TP Report: Design Patterns - Factory and Singleton

Zakariya sabri

October 29, 2025

Exercise 1:

Java Code:

```java
//solution for the first exe_1
public class data_base {
    public String name ;
    public void getConnection(){
        System.out.println("  You   are connected ,to the database
            "+name +".");
    };
    private data_base(String name ){
       this.name=name;
    }
    private static data_base instance =new data_base("_fixe_name"
       );
    public static data_base getInstance(){

       if(instance==null){
           instance=new data_base("_fixe_name");
       }
       return instance;
    }
}
//test in class Main.java
/*
 public static void main(String[] args) {
        data_base obj = data_base.getInstance();
        data_base obj2 = data_base.getInstance();
        //data_base obj4 = new data_base();
        if(obj.equals(obj2)){
            System.out.println("Singleton is work");
        }
        else{
            System.out.println("Singleton is not work");
        }

    }
*/
```

Exercise 2:

Diagram

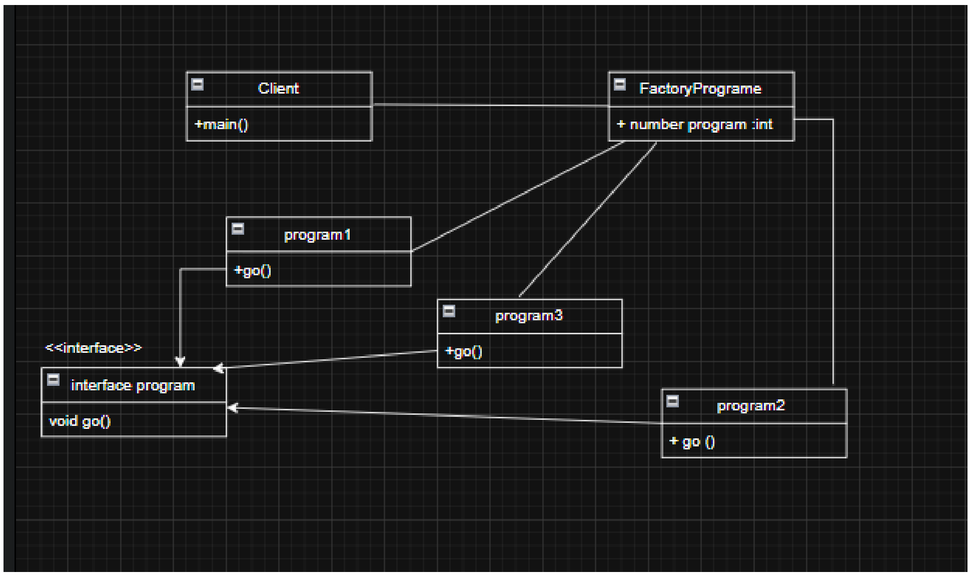

Figure 1: Singleton Pattern UML Diagram

Java Solution Code

Program

```java
public interface Program {
    void go ();
}
```

program 1

```java
public class program1 implements Program
 {
     public program1 () {// The constructor does nothing .
    }
    public void go ()
    {
         System . out . println ("Je suis le traitement 1") ;
    }
 }
```

program 2

```java
public class program2 implements Program
{
    public program2 () {// The constructor does nothing .
    }
    public void go ()
    {
        System . out . println ("Je suis le traitement 2") ;
    }
}
```

program 3

```java
public class program3 implements Program
{
    public program3 () {// The constructor does nothing .
    }
    public void go ()
    {
        System . out . println ("Je suis le traitement 3") ;
    }
}
```

program 4

```java
public class program4 implements Program
{
    public program4 () {// The constructor does nothing .
    }
    public void go ()
    {
        System . out . println ("Je suis le traitement 4") ;
    }
}
```

Program Factory

```java
public class ProgramFactory {
    public static void use_ProgramX(int number_of_programm){
        if(number_of_programm>0 && number_of_programm<=4)
        {
            if (number_of_programm == 1) {
                program1 p = new program1();
                System.out.println("I am main1 ");
                p.go();
            } else if (number_of_programm == 2) {
                program2 p = new program2();
                System.out.println("I am main2 ");
                p.go();
            } else if (number_of_programm == 3) {
                program3 p = new program3();
                System.out.println("I am main3 ");
                p.go();
            } else if (number_of_programm == 4) {
                program4 p = new program4();
```

```java
                    System.out.println("I am main4 ");
                    p.go();
               }
          }
          else {

          System.out.println("The number of programms is invalid ")
              ;
          }
     }


}
```

Main Demo Class

```java
     public static void main(String[] args) {
          int number_of_programm = 4;
          ProgramFactory.use_ProgramX(number_of_programm);

     }
```