POLYTECH ANGERS

TP 5A - Génie Logiciel
Patrons de conception: Singleton

Nicolas Delanoue

(1 séance de 1h20)

Objectif : Se familiariser avec le patron de conception (Design Pattern) Singleton. Amélioration d'un code existant (notion de *refactorying*).

1 Cahier des charges et prototype

L'objectif est de concevoir un programme permettant de rentrer des chaines de caractères une à une (au moyen d'une ou plusieurs interface graphique) et de les stocker dans un objet que l'on nommera stock. A chaque entrée d'une nouvelle chaîne de caractère, on affiche l'ensemble des chaînes stockées. Le projet `TPSingleton_eclipse_sujet` est un exemple d'implémentation possible (voir la figure 1 pour la modélisation UML).

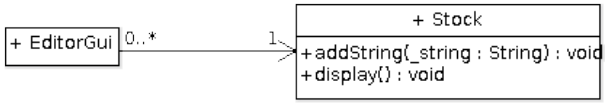

FIGURE 1 – Diagramme de classe UML du projet sujet.

2 Travail à réaliser

On souhaite que ce programme ne requiert pas l'instanciation explicite (par le `main`) de la classe `Stock` puis le passage de sa référence aux instances de `EditorGui`.

Le `Main` fourni est le suivant :

```
1  import singleton.EditorGui;
2  import singleton.Stock;
3
4  public class Main {
5
6      public static void main(String[] args)
7      {
8          Stock monStock = new Stock();
9          EditorGui gui1 = new EditorGui(monStock);
10         EditorGui gui2 = new EditorGui(monStock);
11     }
12   }
```

Le `Main` suivant est celui attendu après l'intégration du Singleton.

```
1  import singleton.EditorGui;
2
3  public class Main {
4
5      public static void main(String[] args)
6      {
7          EditorGui gui1 = new EditorGui();
8          EditorGui gui2 = new EditorGui();
9      }
10 }
```

Ceci permettra de simplifier l'instanciation des éditeurs graphiques, travaillant tous sur une instance unique de la classe `Stock`, sans risque que ces éditeurs travaillent sur des stocks différents. Ceci permettra également de limiter les dépendances (e.g. du `Main` par rapport à la déclaration de `Stock`).

Pour remplir ces exigences, on propose d'implémenter le design pattern *Singleton* (voir figure 2).

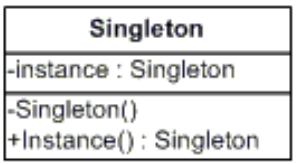

FIGURE 2 – Patron de conception singleton.

Pour résumer, le travail d'intégration du patron de conception singleton à réaliser consiste à :

1. Proposer un diagramme de classe UML adapté à notre contexte,
2. ajuster le code initial en conséquence, afin d'obtenir le même comportement.