

## 2. Working with overloading

### 2.1

```
// OVERLOADING (ADD dvdList to Cart)

// public void addDigitalVideoDisc (DigitalVideoDisc[] dvdList){
//     int i = 0;
//     for (DigitalVideoDisc D : dvdList) {
//         itemsOrdered.add(D);
//         System.out.println("The disc has been added");
//         if (itemsOrdered.size() == 20){
//             System.out.println("The cart is full");
//             break;
//         }
//     }
//     if (itemsOrdered.size() == 20){
//         System.out.println("The cart is full");
//     }
// }
```

```
// OVERLOADING (ADD an arbitrary numbers of dvds to Cart)
public void addDigitalVideoDisc (DigitalVideoDisc... dvdList){
    int i = 0;
    for (DigitalVideoDisc D : dvdList) {
        itemsOrdered.add(D);
        System.out.println("The disc has been added");
        if (itemsOrdered.size() == 20){
            System.out.println("The cart is full");
            break;
        }
    }
    if (itemsOrdered.size() == 20){
        System.out.println("The cart is full");
    }
}
```

The second one is better since I don't need to make a new array for Dvdlist

2.2

```
public void addDigitalVideoDisc (DigitalVideoDisc d1, DigitalVideoDisc d2){
    if (itemsOrdered.size() + 2 <= 20) {
        itemsOrdered.add(d1);
        System.out.println("The disc has been added");
        itemsOrdered.add(d2);
        System.out.println("The disc has been added");
    }
    else if (itemsOrdered.size() == 19) {
        itemsOrdered.add(d1);
        System.out.println("The disc has been added");
    }
    if (itemsOrdered.size() == 20){
        System.out.println("The cart is full");
    }
}
```

3.

```

1 package hust.soict.dsai.test.disc;
2
3 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter {
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
9         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
10
11         swap(jungleDVD, cinderellaDVD);
12         System.out.println("Jungle DVD title: " + jungleDVD.getTitle());
13         System.out.println("Cinderella DVD title: " + cinderellaDVD.getTitle());
14
15         changeTitle(jungleDVD, cinderellaDVD.getTitle());
16         System.out.println("Jungle DVD title: " + jungleDVD.getTitle());
17     }
18
19     public static void swap(Object o1, Object o2) {
20         Object tmp = o1;
21         o1 = o2;
22         o2 = tmp;
23     }
24
25     public static void changeTitle(DigitalVideoDisc dvd, String title) {
26         String oldTitle = dvd.getTitle();
27         dvd.setTitle(title);
28         dvd = new DigitalVideoDisc(oldTitle);
29     }
30
31     // Uncommented implementation of swapping DigitalVideoDisc objects
32     public static void swap(DigitalVideoDisc o1, DigitalVideoDisc o2) {
33         int len = o1.getLength();
34         String title = o1.getTitle();
35         String direc = o1.getDirector();
36         float cost = o1.getCost();
37         String cate = o1.getCategory();
38
39         o1.setCategory(o2.getCategory());
40         o1.setCost(o2.getCost());
41         o1.setDirectory(o2.getDirector());
42         o1.setLength(o2.getLength());
43         o1.setTitle(o2.getTitle());
44
45         o2.setCategory(cate);
46         o2.setCost(cost);
47         o2.setDirectory(direc);
48         o2.setLength(len);
49         o2.setTitle(title);
50     }

```

## Questions:

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

- Java is a pass-by-value programming language even for object references => we don't change it object by object

The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Shows the project structure with 'TestPassingParameter' as the main application.
- Source Editor:** Displays the code for 'TestPassingParameter.java'. The code includes a package declaration, imports, and two methods: 'main' and 'swap'. The 'main' method creates two 'DigitalVideoDisc' objects, 'jungleDVD' and 'cinderellaDVD', and calls the 'swap' method. The 'swap' method is a generic method that swaps the 'title' and 'category' attributes of two 'DigitalVideoDisc' objects.
- Variables View:** Located on the right, it shows the current state of variables. The 'jungleDVD' variable is of type 'DigitalVideoDisc' and has a 'title' of 'Jungle' and a 'category' of 'DVD'. The 'cinderellaDVD' variable is also of type 'DigitalVideoDisc' and has a 'title' of 'Cinderella' and a 'category' of 'DVD'.
- Console View:** At the bottom, it shows the output of the program, which includes the titles and categories of the two DVD objects before and after the swap operation.

eclipse-workspace - OOPLAB/Lab\_02/AimProjects/src/main/TestPassingParameter.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug Project Explorer

TestPassingParameter (Java Application)

