

# Gérer les rendez-vous de mon Agenda



Nom:

Prenom:

Groupe:

## ! Note

Ce Devoir en temps libre, à faire pendant les vacances, vous permettra de préparer l'évaluation de la rentrée. Il permet notamment de revoir les notions suivantes :

- Héritage / Interface
- Exceptions
- Comparable et/ou Comparator

Lisez attentivement l'énoncé et complétez les codes proposés aux endroits signalés par un numéro.

L'objectif est de créer un programme pour gérer des rendez-vous. Pour cela, vous devrez écrire deux classes :

- une classe **RendezVous** représentant la notion de rendez vous : une date de début et une date de fin ;
- une classe **Agenda** qui permettra de gérer nos rendez-vous.

## La classe Date de l'API Java

Pour gérer les dates, vous utiliserez la classe **Date** de l'API java. La classe **Date** implémente *Comparable<Date>* et peut donc *naturellement* être comparée. Voici un extrait de la documentation de deux méthodes utiles de cette classe :

## Method Detail

### compareTo

```
public int compareTo(Date anotherDate)
```

#### Returns:

the value 0 if the argument Date is equal to this Date; a value less than 0 if this Date is before the Date argument; and a value greater than 0 if this Date is after the Date argument.

### toString

```
public String toString()
```

Converts this Date object to a String of the form:

```
dow mon dd hh:mm:ss zzz yyyy
```

Par ailleurs, on pourra convertir une chaîne en **Date** à l'aide la méthode *parse* de **SimpleDateFormat** :

```
lecteur = new SimpleDateFormat("HH:dd/MM/YYYY");  
// précise le format de saisie de la date  
String chaîne = "10:05/12/2017"; // 10h le 5 décembre 2017  
Date d = lecteur.parse(chaîne);  
// Attention, parse peut lever l'exception ParseException  
// si la chaîne n'a pas le format demandé.
```

Complétez le code suivant :

```
lecteur = new SimpleDateFormat("HH:dd/MM/YYYY");  
String chaîne1 = "10:05/12/2017"; // 10h le 5 décembre 2017  
try  
{  
    Date d1 = lecteur.parse(chaîne1);  
    Date d2 = lecteur.parse("10:06/12/2017");  
1    if(                                     )  
    {  
        System.out.println(d1 + " est antérieur à " + d2 );  
    }  
    else  
    {  
        System.out.println(d1 + " est postérieur à " + d2 );  
    }  
}  
2 catch(                                     )  
{  
    System.out.println("Probleme dans le format d'une des deux dates");  
}
```

## La classe RendezVous

Complétez la classe **RendezVous** suivante. On veut que les RendezVous soient naturellement comparables par date de début croissante.

```

3  class RendezVous
    {
        private Date debut;
        private Date fin;

4      public RendezVous(String deb, String fin) {
            SimpleDateFormat lecteur = new SimpleDateFormat("HH:dd/MM/YYYY");
            this.debut = lecteur.parse(deb);
            this.fin = lecteur.parse(deb);

        }
        @Override
        public String toString(){

5
        }
        @Override
6      public int compareTo(
        {
7          return
        }

        public boolean intersecte(RendezVous v)
        {
            // Renvoie vrai si this et v 'tombent en même temps'
8
        }

    }
}

```

## La classe Agenda

Complétez la classe **Agenda** suivante :

```

class Agenda
{
    private List<RendezVous> contenu;
    Agenda(){
9
    }

    public RendezVous getPremier() {
10
        // Renvoie le premier rendez vous de l'agenda

    }

    public void ajoute (RendezVous v) throws PasDeDisponibiliteException {
        for(RendezVous existant : this.contenu)
            if(v.intersecte(existant))
11
                this.contenu.add(v);
    }

    @Override
    public String toString()
12
    {
    }

}

class PasDeDisponibiliteException
13

```

## Un exécutable

Écrivez un exécutable avec :

- La création de 5 rendez vous, dont deux qui s'intersectent. Tous les rendez vous 'valides' doivent être contenus dans l'agenda à la suite de l'ajout de ces derniers.
- L'affichage du premier rendez vous.

## La classe RendezVousCourt

On souhaite créer une classe **RendezVousCourt** qui correspond à un rendez vous de 5 minutes. Dans cette classe, on ne stocke donc que la date de début du rendez vous. Proposez un diagramme de classes permettant d'avoir dans l'agenda à la fois des rendez vous avec l'implémentation précédente et des rendez vous "courts" (Précisez ce qu'il faudrait ajouter et/ou modifier dans le code précédent).