

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

TP11: Introduction aux design patterns

L'objectif des exercices ci-dessous est d'utiliser le Design Pattern « Observer ». Ce patron de conception définit une dépendance un à plusieurs entre les objets de sorte que lorsqu'un objet change d'état, tous ses dépendants soient notifiés et mis à jour automatiquement. L'objet surveillé s'appelle le sujet. Les objets qui observent les changements d'état sont appelés observateurs ou auditeurs.

Exercice 1

En java, il existe une classe Observable (dépréciée) et une interface Observer (également dépréciée). L'idée de ce TP est de ne pas se servir de ces éléments existants afin de bien en comprendre le mécanisme.

- Créez les interfaces IObserver et IObservable
- Créez la classe TweeterUser.java est définit par un nom et un pseudo.
- Créez la classe Follower.java. Un Follower est défini par un nom et peut recevoir une notification (i.e. le tweet émis par un TweeterUser ont il connaît le pseudo)
- La méthode main() de la classe Main.java doit permettre le scénario suivant :

Scénario :

seelpyJoe et orangeAnus sont de utilisateurs de Tweeter

fred, igor, dino et pascal sont des followers.

Fred et dino suivent seelpyJoe tandis que fred, igor, dino et pascal suivent orangeAnus

seelpyJoe envoie le tweet "Donald a demandé à un gouvernement étranger d'inventer des mensonges"

orangeAnus envoie le tweet "Joe devrait laisser tomber"

c'en est trop pour dino qui se désabonne de orangeAnus

orangeAnus envoie le tweet "Hier soir, j'étais en tête (...) TRÈS ÉTRANGE "

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

```
* ----- *
* TP java *
* *
* @author Frédéric Rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD11 - ex1 *
* ----- *
```

- seelpyJoe and orangeAnus are some tweeter users
- fred, igor, dino et pascal are some followers.
- fred and dino follow seelpyJoe while fred, igor, dino and pascal follow orangeAnus

Joe Biden has sent message: "Donald asked a foreign government to invent lies"
==> DRING! fred is notified that #sleepyJoesent the tweet: "Donald asked a foreign government to invent lies"
==> DRING! dino is notified that #sleepyJoesent the tweet: "Donald asked a foreign government to invent lies"

Donald Trump has sent message: "Joe should give up"
==> DRING! fred is notified that #orangeAnussent the tweet: "Joe should give up"
==> DRING! igor is notified that #orangeAnussent the tweet: "Joe should give up"
==> DRING! dino is notified that #orangeAnussent the tweet: "Joe should give up"
==> DRING! pascal is notified that #orangeAnussent the tweet: "Joe should give up"

it's too much for dino who unsubscribes from orangeAnus

Donald Trump has sent message: "Yesterday evening, I was in the lead (...) VERY STRANGE"
==> DRING! fred is notified that #orangeAnussent the tweet: "Yesterday evening, I was in the lead (...) VERY STRANGE"
==> DRING! igor is notified that #orangeAnussent the tweet: "Yesterday evening, I was in the lead (...) VERY STRANGE"
==> DRING! pascal is notified that #orangeAnussent the tweet: "Yesterday evening, I was in the lead (...) VERY STRANGE"

Exercice 2

- Créer la méthode `static ArrayList<Follower> createList(String... names)` dans la classe `Main.java` qui crée une `ArrayList <Follower>` avec les noms (`String`) que l'on souhaite (voir trace)
- Ajouter une classe `Model.java` qui sera le model de données et qui étendra `TweeterUser`
- le constructeur de `Model` permet d'instancier une liste de followers
- La méthode `main()` de la classe `Main.java` doit permettre le scénario suivant :

Scénario :

Joe Biden alias `sleepyJoe` est suivi de fred et dino (créé à partir de la méthode `creatList()`)

Donald Trump alias `orangeAnus` est suivi de fred, igor, dino et pascal (créé à partir de la méthode `creatList()`)

Le follower dino est l'un de la liste d'`orangeAnus`. Il a été retrouvé dans la liste d'`orangeAnus`

`seelpyJoe` envoie le tweet "Donald a demandé à un gouvernement étranger d'inventer des mensonges"

`orangeAnus` envoie le tweet "Joe devrait laisser tomber"

c'en est trop pour dino qui se désabonne de `orangeAnus`

`orangeAnus` envoie le tweet "Hier soir, j'étais en tête (...) TRÈS ÉTRANGE "

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

```
* ----- *
* TP java *
* *
* @author Frédéric rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD11 - ex2 *
* ----- *
```

```
- Joe Biden alias sleepyJoe is followed by fred and dino (created from creatList() method)
- Donald Trump alias orangeAnus is followed by fred, igor, dino and pascal (created from creatList() method)
Follower dino is one of orangeAnus's followers pick up in orangeAnus' list
```

```
Name: Joe Biden, Tweet: Donald asked a foreign government to invent lies
==> DRING! fred is notified that #sleepyJoesent the tweet: "Donald asked a foreign government to invent lies"
==> DRING! dino is notified that #sleepyJoesent the tweet: "Donald asked a foreign government to invent lies"
```

```
Name: Donald Trump, Tweet: Joe should give up
==> DRING! igor is notified that #orangeAnussent the tweet: "Joe should give up"
==> DRING! pascal is notified that #orangeAnussent the tweet: "Joe should give up"
==> DRING! fred is notified that #orangeAnussent the tweet: "Joe should give up"
==> DRING! dino is notified that #orangeAnussent the tweet: "Joe should give up"
```

```
it's too much for dino who unsubscribes from orangeAnus
```

```
Name: Donald Trump, Tweet: Yesterday evening, I was in the lead (...) VERY STRANGE
==> DRING! igor is notified that #orangeAnussent the tweet: "Yesterday evening, I was in the lead (...) VERY STRANGE"
==> DRING! pascal is notified that #orangeAnussent the tweet: "Yesterday evening, I was in the lead (...) VERY STRANGE"
==> DRING! fred is notified that #orangeAnussent the tweet: "Yesterday evening, I was in the lead (...) VERY STRANGE"
```