- main.TestPassingParameter at localhost:64897
  - Thread [main] (Suspended)
    - TestPassingParameter.swap(Object, Object) line: 18
      - TestPassingParameter.main(String[]) line: 9

C:\Program Files\Java\jdk-22\bin\javaw.exe (Nov 24, 2024, 10:13:20AM)

```
1 package main;
2
3 public class TestPassingParameter {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
7         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
8
9         swap(jungleDVD, cinderellaDVD);
10        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
11        System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());
12
13        changeTitle(jungleDVD, cinderellaDVD.getTitle());
14        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
15    }
16
17    public static void swap(Object o1, Object o2) {
18        Object tmp = o1;
19        o1 = o2;
20        o2 = tmp;
21    }
22
23    // public static void swap(DigitalVideoDisc o1, DigitalVideoDisc o2) {
24    //     int len = o1.getLength();
25    //     String title = o1.getTitle();
26    //     String dir = o1.getDirectory();
27    //     float cost = o1.getCost();
28    //     String cat = o1.getCategory();
29    //
30    //     o1.setCategory(o2.getCategory());
31    //     o1.setCost(o2.getCost());
32    //     o1.setDirectory(o2.getDirectory());
33    //     o1.setLength(o2.getLength());
34    //     o1.setTitle(o2.getTitle());
35    // }
```

Variables Breakpoints Expressions

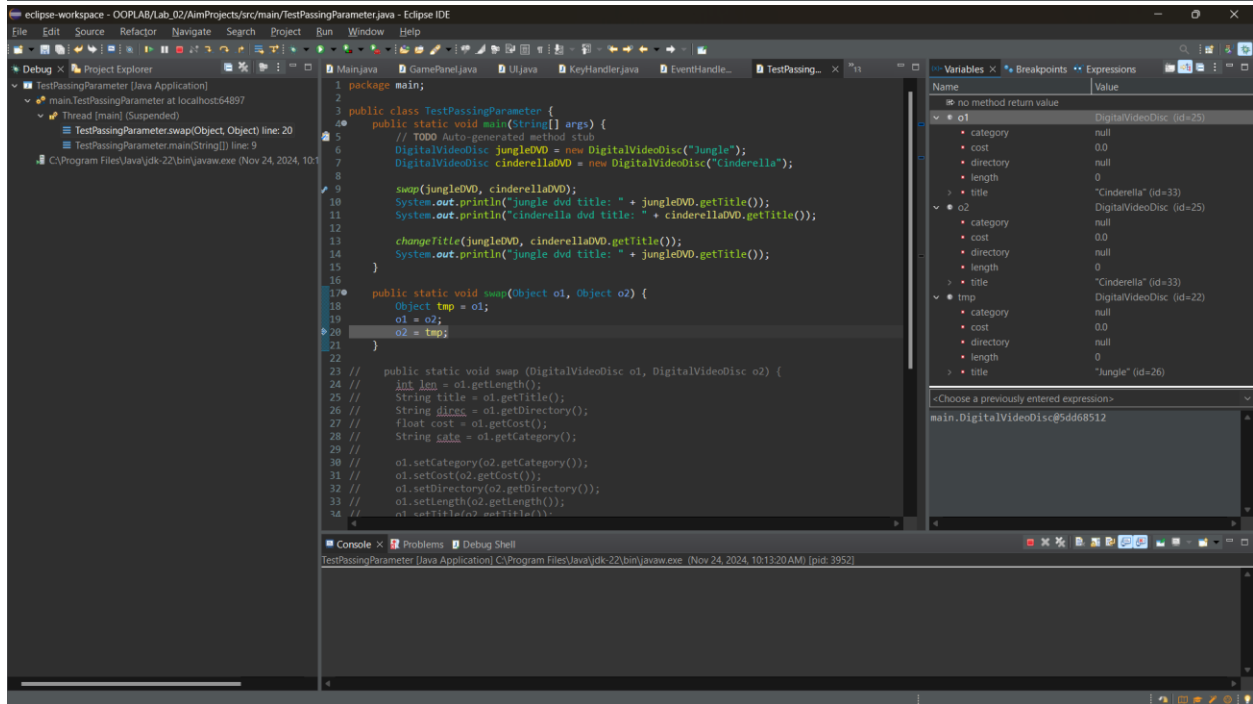
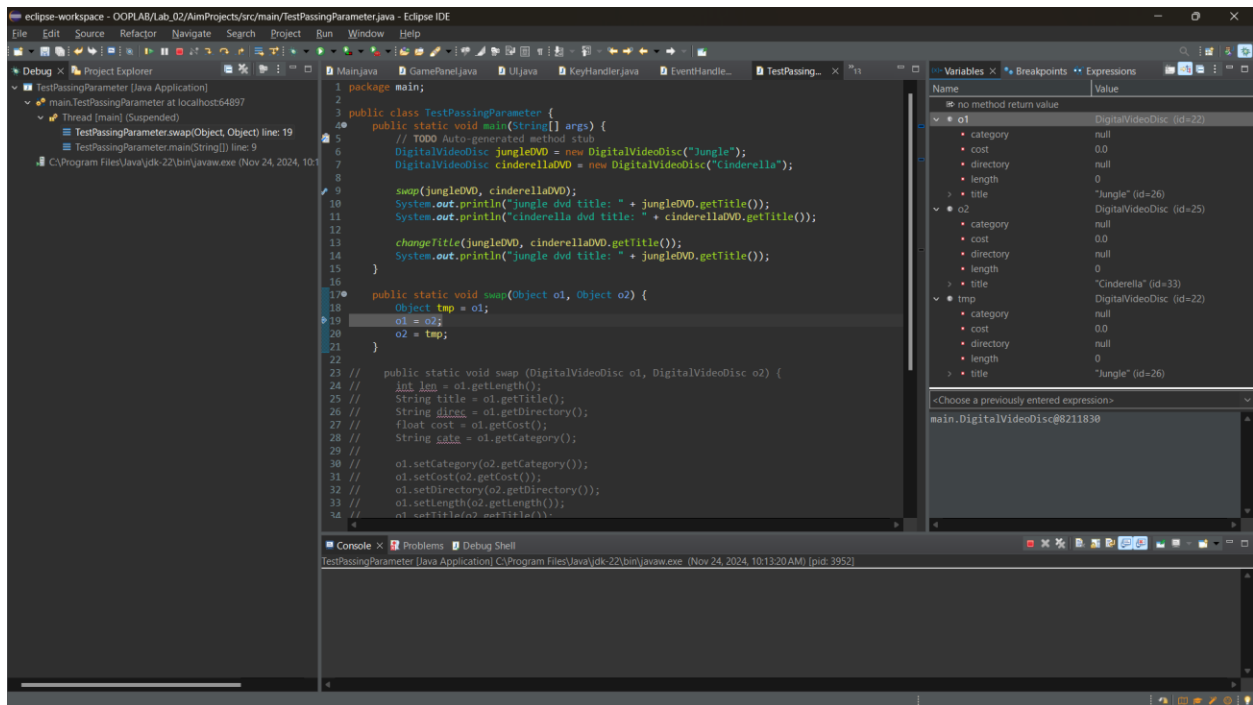
Name	Value
no method return value	
o1	DigitalVideoDisc (id=22)
category	null
cost	0.0
directory	null
length	0
title	"Jungle" (id=26)
o2	DigitalVideoDisc (id=25)
category	null
cost	0.0
directory	null
length	0
title	"Cinderella" (id=33)

