

Final AP4B A22

durée 1h30, aucun document autorisé, dictionnaires autorisés

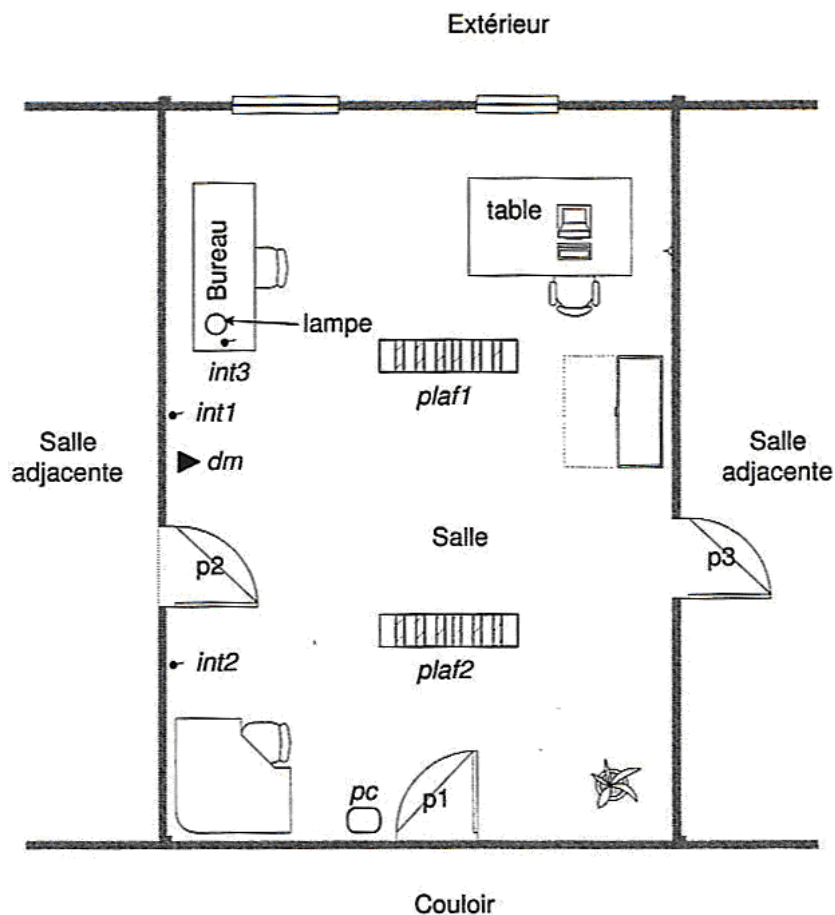
Exercice 1: Qu'affiche le code suivant? (2 points):

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

```
System.out.println(list.stream().map(x -> x*x).reduce((x,y) -> x + y).get());
```

Exercice 2: Economies d'énergie (12 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 suivante.



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 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 3: Ca marche (6 points)

Un robot hexapode est composé d'un corps et de 6 pattes articulées lui permettant de se déplacer à la manière d'un insecte au sol. Le mouvement d'une patte se décompose en deux phases : en **rétraction**, le pied de la patte est posé sur le sol et le mécanisme de la patte effectue un mouvement de l'avant vers l'arrière, participant ainsi à la propulsion du robot. En **protraction**, la patte lève le pied du sol, le ramène en position antérieure et repose le pied au sol.

Une classe Java **Patte** est fournie (ne pas la coder !), munie des méthodes:

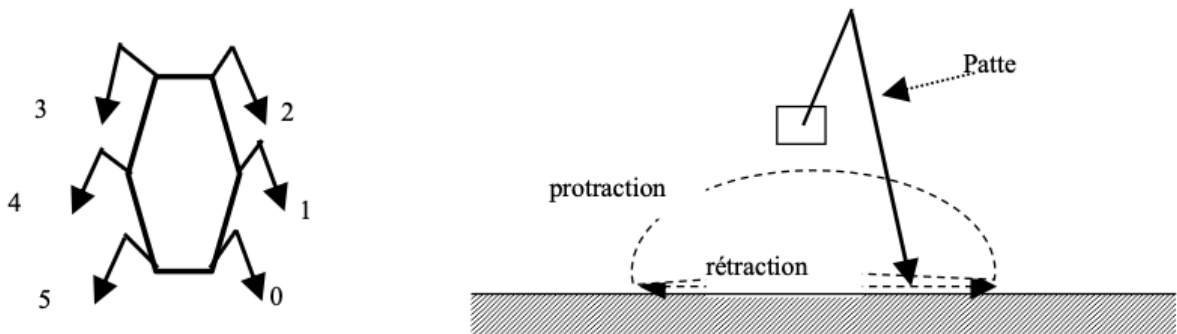
public Patte() - le constructeur

public void retracter() qui effectue la phase de rétraction

public void protracter() qui effectue la phase de protraction

Chaque objet **Patte** est contrôlé par un objet de la classe **ContrôleurPatte** qui en invoque les méthodes **retracter()** et **protracter()**. Afin d'assurer l'équilibre du robot il est nécessaire de synchroniser les phases de **protraction** entre les différentes pattes. La règle est la suivante: une patte a le droit de lever le pied du sol uniquement si ses deux voisines ont le pied posé. Cette synchronisation est assurée par un objet de la classe **Privileges** qui gère les états des 6 pattes sous forme d'un tableau de valeurs booléennes (patte en protraction ou non). Avant de commencer la protraction d'une **Patte**, un **ContrôleurPatte**

doit en obtenir l'autorisation de la part l'objet **Privileges**. Si elle ne peut être accordée, c.-à-d. si une des deux pattes voisines est levée, le **ContrôleurPatte** est mis en attente.



Question 1 : Donnez le code Java des classes **ContrôleurPatte** et **Privileges**. (Note : il est nécessaire que chaque objet **ContrôleurPatte** connaisse son identifiant, c.-à-d. l'indice de la patte qu'il contrôle.)

Question 2 : Donnez le code Java d'une classe **RobotHexapode** dont la seule méthode **main (...)** instancie correctement les objets **Patte**, **Privileges** et **ContrôleurPatte** selon le diagramme d'objets suivant:

