

ITI 1521 - Introduction à l'informatique II - Hiver 2021

Devoir 1

Échéance: **le 15 Février, à 23:30**

En groupes de deux étudiant(e)s.

/25

Instructions

Vous devez faire ce travail **en groupes de deux étudiant(e)s**. Un(e) des deux seulement **doit soumettre le devoir dans le campus virtuel Brightspace**, mais il est important d'indiquer le nom du (de la) deuxième étudiant(e) dans un fichier texte pour qu'il (elle) reçoive lui (elle) aussi la note.

Objectifs

- Maîtriser les concepts liés aux tableaux
- Manipuler des variables références
- Maîtriser les concepts de base de la programmation orientée objet
- Éditer, compiler et exécuter des programmes Java
- Sensibiliser les étudiants face au problème du plagiat, et connaître les règlements de l'université à ce sujet

Exercice 1 (5 points)

Un couple d'entiers est un formé par deux entiers. Exemple : (1, 2) est un couple d'entiers.

- a) Définir une classe nommée `Couple` (classe) avec deux attributs privés `p` et `q` de type `int` (éléments d'un couple).

La classe `Couple` doit être composé :

- de trois constructeurs :
 - ✓ un constructeur pour initialiser les attributs privés du couple.
 - ✓ un deuxième constructeur qui initialisera à 0 les éléments du couple.
 - ✓ un troisième constructeur, qui initialisera un couple à l'aide d'un autre couple.
- des méthodes `getP` et `getQ` permettant d'accéder à chaque élément du couple.
- des méthodes `setP` et `setQ` permettant de modifier chaque élément du couple.
- une méthode `display` qui affichera les variables d'instance.
- une méthode `compare`,
`public boolean compare(Couple);`

qui compare deux couples selon la règle lexicographique suivante :

$(p1, q1) < (p2, q2)$ si et seulement si $(p1 < p2)$ ou $(p1 = p2 \text{ et } q1 < q2)$.

- b) Écrire un programme java nommé `TestCouple` pour tester votre classe `Couple`. Y créer trois objets `c1(2, 3)`; `c2(0,0)` et `c3(2,3)` chacun avec un constructeur. Appeler la méthode `display` pour ces trois objets.
Modifier le premier élément de `c3` en 8 (`c3(8,3)`).
Afficher les éléments de `c3` sans utiliser la méthode `display`.
Comparer `c1` et `c3` par la méthode `compare` et afficher le résultat.

Exemple de sortie :

(2 , 3)

(0 , 0)

(2 , 3)

Après modification, les éléments de `c3` sont : 8 , 3

`c1` est inférieur à `c3` : `true`

Exercice 2 (10 points)

Dans cet exercice, on s'intéresse à créer et trier des tableaux dont les éléments sont de type `Couple` (classe définie dans l'exercice 1).

Écrire un programme java nommé `TriTabCouple` qui contiendra deux méthodes statiques de tri nommées `bubbleSort` et `insertSort` :

```
void bubbleSort(Couple []);  
void insertSort(Couple []);
```

`bubbleSort` consistera à trier un tableau de couples d'entiers par la méthode de tri bulle. Le principe du tri bulle est de parcourir un tableau jusqu'à ce qu'il soit trié. A chaque parcours, on compare deux à deux les éléments consécutifs et on les permute s'ils ne sont pas dans l'ordre.

`insertSort` consistera à trier un tableau de couples d'entiers par la méthode de tri par insertion. Le principe du tri par insertion consiste à considérer une à une les valeurs du tableau, et à les insérer au bon endroit dans le tableau constitué des valeurs précédemment considérées et triées. Le tableau est donc parcouru de droite à gauche, dans l'ordre décroissant. Les éléments sont donc décalés vers la droite tant que l'élément à insérer est plus petit qu'eux.

Dès que l'élément à insérer est plus grand qu'un des éléments du tableau trié, il n'y a plus de décalage et on insère l'élément dans la bonne position.

Utiliser la classe `Couple` de l'exercice 1 ainsi que la méthode `compare` pour comparer deux couples d'entiers.

Exemple de sortie :

Mon tableau trié est : `{(0,5) , (2,3) , (2,5)}`

Exercice 3 (10 points)

Dans cet exercice vous allez implémenter un code correcteur en manipulant des tableaux. Cette technique de codage est destinée à corriger les erreurs de transmission d'une information (message) sur un canal de communication peu fiable. (Référence : https://fr.wikipedia.org/wiki/Code_correcteur).

Un code correcteur envoie des blocs de données (composé des p bits initiaux + les bits ajoutés par le code correcteur pour détecter les erreurs de transmissions).

La qualité d'un code correcteur est déterminée par son rendement (égal à p/n si n est la longueur du bloc transmis) et des 2 nombres suivants:

- le nombre d'erreurs n_1 maximum pour que le code puisse détecter qu'il y a eu des erreurs dans un bloc de transmission. Il est composé des données initiales + les bits ajoutés par le code. Si le nombre d'erreurs dépasse n_1 , on n'est pas certain à 100 % de pouvoir détecter une erreur,
- et le nombre d'erreurs n_2 maximum que le code peut corriger dans un bloc de transmission. Si le nombre d'erreurs dépasse n_2 , on n'est pas assuré que les erreurs seront corrigées par le code.

