Final LO43 A16

durée 2h, 19/01/17

Aucun document autorisé – dictionnaire autorisé

Questions de cours (4 points) :

1. L'ellipse est la possibilité de construire des fonctions avec un nombre quelconque de paramètres. Grâce à quel(s) élément(s) apparu(s) avec Java 1.5 peut-on utiliser l'ellipse ? Donnez un exemple.

2. Qu'est-ce qu'une lambda expression ? A quoi cela sert-il ? Voici un code écrit avec une lambda expression. Transformer le pour qu'il soit compilable par une version de Java avant 1.8. (1.5 par exemple) Comment pourrait-on encore simplifier ce code en Java 1.8?

```
List<Integer> list = Arrays.asList(1, 2, 3, 4, 5, 6, 7);
list.forEach(n -> System.out.println(n));
```

Exercice 2 : Thread (5 points)

Il s'agit ici d'implémenter un système permettant de gérer une application de rédaction partagée de documents texte (façon Google drive). Rassurez-vous, on ne va pas s'occuper de tout le développement de l'application. L'objet de l'exercice est la mise en place du partage de la ressource commune qui est un texte stocké dans un *StringBuffer*. Plusieurs processus peuvent accéder à ce Buffer soit en lecture/écriture (classe *RW*) soit en lecture seule (classe *Read*). Si le buffer est vide, une instance de *RW* a le droit de le modifier et il signale ensuite à tout le monde que le travail est fini, au contraire si une instance de *Read* accède au Buffer, elle fait une recopie locale de son contenu et signale ensuite à tout le monde que la copie est finie. Bien évidemment, l'accès au Buffer est exclusif.

Question 1 : Ecrivez la classe *Texte* qui définit l'élément partagé ainsi que ses méthodes d'accès *...getText(...)* et … *addText(...)*. Est-on obligé d'utiliser un modifieur de champ particulier pour ces méthodes ?

Question 2 : Supposons que l'on désire utiliser la classe *SharedText* (déjà compilée) pour jouer le rôle d'élément partagé. Cette classe possède deux méthodes *public String readText()* et *public void setText(String s)* permettant de lire et d'écrire un texte dans un attribut privé. Ecrivez les classes *RW* et *Read* possédant toutes les méthodes nécessaires pour pouvoir utiliser la classe *SharedText* comme texte partagé.

Exercice 3 : Conception (11 points)

On veut automatiser la gestion de la lumière dans les bâtiments de l'Université afin de réaliser des économies d'énergie. Le cas est volontairement simplifié et réduit au contrôle de lumière des pièces. Une pièce est organisée selon le schéma de la figure 1. La pièce communique avec le couloir

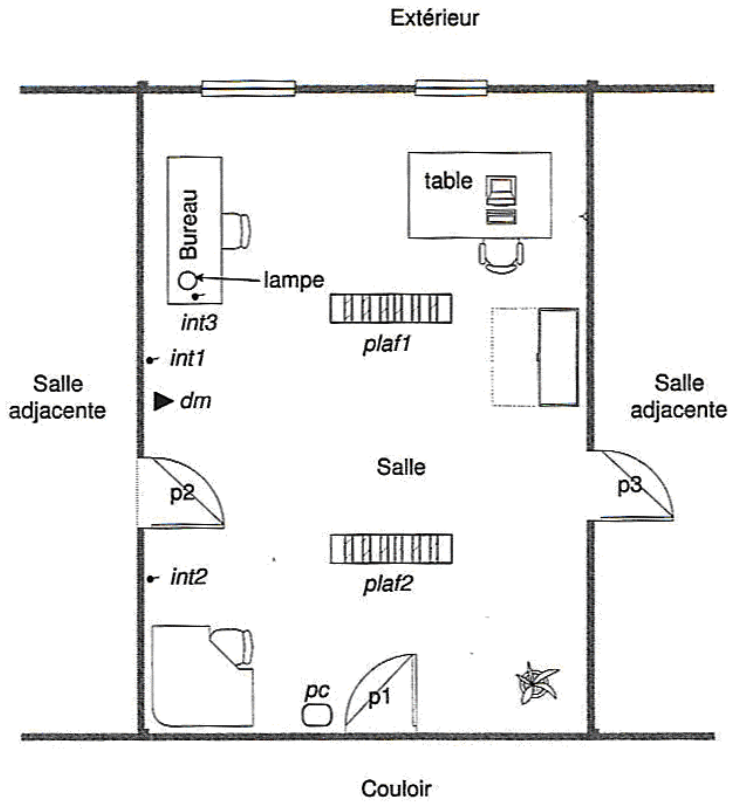
par une porte (***p1***) et avec éventuellement des pièces adjacentes par d'autres portes (***p2*** et ***p3***). Une porte fait partie de la pièce si elle s'ouvre dans la pièce. Chaque porte ***p*** est équipée d'un capteur de contact (***cc<p>***) permettant de savoir si la porte est fermée. Le contrôle de la lumière d'une pièce utilise les équipements suivants :

