TP : Le thermo-geek-ostat

Vous avez commencé à voir en M2105 (Interfaces Homme/Machine) le patron MVC (Modèle-Vue-Contrôleur). Dans ce TP vous allez approfondir votre compréhension en implémentant ce patron sur une application très simple : un thermostat (de *geek*).

Ce patron se basant sur la notion d'Observateur/Observé, vous réutiliserez les classes du TP précédant.

Il sera aussi l'occasion de réviser vos connaissances en JavaFX

1. Un modèle, des vues

Un thermostat dans sa plus simple expression est un pareil ayant une température désirée et régulant le chauffage pour que la température réelle soit le plus proche possible de la température désirée.
On ne s'occupera pas des capteurs et de la température réelle ici (ni du chauffage) mais seulement de pouvoir régler la température désirée.
On a donc un modèle et un affichage (vue + contrôleur) très simple.

Q1.1 Quel affichage est nécessaire pour seulement régler la température désirée ?
Facile : affichage de la valeur, possibilité d'entrer une nouvelle valeur, possibilité de l'incrémenter/décrémenter (v. l'image ci-dessous)
Tout ca peut se faire avec :

- un Slider
- Un TextField + des boutons « + » et « - »

Q1.2 Que devrait contenir le modèle de l'application ?
Pour rappel, le modèle :

- Est indépendant de la (des) vue(s). Par exemple, pas de « `import javafx` » ;
- contient les données nécessaires à l'application ;
- contient aussi le plus de comportement possible sur ces données (mais sans s'occuper de l'affichage)

Donc quelle donnée doit contenir ce modèle ? Quel comportement ?
Donnée : température désirée
Comportement : set/get simples , idéalement aussi des méthodes d'incrément/décrément de la température

Q1.3 Faites une première version de l'application avec un modèle et un affichage simple.
On doit pouvoir voir la valeur désirée du thermo-geek-ostat, changer cette valeur soit en absolu, soit en l'incrémentant ou décrémentant de 1 degré à la fois.

Optionnel : Aussi permettre l'incrément/décrément de 5 en 5 (avec une interaction adaptée)

S'agissant d'une implémentation MVC, les communications du modèle vers la vue (l'affichage) doivent se faire suivant le patron observateur/observé. Reprenez la classe et l'interface du TP précédant et copiez les dans un package `utils` de ce TP.

Qui est l'observateur ? Qui est l'observé ?

Notez qu'avec une seule vue, ce patron est inutilement compliqué. Mais si l'on a plusieurs vues (par exemple un thermostat à deux extrémités d'une grande pièce) ?

Q1.4 Vous devez pouvoir rajoutez très facilement une nouvelle vue (ou 2, ou 3) en créant simplement des nouveaux objets de la classe vue (une instruction Java). Vérifiez que toutes les vues sont toujours synchronisées (affichent la même température désirée) grâce au MVC (et aux observateurs/observé).

Q1.5 Créez maintenant une nouvelle vue utilisant un Slider pour afficher et modifier la température désirée.
Bien sûr, les anciennes vues doivent continuer à fonctionner.

2. Un modèle plus riche

Pour mériter véritablement son nom de thermostat *geek*, le themo-geek-ostat doit pouvoir accepter des températures dans n'importe quelle échelle de température. La page https://fr.wikipedia.org/wiki/Température vous donnera les informations nécessaires sur toutes les échelles de températures inventées : Celsius, Delisle, Fahrenheit, Kelvin, Leyden, Newton, Rankine, Réaumur ou Rømer.

2.1 Les échelles

Vous allez d'abord créer un type énuméré avec toutes ces possibilités. En plus de définir les constantes identifiant chaque échelle, votre type énuméré peut aussi associer un nom affichable (i.e. une *String*) à la constante et une abréviation (autre *String*). Pour cela définissez des attributs, un constructeur et des *getters* dans votre *enum*. Pour continuer, dans le cadre d'un thermostat (même pour *geeks*), on peut se dire que les températures sont bornées entre -10.0 °C et +30.0°C. Modifiez les enum pour pouvoir donner cette information pour chaque échelle. Cela sera utile pour créer des *Sliders* automatiquement pour n'importe qu'elle échelle de température.

On veut pouvoir convertir les températures entre les différentes échelles. Pour cela on vous propose d'utiliser un échelle « pivot » vers laquelle et à partir de

laquelle toutes les autres pourront être converties. On utilisera par exemple l'échelle Kelvin pour cela (mais n'importe quelle autre choix est possible.

Donc terminez en implémentant les méthodes

- `double toKelvin(double value)`
- `double fromKelvin(double value)`

qui pour une échelle donnée, transforme la valeur en paramètre en Kelvin ou de Kelvin vers l'échelle elle même.

Votre type Echelle est maintenant terminé

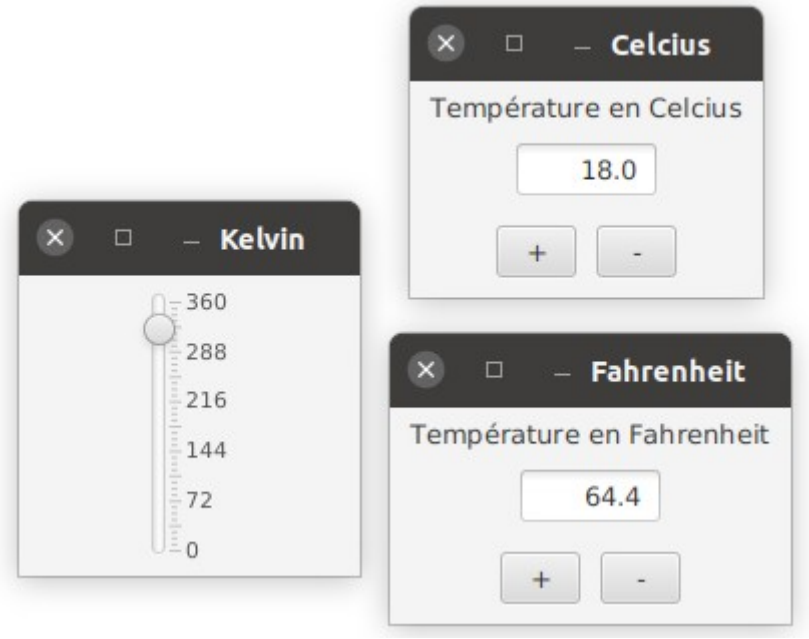
2.2 Les températures

Vous allez maintenant faire de la température une classe connaissant sa propre valeur (la température) et son échelle. Cette classe aura des accesseurs travaillant directement dans l'échelle de l'objet, c'est à dire que un message comme `setValue(0.0)` pour un objet en échelle Celsius (= 273,15 K) n'indiquera pas la même température que le même message pour un objet en échelle Rankine (= 0,0 K).

2.3 Joignons le tout

Reprenez les fenêtres d'affichage faites à la question 1 et modifiez les pour qu'elles aient une échelle associée. Chaque fenêtre travaille sur une température dans une échelle qui lui est propre, mais toutes les fenêtre sont liées entre elles par le modèle. Pour cela, il suffit que les températures (chacune ayant sa propre échelle, v. 2.2) associées aux fenêtres soient toutes liés entre elles (propriétés connectables) : Chaque fenêtre est liée à sa propre propriété Température par un MVC, et toutes les températures sont liées entre elles par des connections bi-directionnelles. Changer la température désirée dans une fenêtre met à jour sa propriété Température par le MVC, toutes les Températures sont liées entre elles par des connections bi-directionnelles, donc elles se mettent toute à jour (chacune dans sa propre échelle). Et toutes les fenêtres se mettent à jour en conséquence du changement de leur propriété Température par le MVC.

Exemple d'affichage final :


Celcius
Température en Celcius
18.0
+
-
Kelvin
360
288
216
144
72
0
Fahrenheit
Température en Fahrenheit
64.4
+
-