

Contrôle de Langage Java

Durée : 2h

Aucun document autorisé

Note : la qualité des commentaires, avec notamment la présence d'affirmations significatives, ainsi que les noms donnés aux variables, et la bonne indentation rentreront pour une part importante dans l'appréciation du travail.

L'exercice 1 est obligatoire. Vous ferez ensuite l'exercice 2 ou (exclusif) 3 selon votre choix.

Exercice 1 : Héritage**La classe Matrice**

- 1. Écrivez la classe abstraite générique `Matrice`. On appellera `T` le type des éléments génériques de la matrice. Cette classe contient :
- 2 variables membres protégées, `nbL` et `nbC`, qui sont respectivement le nombre de lignes et de colonnes de la matrice courante ;
 - un constructeur public avec 2 paramètres (le nombre de lignes et de colonnes pour initialiser les 2 variables membres ;
 - le constructeur public par défaut ;
 - les méthodes publiques `getNbLignes` et `getNbColonnes` qui renvoient, respectivement, le nombre de lignes et de colonnes de la matrice courante ;
 - la méthode publique abstraite `get` qui renvoie l'élément de type `T` en (i, j) ;
 - la méthode publique abstraite `set` qui met un l'élément `x` de type `T` en (i, j) ;
 - la méthode publique abstraite `subMat` qui possède 4 paramètres entiers `i1`, `i2`, `j1` et `j2` (avec $i1 \leq i2$ et $j1 \leq j2$), et qui renvoie un objet de type `matrice<T>` formé des lignes `i1` à `i2` et des colonnes `j1` à `j2`.

La classe MatriceDouble

On souhaite manipuler des matrices de réels `Double`. Pour cela, vous allez définir la classe `MatriceDouble` qui hérite de la classe générique `Matrice` précédente, en fixant le type des éléments au type `Double`. Les réels `Double` de la `MatriceDouble` seront placés dans un tableau à deux dimensions.

- 2. Écrivez la classe `MatriceDouble` avec :
- la variable `mat`, le tableau à deux dimensions ;
 - un constructeur pour initialiser une `MatriceDouble` $m \times n$ à une valeur `v` ;
 - le constructeur par défaut ;
 - la programmation des méthodes abstraites `get` et `set` (vérifiez la validité des indices) ;
 - la programmation de la méthode abstraite `submat`. Si les indices ne sont pas valides, vous émettez l'exception `IndexInvalide`. Vous déclarerez la classe `IndexInvalide` héritière de `Exception`. L'exception indiquera le ou les indices invalides.
 - la méthode publique `toString` qui renvoie la représentation sous forme d'une chaîne de caractères de l'objet courant ;
- 3. Écrivez la classe `Test` avec sa méthode `main` qui :

- déclare une `MatriceDouble` `m1` 3×6 dont les éléments sont initialisés à 1.0 ;
- écrit sur la sortie standard `m1`
- déclare une `MatriceDouble` `m2` initialisée à partir de `m1` ;
- crée et écrit sur la sortie standard la sous-matrice prise entre les lignes 1 et 2 et les colonnes 1 et 3 de `m2`. Vous gérerez l'exception.

Exercice 2 : Threads

Dans la classe `MatriceDouble`, on souhaite ajouter la méthode publique `sommeTotale` qui renvoie la somme totale de tous les réels `Double` de la matrice courante. Pour calculer cette somme, on utilise `nbC threads` qui calculent en parallèle la somme des éléments de chacune des colonnes de la matrice. Chaque thread additionnera la somme de sa colonne à une variable partagée qui mémorise la somme totale des éléments de la matrice. Quand tous les threads ont terminé, la fonction `sommeTotale` renvoie la valeur de la variable partagée.

- 4. Écrivez la méthode `sommeTotale`, et les classes nécessaires pour représenter les threads et la variable partagée.

Exercice 3 : Réseau

Écrivez le client UDP, `DayTimeUDPClient.java`, permettant d'interroger le démon `DayTime` qui est accessible sur le port 13 d'une machine dont l'adresse est passée en paramètre au programme. Lorsque le démon `DayTime` reçoit un datagramme (quel qu'il soit), il renvoie au client un datagramme qui contient la date et l'heure courante. Vous afficherez le contenu de ce datagramme de retour sur la sortie standard.

Par exemple, l'exécution du programme qui interroge le serveur `server.unice.fr` donnera le résultat suivant :

```
$ javac DayTimeUDPClient.java && java DayTimeUDPClient server.unice.fr
```

```
Création de la socket locale attachée :
à l'adresse 0.0.0.0/0.0.0.0
au port 53917
```

```
Émission du datagramme
Attente de la réponse...
Réponse reçue de (server.unice.fr:13) :
17 NOV 2021 08:12:57 CET
```