- 1 détecteur de mouvement composé éventuellement de plusieurs capteurs, (***dm***) qui couvre l'ensemble de la pièce.
- 2 plafonniers (lumières) à intensités variable (***plaf1*** et ***plaf2***) (valeurs comprises entre 0 et 100) commandés par deux variateurs (***var1*** et ***var2***) .
- 2 interrupteurs (***int1*** et ***int2***) pour contrôler les plafonniers (passage de la valeur 0 à 100 ou de 100 à 0)
- 1 lampe de bureau (***lb***) et son interrupteur (***int3***) (la valeur de la lampe de bureau ne peut être que 0 (éteinte) ou 25 (allumée))
- 1 panneau de contrôle pour régler l'intensité des plafonniers ou sélectionner une lumière ambiante (on suppose que le contrôleur sait calculer les intensités des plafonniers en fonction de l'intensité lumineuse extérieure et de l'état de la lampe de bureau)
- 3 capteurs permettant de connaître l'état des sources de lumière (***le1***, ***le2*** et ***le3***)

Les capteurs ont les caractéristiques suivantes :

- Le capteur de contact envoie la valeur 1 si la porte est fermée et 0 sinon.
- Un interrupteur envoie le signal 1 tant qu'on appuie dessus.
- Le détecteur de mouvement envoie le signal 1 si une personne a bougé. Il est remis à 0 de façon régulière.
- Chaque façade du bâtiment dispose d'un capteur de lumière extérieur qui mesure la luminosité (valeur entre 1 et 100).

Les capteurs sont tous actifs c'est à dire qu'ils envoient des signaux de façon régulière.



Les contraintes suivantes sont à respecter :
a) Lorsque la pièce est occupée, l'intensité lumineuse ambiante doit être suffisante pour pouvoir se déplacer (la somme entre la moyenne des capteurs extérieurs et la moyenne des valeurs des lampes de la salle doit être égale à 100), sauf si une intensité de lumière ambiante a été choisie.
b) La configuration de lumière ne varie pas sauf perturbation du système.
c) Si un occupant revient dans une pièce avant T1 minutes, il retrouve la même configuration de lumière que lorsqu'il a quitté la pièce.
d) Si un occupant revient dans une pièce après T1 minutes, la configuration de lumière standard est activée.
e) Lorsqu'une pièce est inoccupée depuis plus de T2 minutes, les lumières sont éteintes.
f) A partir du panneau de contrôle, l'occupant peut choisir un niveau de lumière ambiante, sélectionner la configuration standard, changer la lumière du bureau (allumer/éteindre) ou des plafonniers (allumer/ambiant/éteindre).
g) Les valeurs suivantes sont paramétrées pour chaque pièce par l'administrateur du système : T1, T2, configuration standard.
h) Le dysfonctionnement des capteurs de lumière ou de mouvement est signalé au contrôleur de la pièce.
i) Si le capteur de lumière extérieur ne fonctionne pas, on utilise sa dernière valeur correcte.
j) Si le détecteur de mouvement ne fonctionne pas correctement, on suppose que la pièce est occupée.

Question 1 : Faire un diagramme de cas d'utilisation. (1 point)

Question 2 : Faire un diagramme de classe correspondant au système. (4 points)

Question 3 : Faire le diagramme de séquence correspondant à l'action « choix d'une lumière ambiante » (2 points)

Question 4 : Faire le diagramme de séquence correspondant à l'action « Paramétrage des lumière » (2 points)

Question 5 : Faire un diagramme d'état transition d'un plafonnier. (2 point)

Exercice Bonus (2 points)

Que fait le code suivant ?

```
List<Integer> list = Arrays.asList(1,2,3,4,5,6,7);

int sum = list.stream().map(x -> x*x).reduce((x,y) -> x +
y).get();

System.out.println(sum);
```