TP7 JAVA

Concepts abordés

- JavaBeans, modèle événementiel

Consignes générales de travail

- Commencez chaque TP dans un nouveau *Projet Java (ici TP4)*
- Créez toujours un (ou plusieurs) paquetages pour contenir ses classes (ne jamais utiliser le paquetage par défaut).
- Écrivez chaque nouvelle classe dans son propre fichier (portant le nom de la classe).
- Pour ce TP, déclarez systématiquement vos attributs en private et vos méthodes en *package friendly* ou public.
- Définissez systématiquement les méthodes equals():boolean, hashcode:int et toString():String *pour chaque classe* que vous écrivez.
- Lisez chaque exercice en entier avant de commencer vos réponses.

Pour mener à bien ces exercices, il est impératif d'ouvrir

1) **la Javadoc** **http://docs.oracle.com/javase/6/docs/api/**
2) le tutoriel de Sun sur le collections
   http://docs.oracle.com/javase/tutorial/collections/index.html

L'3 TP JAVA 2015-2016

L'objectif est de mieux comprendre le modèle d'événements des JavaBeans, notamment la génération d'événement et la notification des écouteurs. Pour mémoire, ces deux notions sont totalement transparentes dans le cadre des composants Swing.
Le TP simule le fonctionnement d'un thermostat selon un point chaud pour lequel il est réceptif ("posé dessus").

Exercice 1 :

TempChangeEvent	Classe pour l'événement "changement de température"
TempChangeListener	Interface pour la méthode tempChanged appelée lors d'un changement de température
PointChaud	Classe pour la source de l'événement "changement de température" Un point chaud doit gérer les thermostats qui lui ont été posés. Un point chaud possède constamment une température courante. Lorsque celle-ci change, il prévient tous les thermostats concernés par ce phénomène.
Thermostat	Classe pour l'écouteur (récepteur, auditeur, listener) d'événement Un thermostat est toujours créé en le "posant sur" un point chaud préexistant. Il doit réagir à la nouvelle température.

Une collection d'auditeurs de type ***Set<T>*** est conseillée avant d'éviter les redondances lors de la notification d'événements.

1) Donner le diagramme de classes UML de l'application
2) Ecrire les 3 classes et l'interface décrites ci-dessus
3) Ecrire une classe de test pour le lancement de l'application de sorte que le code suivant fonctionne :

```
public static void main(String[] argv) {
        System.out.println("POINT CHAUD / THERMOSTAT auditeur");
        PointChaud pointChaud1 = new PointChaud(30.5f);
        Thermostat thermostat1 = new Thermostat(pointChaud1);
        Thermostat thermostat2 = new Thermostat(pointChaud1);
        Thermostat thermostat3 = new Thermostat(pointChaud1);

        System.out.println(pointChaud1);
        System.out.println(thermostat1);
        System.out.println(thermostat2);
        System.out.println(thermostat3);

        pointChaud1.setPointChaud(37);

        System.out.println(pointChaud1);
        System.out.println(thermostat1);
        System.out.println(thermostat2);
        System.out.println(thermostat3);

        thermostat3.retirer();
```

```
        PointChaud pointChaud2 = new PointChaud(8);
        thermostat2.poserSur(pointChaud2);

        System.out.println(pointChaud1);
        System.out.println(pointChaud2);
        System.out.println(thermostat1);
        System.out.println(thermostat2);
        System.out.println(thermostat3);
}
```

TRACE D'EXECUTION

POINT CHAUD / THERMOSTAT auditeur

(Temperature : 30.5, [(Thermostat : 30.5), (Thermostat : 30.5), (Thermostat : 30.5)])
(Thermostat : 30.5)
(Thermostat : 30.5)
(Thermostat : 30.5)
Notification nouvelle temperature = 37.0
Notification nouvelle temperature = 37.0
Notification nouvelle temperature = 37.0
(Temperature : 37.0, [(Thermostat : 37.0), (Thermostat : 37.0), (Thermostat : 37.0)])
(Thermostat : 37.0)
(Thermostat : 37.0)
(Thermostat : 37.0)
(Temperature : 37.0, [(Thermostat : 37.0)])
(Temperature : 8.0, [(Thermostat : 8.0)])
(Thermostat : 37.0)
(Thermostat : 8.0)
(Thermostat : sans point chaud associe)

Exercice 2 :

Ecrire une nouvelle application en faisant de l'attribut ***temperature*** d'un point chaud une **propriété liée.** Le code ci-dessus doit toujours fonctionner de la même manière.