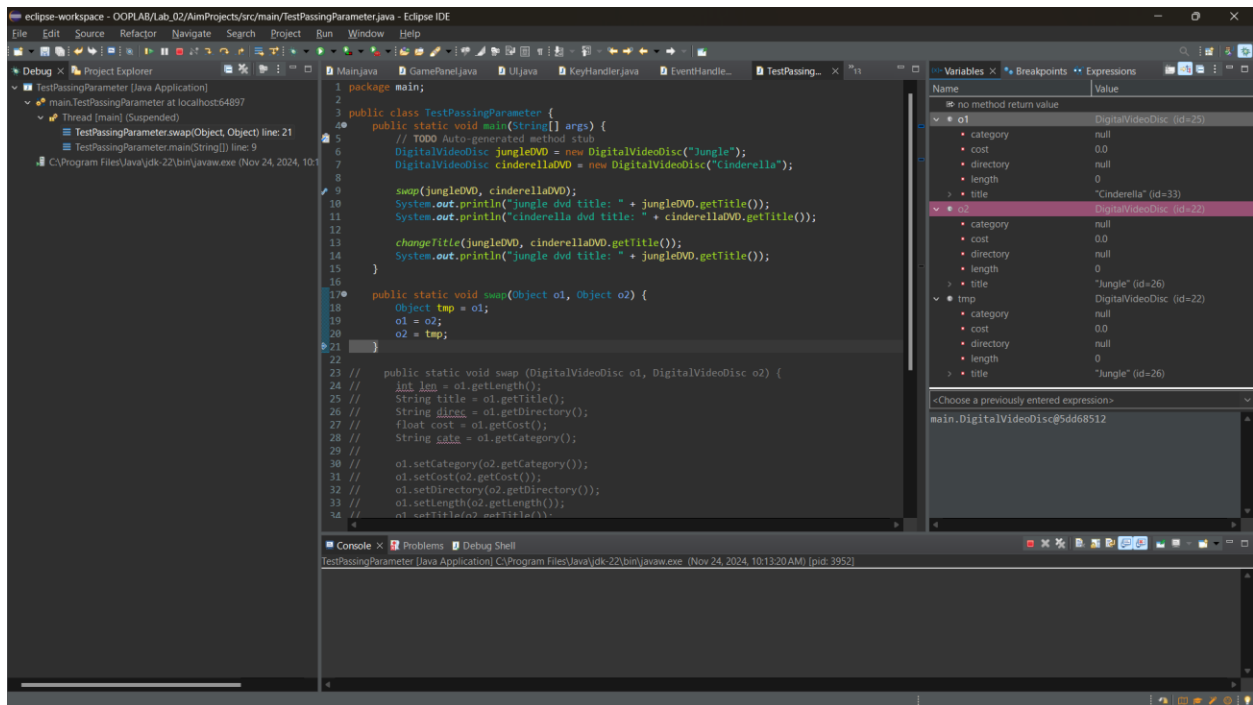
<Choose a previously entered expression>

