currentTimeMillis();
55 System.out.println(endTime - startTime);
56 }
57 }
58
```

Result:

```

/Library/Java/JavaVirtualMachines/jdk-23.jdk/Contents/Home/bin/j
1273
9
5
32

Process finished with exit code 0

```