

Examen Commun 1 d'octobre 2022 – Durée 1 h 30

Documents autorisés : page recto A4 manuscrite.

Barème indicatif

Sujet Recto-verso

1 Problème : gestion d'ouverture de portes avec des badges (20 points)

Une entreprise veut réserver l'accès à certaines salles à ceux qui ont une autorisation (niveau de sécurité) suffisante. Pour cela les portes des salles ont un lecteur de badges intégré qui permet d'en contrôler l'accès. On veut mettre en place le système informatique permettant la gestion de ce système.

1.1 Autorisation (8 points)

Les portes ont une autorisation et seules les personnes ayant une autorisation supérieure ou égale peuvent ouvrir la porte.

Plus précisément, une autorisation a un niveau de sécurité (un entier) et elle peut être activée ou désactivée.

Écrivez la déclaration de la classe `autorisation` représentant une autorisation et qui contient :

- le niveau de sécurité `d_niveau` et le fait qu'elle soit activée ou désactivée `d_activee` (vrai si activée, faux sinon) ;
- un `constructeur` par défaut (niveau de sécurité valant 1 et inactivée) et un constructeur prenant en paramètre le niveau de sécurité et le fait qu'elle soit activée ou désactivée ;
- les méthodes `niveau` et `activee` renvoyant respectivement le niveau de sécurité et le fait qu'elle soit activée ou désactivée ;
- la méthode `inferieureA` qui indique si l'autorisation est inférieure à celle passée en paramètre (une autorisation inactivée est toujours inférieure à une autorisation activée, dans les autres cas une autorisation est inférieure à une autre si son niveau de sécurité est strictement inférieur à celui de l'autre) ;
- les méthodes `active()` et `desactive()` qui respectivement active ou désactive l'autorisation ;
- la méthode `afficheSur` qui affiche l'autorisation sur le flot passé en paramètre sous la forme `[A niveau]` si l'autorisation est activée, sous la forme `[D niveau]` si l'autorisation est désactivée (p.ex. `[A 2]` ou `[D 0]`) ;

- la méthode `litDepuis` qui lit l'autorisation depuis le flot passé en paramètre sous la même forme que pour l'affichage.

Écrivez les définitions des constructeurs et des méthodes de cette classe (**exceptées** les méthodes `activee` et `desactive`) : si la lecture des données depuis le flot se passe mal, l'autorisation devra rester inchangée.

Écrivez la procédure de test `test1()` qui crée l'autorisation inactivée de niveau 2, l'affiche à l'écran, l'active et la ré-affiche.

Vous supposerez dans la suite que les surcharges des opérateurs d'écriture et de lecture d'une autorisation sur et depuis un flot ont été déclarées et définies.

1.2 Badges (5,5 points)

Les personnes de l'entreprise ont un badge qui contient l'autorisation de la personne et qu'elles utilisent pour ouvrir les portes.

Écrivez la déclaration de la classe `badge` représentant un badge et qui contient :

- l'identifiant de la personne propriétaire du badge `d_id` (une chaîne de caractères) et l'autorisation `d_autorisation` associée ;
- un constructeur avec l'identifiant et l'autorisation et un constructeur avec l'identifiant et le niveau de sécurité de l'autorisation (l'autorisation sera alors inactivée) ;
- la méthode `id` renvoyant l'identifiant ;
- la méthode `estValide` indiquant si l'autorisation du badge est valide par rapport à l'autorisation passée en paramètre, c'est-à-dire si elle n'est pas inférieure à l'autorisation passée en paramètre ;
- les méthodes `active` et `desactive` qui respectivement active et désactive le badge (c.-à-d. son autorisation) ;
- la méthode `afficheSur` qui affiche le badge sur le flot passé en paramètre sous la forme `[identifiant [A niveau]]` si son autorisation est activée, sous la forme `[identifiant [D niveau]]` si son autorisation est désactivée (p.ex. `[ES1234 [A 3]]` ou `[ES1234 [D 1]]`).

Écrivez les définitions des constructeurs et des méthodes de la classe `badge` (**exceptées** les méthodes `id` et `desactive`).

1.3 Portes (5,5 points)

Les portes ont un lecteur de badge intégré : on doit passer un badge devant la porte pour qu'elle lise le badge et éventuellement s'ouvre. Une porte a une autorisation qui permet d'indiquer si elle sera ouverte ou bloquée en fonction du badge lu. Elle enregistre aussi (dans un tableau) les identifiants des badges qu'elle a lus.

Pus précisément, quand la porte lit un badge, elle enregistre le badge et si le badge est valide pour l'autorisation de la porte alors elle s'ouvre, sinon elle se bloque.

Écrivez la déclaration de la classe `porte` représentant une porte et qui contient :

- son autorisation `d_autorisation` et le tableau des identifiants des badges qu'elle a lus `d_badgesLus` ;
- un constructeur avec son niveau d'autorisation (l'autorisation est activée et aucun badge n'a été lu) ;
- la méthode `litBadge` qui lit le badge passé en paramètre ;
- les méthodes `ouvre` et `bloque` qui respectivement ouvre la porte et la bloque ;
- la méthode `enregistreBadge`, réservée à l'implémentation de la classe, qui ajoute l'identifiant du badge passé en paramètre dans le tableau des badges ;
- la méthode `nombreFoisLu` qui renvoie le nombre de fois où le badge passé en paramètre a été lu par la porte.

Écrivez les définitions du constructeur et des méthodes de la classe `porte` (**exceptées** les méthode `ouvre` et `bloque`).

1.4 Un petit dessin (1 point)

Indiquez grâce à la notation UML les relations entre les classes `autorisation`, `badge` et `porte`.