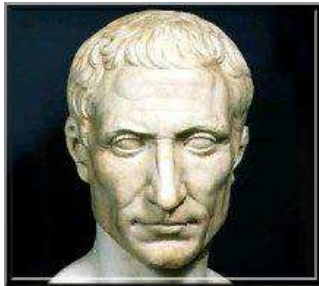


Gérer la date et l'heure en Java (suite)

Les classes Calendar, GregorianCalendar

1 Le concept du calendrier

Voici quelques exemples de calendrier.



Le calendrier Julien
(Jules Cesar)

Le calendrier Grégorien
(Pape Grégoire XIII)



Le calendrier des Français
(La convention)

2 La classe **GregorianCalendar**

La classe `java.util.GregorianCalendar` (sous-classe de `java.util.Calendar`) représente une date et une heure du calendrier grégorien (calendrier occidental). Cette classe dispose de plusieurs constructeurs pour initialiser une instance soit au moment présent, soit à une date et une heure données. Ensuite, on peut manipuler cette date-heure avec divers méthodes.

3 Les méthodes de la classe **GregorianCalendar**

Les méthodes de `java.util.GregorianCalendar` sont similaires à celles de la `Date`.

3.1 Méthodes *get* et *set*

Les méthodes `get` et `set` sont héritées de la super classe `Calendar`. Elles remplacent les méthodes dépréciée (deprecated) `getYear`, `setYear`, `getMonth`, `setMonth` ,..., de la classe `java.util.Date` .

Voici la signature des méthodes `get` et `set` de `GregorianCalendar` :

```
public final int get (int field)
public final void set (int field, int value)
```

avec

<field> qui est une constante de la classe `Calendar` . Voici quelques une des possibilités :
ERA, YEAR, MONTH, DATE, WEEK_OF_YEAR, WEEK_OF_MONTH,
DAY_OF_YEAR, DAY_OF_MONTH, DAY_OF_WEEK,
DAY_OF_WEEK_IN_MONTH, HOUR_OF_DAY, HOUR, AM_PM,
ZONE_OFFSET, DST_OFFSET, MINUTE, SECOND, MILLISECOND.
Chacune de ces constantes est utilisée pour interroger ou modifier l'information correspondante d'une instance de `GregorianCalendar`.

3.2 Méthode *getTime*

Voici le prototype :

```
public final Date getTime()
```

Cette méthode `getTime` renvoie une instance de la classe `Date` :

```
Date d1 = c1.getTime() ;
```

Cette date peut être utilisée avec la classe `java.text.DateFormat` pour convertir une date en texte :

```
System.out.println( DateFormat.getDateInstance().format( d1));
```

3.3 Méthodes pour comparer des dates entre elles

- `public boolean before (Object when)`
- `public boolean after (Object when)`
- `public boolean equals (Object obj)`

4 Programme BonAnniversaire

Cette application demande à l'opérateur sa date d'anniversaire puis elle affiche:

- soit un message vous souhaitant Bon anniversaire si votre anniversaire est aujourd'hui,
- soit un message avec le nombre de jours à attendre pour votre prochain anniversaire en tenant compte des années bissextiles.

Dans cet exemple, on montre comment utiliser les méthodes get et set de la classe `GregorianCalendar` pour calculer des durées.

