المدرسة الوطنية العليا للذكاء الاصطناعي وعلوم المعطيات - تارودانت
ÉCOLE NATIONALE SUPÉRIEURE DE L'INTELLIGENCE ARTIFICIELLE
ET SCIENCES DES DONNÉES - TAROUDANT

Compte-rendu

TP3

Génie Logicielle

Module : Génie logiciel et gestion Projet

Réalisé par

Moutaoikil Mohamed

Exercice :

Modélisons le premier cas , ou approche normale et banale , on créer la classe la class Program et client ,

Aussi prochainement on ajoute l'interface Project pour implémenter la fonction go

Interface Project

```java
package com.project;

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

Class Program1

```java
package com.project;

public class Program1 implements Program {
    public void go() {
        System.out.println("Je suis le programme 1");
    }
}
```

Class Program2

```java
package com.project;

public class Program2 implements Program {
    public void go() {
        System.out.println("Je suis le programme 2");
    }
}
```

Class Program3

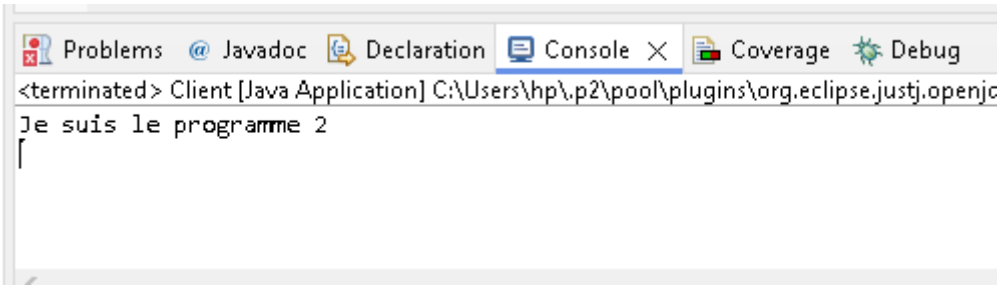
```java
package com.project;

public class Program3 implements Program {
    public void go() {
        System.out.println("Je suis le programme 3");
    }
}
```

On a créé pour chaque Programme class pour l'appeler au class Client

Class client:

1-On peut réaliser la fonctionnalité en modifiant la classe Client pour qu'elle instancie Program1, Program2 ou Program3 selon le paramètre passé en argument.

```java
package com.project;


public class Client {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("Veuillez passer un numéro de programme!");
            return;
        }

        int choix = Integer.parseInt(args[0]);
        Program p;

        if (choix == 1)
            p = new Program1();
        else if (choix == 2)
            p = new Program2();
        else if (choix == 3)
            p = new Program3();
        else
            throw new IllegalArgumentException("Programme inconnu");

        p.go();
    }
}
```



2 & 4

Remarques :

Le code du client caractérisé par couplage forte à toutes les classes Program1, Program2, Program3 et chaque programme on adapte

Alors Le code devient difficile à maintenir et à étendre, donc approche inadéquat

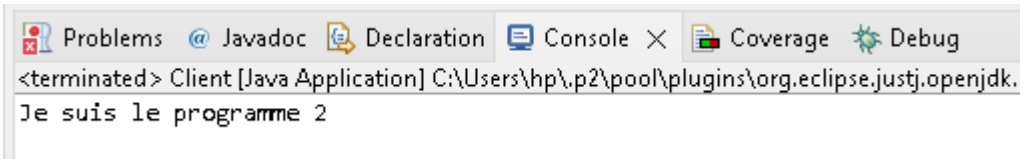
On introduit une classe intermédiaire :

Class ProgramFactory:

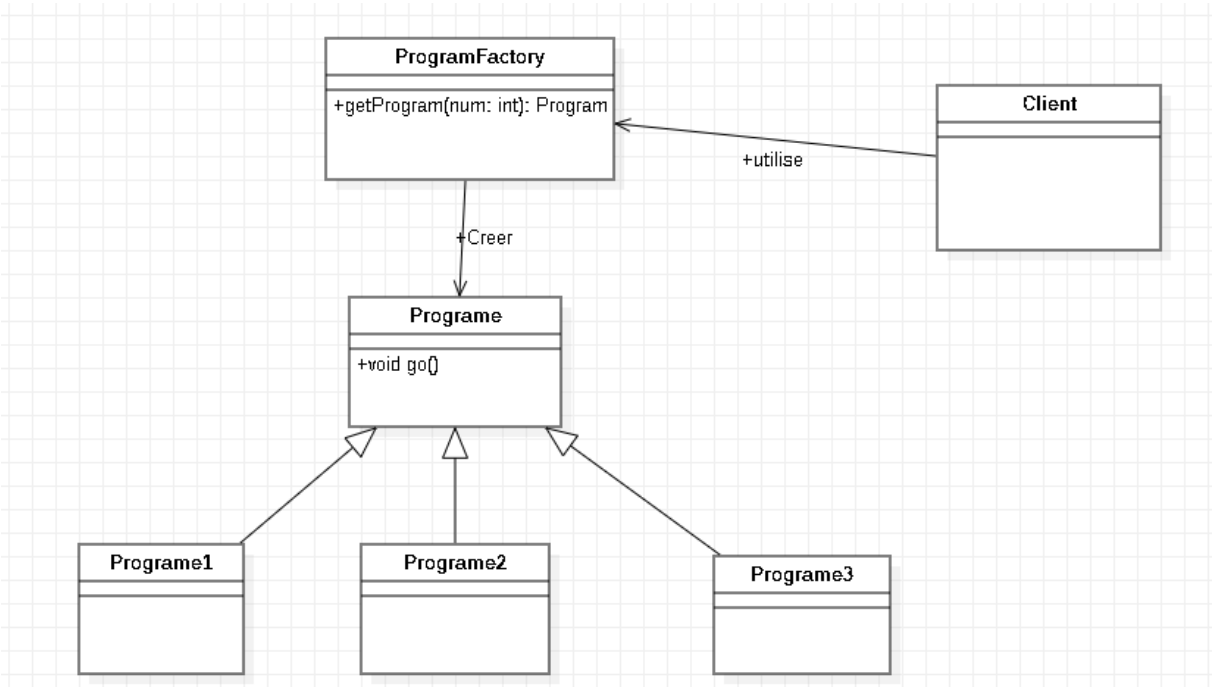
```java
package com.project;

public class ProgramFactory {

    public static Program getProgram(int num) {
        switch (num) {
            case 1: return new Program1();
            case 2: return new Program2();
            case 3: return new Program3();
            case 4: return new Program4();
            default: throw new IllegalArgumentException("Programme inconnu");
        }
    }
}
```

nouveau class Client:

```java
package com.project;


public class Client {
    public static void main(String[] args) {
        Program p = ProgramFactory.getProgram(2);
        p.go();
    }
}

```



3-Diagramme de class :



5-Pour ajouter Programme 4

Il suffit d'ajouter une nouvelle classe :

Class Program4:

```
package com.project;

public class Program4 implements Program {
    public void go() {
        System.out.println("Je suis le programme 4");
    }
}
```