

TP 3 — Collections en Java

Les *collections* sont des éléments importants de la programmation Java. Très souvent, un tableau ne sera pas l'objet le plus adapté pour capturer un ensemble d'éléments. Dans ce TP nous allons nous initier aux principes de base de l'utilisation des collections. Vous aurez besoin de la documentation Java.

À chaque question le but sera de chercher si l'objet en question peut utiliser des méthodes déjà pré-implementées en Java. Les collections sont une abstraction, permettant de réunir sous un même concept les tableaux, les ensembles, les listes, et tous les objets itérables.

Créez un projet R201_tp3_<nomDeFamille> dans Netbeans et une classe du même nom munie d'une méthode `public static void main(String args[])`.

La class HashSet

Consultez la documentation de la classe [HashSet](#).

Exercice 1. 1) Créer une classe `Ensembles`.

- 2) Instancier un objet `ensemble` qui contient les entiers entre 0 et 100. Quelle est la classe précise de cet objet ?
- 3) Intégrer la méthode suivante dans la classe TP4 :

```
1 public static Integer getRandomInteger(Integer min, Integer max) {  
2     Integer diff = max-min;  
3     Random r = new Random();  
4     return (r.nextInt(diff) + min)  
5 }
```

TP4.class

Que fait cette méthode ? Pourquoi est-elle `static` ?

- 4) Utiliser cette méthode pour instancier un ensemble de 20 éléments contenant des entiers entre 50 et 150.
- 5) Proposer une solution efficace pour tester que l'entier 50 appartient à l'ensemble.
- 6) Ajouter l'entier 42 à l'ensemble. Que se passe-t-il à l'exécution ? Aura-t-on le même résultat à chaque exécution du programme ?
- 7) Proposer une solution efficace pour parcourir et afficher tous les éléments d'un objet de classe `HashSet`.
- 8) Proposer une méthode qui affiche la valeur la plus proche de 100 et sa position dans l'objet `ensemble`. Quelle sera la signature de cette méthode ?
- 9) Proposer une méthode qui permet de copier tous les éléments de l'objet `ensemble` dans un tableau d'entiers (de classe `Integer`) de la même taille.

La classe ArrayList

Consulter la documentation de la classe [ArrayList](#).

Exercice 2. 1) Qu'est-ce qu'une ArrayList ?

- 2) Dans la méthode `main`, instancier un objet de classe `ArrayList` contenant 20 entiers tirés aléatoirement.
- 3) Trouver une façon efficace d'afficher la liste et sa taille.
- 4) Vérifier si l'entier 50 appartient à la liste.
- 5) Ajouter l'entier 42 à la liste.
- 6) Comment parcourir une `ArrayList` ? *Indication — Les `ArrayList` sont indexées.*
- 7) Proposer une méthode qui affiche la valeur la plus proche de 100 et sa position dans l'objet `ensemble`. Quelle sera la signature de cette méthode ?
- 8) Proposer une méthode qui permet de copier tous les éléments de l'objet `ensemble` dans un tableau d'entiers (de classe `Integer`) de la même taille.
- 9) Quelles sont les différences théoriques et pratiques entre les collections `HashSet` et `ArrayList` ?

Exercice 3. On souhaite maintenant utiliser la collection `ArrayList` pour gérer une liste d'étudiant-es. Pour ce faire, nous allons d'abord créer une classe `Etudiant`.

- 1) Créer un fichier `Etudiant.java`. La classe `Etudiant` est munie des attributs et méthodes suivantes :

- `private String nom`
- `private int numero`
- `public String getNom()`
- `public String getNumero()`
- `public String toString()`

Implémenter les méthodes demandées.

- 2) Dans la classe `main`, instancier un objet de classe `ArrayList` dont chaque élément est un objet de classe `Etudiant`, contenant la liste des étudiant-es suivante :

Nom	Numéro
Dupont	0123
Miette	1025
Tran	1010
Hamrouni	1445
Delauney	4791

- 3) Comment retrouver un-e étudiant-à partir de son numéro ?
- 4) À l'inverse, comment retrouver le numéro à partir du nom ?
- 5) Afficher tous les numéros d'étudiant-e alloués.

La classe HashMap

Dans cette partie, on s'intéresse à la classe [HashMap](#), qui permet d'implémenter un système de dictionnaire avec des couples clé/valeur. Nous allons instancier un objet `dicoEtudiants`, de classe `HashMap`, dans lequel la clé sera le numéro d'étudiant-e et la valeur, le nom.

Créez un fichier `EtudiantsHashMap.java` avec la classe correspondante.

- Exercice 4.**
- 1) Quelles peuvent être les classes des objets utilisés comme clés ? et comme valeurs ?
 - 2) Déclarer et instancier l'objet `dicoEtudiants` dans la méthode `main`.
 - 3) Ajouter les 5 noms et numéro de l'exercice précédent à la `HashMap` `dicoEtudiants`.
 - 4) Reprendre les mêmes question qu'à l'exercice précédent, avec l'objet `dicoEtudiants`.

- Exercice 5** (La classe `TreeMap`).
- 1) Lisez [la documentation](#) sur la classe `TreeMap`.
 - 2) Quelles sont les différences entre `HashMap` et `TreeMap` ?
 - 3) Reprendre l'exercice avec une `TreeMap`.