

Aéroport2

Exercice 1 : Horaires.

On appelle horaire une donnée caractérisant un moment dans une journée, indépendamment de la date de cette journée. Un horaire permet par exemple de préciser le moment du départ d'un train : "le train de 17h42 " où 17h42 représente l'horaire du train évoqué. Nous allons représenter un horaire par une instance de la classe Horaire.

Une instance de Horaire est donc représentée par la donnée de deux nombres : le premier représente l'heure (entre 0 et 23) et le second les minutes (entre 0 et 59).

La classe Horaire implémente l'interface Comparable en réalisant la relation d'ordre intuitive sur ces horaires. Elle propose en plus les méthodes suivantes :

- les accesseurs pour les heures et les minutes ;
- la méthode equals, deux horaires étant égaux s'ils ont mêmes heures et mêmes minutes ;
- une méthode

public int ecart(Horaire h)

dont le résultat est l'écart en minutes entre l'objet horaire invoquant la méthode et l'horaire h passé

en paramètre. Cette méthode déclenche une exception IllegalArgumentException si l'horaire h est plus tôt ("dans la journée") que l'horaire invoquant.

Q 1 . Donnez le code java de la méthode equals.

Q 2 . Le code java de la méthode ecart.

Q 3 . Pour toutes les méthodes de votre classe Horaire autres que les accesseurs et le constructeur donnez le code des méthodes de test que vous proposez.

Exercice 2 : Aéroport et vols.

Dans cet exercice nous utiliserons la classe util.Horaire définie à l'exercice précédent

On s'intéresse à la représentation de vols entre aéroports.

Les vols sont supposés être tous journaliers et il s'agit de "vols courts" donc les horaires de départ et d'arrivée sont toujours dans la même journée. On ne s'occupe donc pas des jours de départ, seul l'horaire compte.

Vols.

Un vol est caractérisé par un numéro unique (une chaîne de caractères), un horaire de départ et un horaire d'arrivée (de type Horaire), les aéroports de départ et de destination. Voici le diagramme de la classe avion.Vol qui permet de représenter ces données :

```
Vol
- numero : String
- depart : Aeroport
- dest : Aeroport
- heureDepart : Horaire
- heureArrivee : Horaire
+ Vol(...)
+ getNumero() : String
+ getDepart() : Aeroport
+ getDestination() : Aeroport
+ getHeureDepart() : Horaire
+ getHeureArrivee() : Horaire
+ equals(o : Object) : boolean
+ hashCode() : int
```

Aéroports.

Un aéroport est caractérisé par un identifiant unique (une chaîne de caractères) et une table de hachage qui associe pour chacun des vols qui partent de cet aéroport, le numéro de vol à l'objet vol correspondant.

Deux objets Aeroport sont donc égaux s'ils ont le même identifiant (on supposera que dans ce cas la liste des vols est nécessairement la même pour les deux objets).

Q 1 . Donnez le code java de l'entête de la classe avion.Aeroport ainsi que la déclaration de ses attributs et de son constructeur sachant qu'initialement la table des vols est vide.

Q 2 . Donnez le code d'une méthode ajouteVol qui prend en paramètre un vol et l'ajoute à cet aéroport.

Cette méthode déclenche une IllegalArgumentException si cet aéroport n'est pas l'aéroport de départ de ce vol.

Q 3 . Donnez le code java d'une méthode volsDirects qui prend en paramètre un aéroport destination

dest. Cette méthode retourne la liste des vols qui partent de cette instance d'aéroport et dont la destination est dest.

Q 4 . Donnez le code java d'une méthode prochainVolDirect qui prend en paramètre un aéroport destination dest et un horaire h et dont le résultat est le numéro du premier vol de cette instance d'aéroport

qui part dans la journée pour dest après l'horaire h fourni. Une exception NoSuchElementException est déclenchée si aucun vol ne convient.

Donnez le code java du corps de la méthode suivante :

```
/** renvoie true ssi le prochain vol direct à destination de dest
 * part dans moins de delai minutes de cet aéroport
 * @param dest la destination
 * @param delai le délai maximal avant le départ du prochain vol pour dest
 * @return true si le prochain vol direct à destination de dest
 * part dans moins de delai minutes de cet aéroport et
 * false si ce n'est pas le cas ou s'il n'y pas de prochain vol direct
 */
public boolean volEnPartance(Aeroport dest, int delai) {
...
}
```

Q 7 . Donnez le code java d'une méthode volsParDestination dont le résultat est la table de hachage

qui pour cet (this) aéroport associe à chaque aéroport a la liste des vols directs partant de this dont a est la destination.

Revision #3

Created 10 September 2020 03:52:48 by ggpilou2

Updated 10 September 2020 08:06:13 by ggpilou2