

## 1 Classe CreneauHoraire

**Question 1** *Développer une classe CreneauHoraire caractérisée par un numéro de jour, d'heure et de minute de début, ainsi que par une durée en minutes.*

**Question 2** *Développer le constructeur CreneauHoraire(int, int, int, int).*

**Exécuter :** CreneauHoraire c1 = new CreneauHoraire(87, 10,00,1); CreneauHoraire c2 = new CreneauHoraire(87, 10,00,1);

**Question 3** *Que retourne (c1==c2) ? Pourquoi ? Que retourne (c1.equals(c2)) ? Pourquoi ?*

**Question 4** *Étudier la documentation API de la classe Object. Redéfinir la fonction equals de la classe Object. Vérifier son fonctionnement.*

**Question 5** *Développer pour la classe une fonction statique de saisie d'un créneau horaire sur l'entrée standard (se documenter sur l'utilisation de la classe Scanner de la librairie standard).*

**Question 6** *Développer la fonction de clonage, et tester.*

## 2 Classe CreneauHoraireComparable

Soit une classe CreneauHoraireComparable qui hérite de CreneauHoraire, et qui propose la fonction : *int compareTo(Object o)*. Cette fonction retourne 1, 0, -1 respectivement lorsque la durée du créneau passé en paramètre est supérieure, égale, inférieure.

**Question 7** *Implanter la classe CreneauHoraireComparable et tester la fonction de comparaison.*

Se documenter sur l'interface Comparable (de l'API 1.4.2 dans un premier temps).

**Question 8** *CreneauHoraireComparable pourrait implanter l'interface Comparable ? Quelle en serait l'utilité ?*

**Question 9** *Mettre à jour la classe `CreneauHoraireComparable` pour qu'elle implante l'interface `Comparable`.*

### 3 Classe `ListeTrie`

Soit la classe `ListeTrie` qui hérite de `ArrayList`, et maintient la liste triée en ordre croissant.

La classe `ArrayList` permet de gérer le stockage et l'accès aux différentes références d'objets. `ListeTrie` doit cependant redéfinir certaines fonctions, afin de maintenir la liste en ordre croissant.

**Question 10** *La liste triée accepte uniquement des éléments comparables (de type `Comparable`). Redéfinissez les fonctions d'ajout d'éléments afin de garantir que cette spécificité est respectée (utiliser `instanceof`).*

**Question 11** *Compléter l'implantation afin que la liste soit maintenue triée.*

**Question 12** *Est-il utile de redéfinir la fonction `toString()` de `ArrayList` ? Pourquoi ?*

**Question 13** *Utiliser et tester votre classe avec des objets de type `CreneauHoraireComparable`.*

**Question 14** *On souhaite désormais définir à la compilation le fait que les éléments manipulés par `ListeTrie` sont des comparables en utilisant les types génériques (`ListeTrie` hérite désormais de `ArrayList<Comparable<E> >`, `CreneauHoraireComparable` implante désormais `Comparable<E>`). Quels intérêts a-t-on à faire cela ? (lisibilité ? fiabilité ? vitesse d'exécution ?). Modifier l'implantation de votre classe.*