main.DigitalVideoDisc@8211830

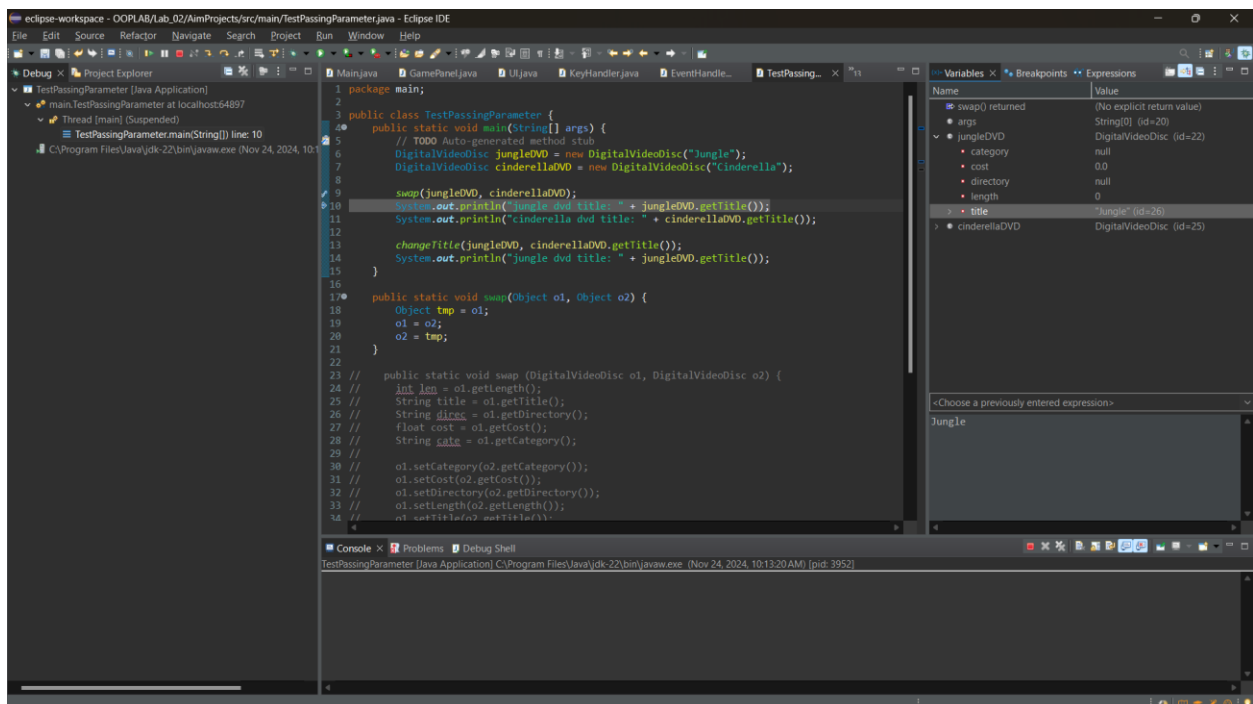
Console Problems Debug Shell

TestPassingParameter (Java Application) C:\Program Files\Java\jdk-22\bin\javaw.exe (Nov 24, 2024, 10:13:20AM) [pid: 3952]





## Result



## 5. Classifier member and instance member



```
1 package hust.soict.dsai.test.disc;
2 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
3
4 public class DigitalVideoDiscWrapper {
5     DigitalVideoDisc dvd;
6     DigitalVideoDiscWrapper(DigitalVideoDisc dvd) {
7         super();
8
9         this.dvd = dvd;
10    }
11 }
```

```

1 package hust.soict.dsai.test.disc;
2 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
3
4 public class TestPassingParameter {
5
6     Run | Debug
7     public static void main(String[] args) {
8         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
9         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
10
11         DigitalVideoDiscWrapper wjungleDVD = new DigitalVideoDiscWrapper(jungleDVD);
12         DigitalVideoDiscWrapper wcinderellaDVD = new DigitalVideoDiscWrapper(cinderellaDVD);
13
14         // Wrong swap() function
15         swap(jungleDVD, cinderellaDVD);
16         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
17         System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());
18
19         // Correct swap() function
20         swap(wjungleDVD, wcinderellaDVD);
21         System.out.println("Correct swapped jungle dvd title: " + wjungleDVD.dvd.getTitle());
22         System.out.println("Correct swapped cinderella dvd title: " + wcinderellaDVD.dvd.getTitle());
23     }
24
25     // Wrong swap() function
26     public static void swap(Object o1, Object o2) {
27         Object tmp = o1;
28         o1 = o2;
29         o2 = tmp;
30     }
31
32     // Correct swap() function
33     public static void swap(DigitalVideoDiscWrapper o1, DigitalVideoDiscWrapper o2) {
34         DigitalVideoDisc tmp = o1.dvd;
35         o1.dvd = o2.dvd;
36         o2.dvd = tmp;
37     }
38
39     public static void changeTitle(DigitalVideoDisc dvd, String title) {
40         String oldTitle = dvd.getTitle();
41         dvd.setTitle(title);
42         dvd = new DigitalVideoDisc(oldTitle);
43     }
44 }

```

## 6. Open cart class

```

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

public void printCart() {
    System.out.println("*****CART*****");
    System.out.println("Ordered Items:");
    int count = 1;
    for (DigitalVideoDisc item : itemsOrdered) {
        System.out.print(count + ". ");
        System.out.println(item.toString());
        count++;
    }
    System.out.print("Total cost: ");
    System.out.println(this.totalCost());
}

```

```

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("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        cart.addDigitalVideoDisc(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
        cart.addDigitalVideoDisc(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
        // hust.soict.dsai.aims.disc.DigitalVideoDisc
        Cart cart = hust.soict.dsai.test.cart.CartTest.main(String[])
        cart.print();
    }
}

```

## 7. implement store class

```

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

```
1 package hust.soict.dsai.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         long start = System.currentTimeMillis();
9         String s = "";
10        for (int i = 0; i < 65536; i++) {
11            s += r.nextInt(2);
12        }
13        System.out.println(System.currentTimeMillis() - start); //
14
15        r = new Random(123);
16        start = System.currentTimeMillis();
17        StringBuilder sb = new StringBuilder();
18        for (int i = 0; i < 65536; i++) {
19            sb.append(r.nextInt(2));
20        }
21        s = sb.toString();
22        System.out.println(System.currentTimeMillis() - start);
23    }
```

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

import java.io.IOException;

public class NoGarbage {
    public static void main(String[] args) {
        String filename = "test.exe"; // Path to a large file
        byte[] inputBytes = { 0 };

        long startTime, endTime;

        try {
            // Read all bytes from the file
            inputBytes = Files.readAllBytes(Paths.get(filename));
            startTime = System.currentTimeMillis();

            // Efficient concatenation using StringBuilder
            StringBuilder outputStringBuffer = new StringBuilder();
            for (byte b : inputBytes) {
                outputStringBuffer.append((char) b);
            }

            endTime = System.currentTimeMillis();
            System.out.println("Time taken with StringBuilder: " + (endTime - startTime) + " ms");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

1 package hust.soict.dsai.garbage;
2
3 import java.io.IOException;
4
5
6
7 public class GarbageCreator {
8     public static void main(String[] args) {
9         String filename = "test.exe"; // Path to a large file
10        byte[] inputBytes = { 0 };
11        String outputString = "";
12        long startTime, endTime;
13
14        try {
15            // Read all bytes from the file
16            inputBytes = Files.readAllBytes(Paths.get(filename));
17            startTime = System.currentTimeMillis();
18
19            // Inefficient concatenation using "+"
20            for (byte b : inputBytes) {
21                outputString += (char) b;
22            }
23
24            endTime = System.currentTimeMillis();
25            System.out.println("Time taken with String concatenation: " + (endTime - startTime) + " ms");
26        } catch (IOException e) {
27            e.printStackTrace();
28        }
29    }
30 }

```