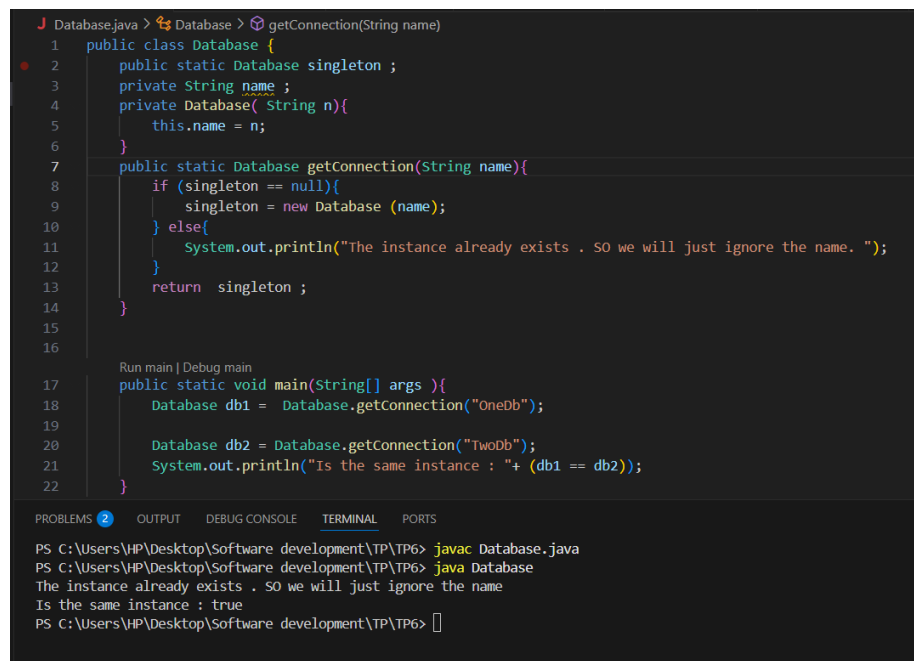


# Design Patterns TP

El Gourji Nasreddine

October 2025

## 1 EX 1 :



```
J Database.java > Database > getConnection(String name)
1 public class Database {
2     public static Database singleton ;
3     private String name ;
4     private Database( String n){
5         this.name = n;
6     }
7     public static Database getConnection(String name){
8         if (singleton == null){
9             singleton = new Database (name);
10        } else{
11            System.out.println("The instance already exists . SO we will just ignore the name. ");
12        }
13        return singleton ;
14    }
15
16
17    Run main | Debug main
18    public static void main(String[] args ){
19        Database db1 = Database.getConnection("OneDb");
20
21        Database db2 = Database.getConnection("TwoDb");
22        System.out.println("Is the same instance : "+ (db1 == db2));
23    }
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

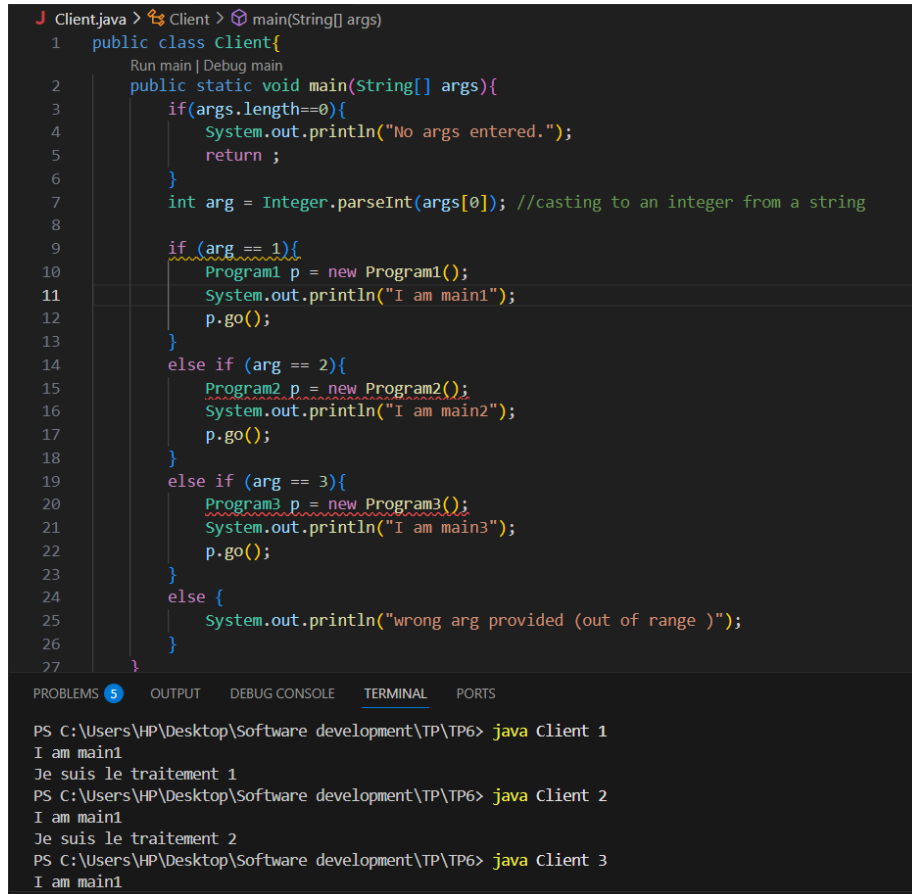
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\Desktop\Software development\TP\TP6> javac Database.java
PS C:\Users\HP\Desktop\Software development\TP\TP6> java Database
The instance already exists . SO we will just ignore the name
Is the same instance : true
PS C:\Users\HP\Desktop\Software development\TP\TP6> 
```