Le code correcteur qu'on va utiliser dans cet exercice travaille sur des blocs de données initiales de longueur $p = 8$ bits (type primitif `byte` (un octet), comme suit :

- chaque octet est envoyé en triple exemplaire. Celui qui reçoit les données compare les 3 exemplaires. Si au moins 2 exemplaires sont identiques, il suppose que la valeur envoyée est donnée par ces 2 exemplaires. Sinon, il ne peut rien dire.

Question:

Écrire un programme java nommé `Coding` qui implémente ce type de code pour transmettre des données. Les données transmises seront représentées par un tableau d'octets (type primitif `byte`).

La classe `Coding` contiendra une méthode `code` qui sera utilisée par celui qui envoie les données :

```
byte[][] code(byte[] data);
```

`code` prendra en paramètre un tableau d'octets à 1 dimension (`data`) qui représentera les données à transmettre. Elle retournera un tableau à 2 dimensions qui contient les 3 exemplaires du tableau à transmettre. Par exemple, le tableau 1 2 4 11 26 15 sera transformé en tableau à 2 dimensions dont les 3 lignes sont identiques au tableau à transmettre (la 1ère colonne, par exemple, contiendra le 1er octet à envoyer, suivi de ses 2 copies).

```
1 2 4 11 26 15
1 2 4 11 26 15
1 2 4 11 26 15
```

Ce tableau sera envoyé sur la ligne de télécommunication sous la forme 1 1 1 2 2 2 4 4 4 11 11 11 26 26 26 15 15 (le tableau sera envoyé colonne par colonne) et sera reçu sous la forme d'un tableau de `byte` à 2 dimensions. Les données reçues seront rangées dans un tableau à 2 dimensions ayant la même structure que le tableau envoyé. Par exemple, si les données reçues sont 1 1 1 2 3 2 4 4 4 14 11 14 26 26 26 15 15 15 (il y a eu des erreurs durant la transmission pour les nombres en rouge), elles sont rangées dans le tableau :

```
1 2 4 14 26 15
1 3 4 11 26 15
1 2 4 14 26 15
```

La classe `Coding` contiendra une autre méthode nommée `decode` qui prend ce tableau en paramètre et retourne un tableau de dimension 1 qui est supposé être le résultat de la réception de données (après correction des erreurs).

```
byte[] decode(byte[][] data);
```

Pour l'exemple ci-dessus, la méthode `decode` renverra le tableau qui contient 1 2 4 14 26 15 (une erreur n'a pu être corrigée ; en cas de 3 nombres différents on garde un des nombres, ici celui de la ligne 0).

Dans votre programme, les erreurs de transmission seront simulées. Par exemple :
`data[0][0] = 50;`

Exemple de sortie :

Le tableau codé si pas d'erreur de transmission :

```
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
```

Le tableau codé reçu est (contient 3 erreurs en gras):

```
50 2 3 4 5 6
1 2 11 4 5 6
1 2 3 32 5 6
```

Le tableau corrigé est :

```
1 2 3 4 5 6
```

Fraude scolaire :

Cette partie du devoir a pour but de sensibiliser les étudiants face au problème de fraude scolaire (plagiat). Consulter les liens suivants et bien lire les deux documents:

<https://www.uottawa.ca/administration-et-gouvernance/reglement-scolaire-14-autres-informations-importantes>

https://www.uottawa.ca/administration-et-gouvernance/sites/www.uottawa.ca/administration-et-gouvernance/files/processus_de_traitement_des_cas_de_fraude_academique_-_nov_2019.pdf

Les règlements de l'université seront appliqués pour tout cas de plagiat.

En soumettant ce devoir :

1. vous témoignez avoir lu les documents ci-haut ;
2. vous comprenez les conséquences de la fraude scolaire.

Créer et soumettre un seul fichier zip

Directives

- Créez un répertoire que vous nommerez *Devoir1_ID*, où vous remplacerez ID par votre numéro d'étudiant (celui qui soumet le devoir).
Mettez tous les fichiers suivants dans votre répertoire compressé *Devoir1_ID.zip* pour soumission dans le campus virtuel Brightspace.

Fichiers :

- ✓ *README.txt*
 - ✓ *Couple.java*
 - ✓ *TestCouple.java*
 - ✓ *TriTabCouple.java*
 - ✓ *Coding.java*
- N'oubliez pas d'ajouter des commentaires dans chaque programme pour expliquer le but du programme, la fonctionnalité de chaque méthode et le type de ses paramètres ainsi que le résultat.
 - Dans le répertoire *Devoir1_ID*, créez un fichier texte nommé *README.txt*, qui devra contenir **les noms des deux étudiant(e)s**, ainsi qu'une brève description du contenu :

Nom étudiant :

Numéro d'étudiant :

Code du cours : ITI1521

Section Lab: