

Les classes standard des dates en Java

Temps de lecture : 3 minutes

Les classes présentées dans cette leçon peuvent être rencontrées dans du code historique. Il n'est pas conseillé d'utiliser ces classes dans du nouveau code. Il faut préférer l'utilisation de *l'API Date and Time* que nous verrons dans la leçon suivante.

Cependant, il est important de voir le fonctionnement des classes historiques car elles sont utilisés dans certains programmes et il est possible de les rencontrer.

`java.util.Date` et timestamps

Cette classe permet d'encapsuler un `timestamp`, qui représente donc un nombre de millisecondes depuis le *1 janvier 1970 à 00 heure 00 GMT*.

Exemple d'affichage du timestamp de "*l'instant présent*" :

```
var date = new Date(); //import java.util.Date
System.out.println(date.getTime());
```

Pour obtenir ce point dans le temps, `date.getTime` se base par défaut sur l'appel de la méthode statique `System.currentTimeMillis()` ce qui rend sa précision dépendante du système d'exploitation.

Il est également possible de créer un objet `Date` en passant en entrée du constructeur un `timestamp`, ce qui va donc créer une instance de date à un instant donné :

```
var date = new Date(1686313491038L);
```

`java.util.Calendar`

Cette classe encapsule également un `timestamp`, mais elle permet surtout la représentation et la manipulation de celui-ci dans un calendrier et un fuseau horaire.

Pour instancier un calendrier dans "*l'instant présent*" :

```
Calendar calendar = Calendar.getInstance();
```

java.util.GregorianCalendar

`Calendar` étant une classe abstraite, Java propose en standard une implémentation de cette classe : `GregorianCalendar`. À noter que `Calendar.getInstance()` retourne une instance de `GregorianCalendar` par défaut.

Comme son nom l'indique, cette classe permet de représenter un calendrier Grégorien. La méthode `isLeapYear()` permet de savoir si l'année encapsulée par la classe est bissextile.

Manipulation des instances de Calendar

Il est possible d'appeler des méthodes de manipulation des dates encapsulées :

```
Calendar calendar = Calendar.getInstance();  
calendar.set(Calendar.YEAR, 2023);  
calendar.add(Calendar.HOUR_OF_DAY, 2);
```