Figure 1: Code and output of the first exercise.

## 2 EX 2 :

### 2.1 Naive Solution :



```
J Client.java > Client > main(String[] args)
1 public class Client{
    Run main | Debug main
2 public static void main(String[] args){
3     if(args.length==0){
4         System.out.println("No args entered.");
5         return ;
6     }
7     int arg = Integer.parseInt(args[0]); //casting to an integer from a string
8
9     if (arg == 1){
10        Program1 p = new Program1();
11        System.out.println("I am main1");
12        p.go();
13    }
14    else if (arg == 2){
15        Program2 p = new Program2();
16        System.out.println("I am main2");
17        p.go();
18    }
19    else if (arg == 3){
20        Program3 p = new Program3();
21        System.out.println("I am main3");
22        p.go();
23    }
24    else {
25        System.out.println("wrong arg provided (out of range )");
26    }
27 }
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\Desktop\Software development\TP\TP6> java Client 1
I am main1
Je suis le traitement 1
PS C:\Users\HP\Desktop\Software development\TP\TP6> java Client 2
I am main1
Je suis le traitement 2
PS C:\Users\HP\Desktop\Software development\TP\TP6> java Client 3
I am main1
```

Figure 2: Code and output of the second exercise part 1.

## 2.2 Apply Design Patterns:

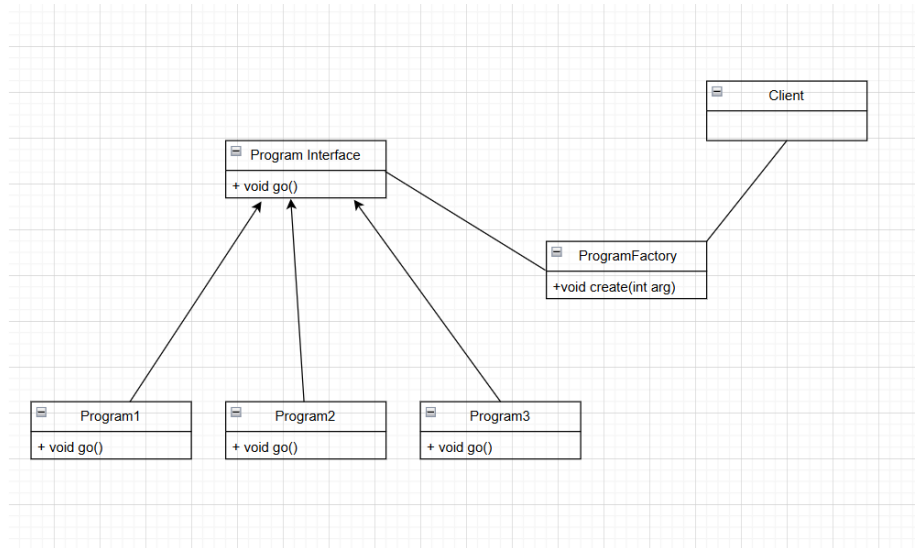
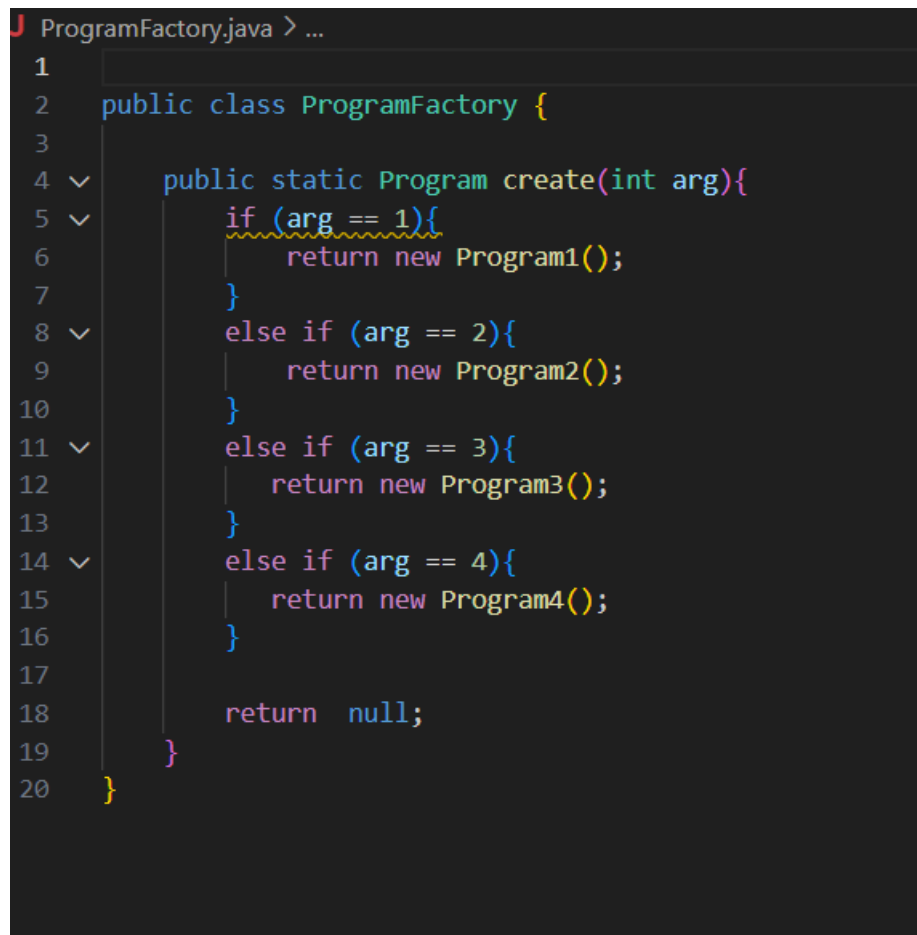


Figure 3: Class diagram



```
1
2 public class ProgramFactory {
3
4     public static Program create(int arg){
5         if (arg == 1){
6             return new Program1();
7         }
8         else if (arg == 2){
9             return new Program2();
10        }
11        else if (arg == 3){
12            return new Program3();
13        }
14        else if (arg == 4){
15            return new Program4();
16        }
17
18        return null;
19    }
20 }
```

Figure 4: class ProgramFactory

```

J Program1.java > Program1
1 public class Program1 implements Program{
2     public Program1 (){
3         // The constructor does nothing .
4     }
5     public void go (){
6         System . out . println ("Je suis le traitement 1") ;
7     }
8 }

```

Figure 5: Class Program1 (Same implementation for the Program2,Program3 and Program4.)

```

1 public interface Program {
2     void go();
3 }

```

Figure 6: interface Program

```

J Client.java > Client > main(String[] args)
1 public class Client{
2     Run main | Debug main
3     public static void main(String[] args){
4         if(args.length==0){
5             System.out.println("No args entered.");
6             return ;
7         }
8         int arg = Integer.parseInt(args[0]); //casting to an integer from a string
9         Program p = ProgramFactory.create(arg);
10        p.go() ;
11    }
12 }
13 }

```

Figure 7: Updated Client Class