Voici le code :

```
import java.text.DateFormat;
import java.util.GregorianCalendar;
import java.util.Date;
import javax.swing.JOptionPane;

class BonAnniversaire_reorg_AFAIRE {
    public static void main(String[] args) {
        String date = JOptionPane
            .showInputDialog("Date d'anniversaire (jj/mm) :");

        // Recuperation du jour et du mois
        int indiceSlash = date.indexOf('/');
        String jourDateSaisie = date.substring(0, indiceSlash);
        String moisDateSaisie = date.substring(indiceSlash + 1);

        // Creation d'un format pour mise en forme de la date au format complet (par
        // exemple
        // lundi 12 mai 2003)
        DateFormat formatJour = DateFormat.getDateInstance(DateFormat.FULL);

        // Creation d'un objet GregorianCalendar a la date d'aujourd'hui
        GregorianCalendar aujourd'hui = new GregorianCalendar();
        // aujourd'hui.setTime(new Date());

        // Creation d'un objet GregorianCalendar à la date d'anniversaire entré au
        // au clavier
        // Attention ! l'indice du mois debute a 0 (0 pour janvier, 1 pour
        // fevrier,...)
        GregorianCalendar monProchainAnniversaire = new GregorianCalendar(
            aujourd'hui.get(GregorianCalendar.YEAR), Integer
                .parseInt(moisDateSaisie) - 1, Integer
                .parseInt(jourDateSaisie));

        String message;// La chaîne qui sera fournie à showMessageDialog

        // Calcul de la différence de jours entre les deux dates
        long nbJoursRestant = monProchainAnniversaire
            .get(GregorianCalendar.DAY_OF_YEAR)
            - aujourd'hui.get(GregorianCalendar.DAY_OF_YEAR);

        if (nbJoursRestant == 0)
            message = "Bon anniversaire !";
        else {
            // Si l'anniversaire est déjà passé, la date d'anniversaire passe a
            // l'année prochaine
            if (monProchainAnniversaire.before(aujourd'hui)) {
                // On ajoute un an à la date d'anniversaire
                monProchainAnniversaire.set(GregorianCalendar.YEAR, aujourd'hui
                    .get(GregorianCalendar.YEAR) + 1);
                // GregorianCalendar monAnneeProchaine = monAnniversaire;

                int nbJoursRestantPourAnneeEnCours;

                // rapide calcul du nombre de jours restant
```

```
nbJoursRestantPourAnneeEnCours = monProchainAnniversaire
    .get(GregorianCalendar.DAY_OF_YEAR)
    - aujourd'hui.get(GregorianCalendar.DAY_OF_YEAR);
System.out
    .println("Premier rapide calcul du nonmbre de jour restant:"
        + nbJoursRestantPourAnneeEnCours + "<");

// On construit la date du 31 decembre de cette annee pour
// compter le nb jours restant pour cette annee
GregorianCalendar prochain31Decembre = new GregorianCalendar(
    aujourd'hui.get(GregorianCalendar.YEAR),
    GregorianCalendar.DECEMBER, 31);

nbJoursRestantPourAnneeEnCours = prochain31Decembre
    .get(GregorianCalendar.DAY_OF_YEAR)
    - aujourd'hui.get(GregorianCalendar.DAY_OF_YEAR);

nbJoursRestant = monProchainAnniversaire
    .get(GregorianCalendar.DAY_OF_YEAR)
    + nbJoursRestantPourAnneeEnCours;

// Autre Tests
long aujourd'huiLong = aujourd'hui.getTime().getTime();
long monAnniversaireLong = monProchainAnniversaire.getTime()
    .getTime();

long difference = aujourd'huiLong - monAnniversaireLong;

Date differenceDate = new Date(difference);

GregorianCalendar differenceCalendar = new GregorianCalendar();
differenceCalendar.setTime(differenceDate);

System.out.println("La difference :<
    + formatJour.format(differenceDate.getTime()) + "<");

message = "Votre anniversaire est dans " + nbJoursRestant
    + " jours" + "<nle "
    + formatJour.format(monProchainAnniversaire.getTime());

} else {
    message = "Votre anniversaire est dans " + nbJoursRestant
        + " jours" + "<nle "
        + formatJour.format(monProchainAnniversaire.getTime());

} //if (monProchainAnniversaire.before(aujourd'hui))

} // if( nbJoursRestant == 0 )

JOptionPane.showMessageDialog(null, message);
System.exit(0);
}}
```

5 Hiérarchie d'héritage

