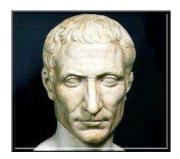
Gérer la date et l'heure en Java (suite)

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 pout 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 aujourdhui = new GregorianCalendar();
      // aujourdhui.setTime(new Date());
      // Creation d'un objet GregorianCalendar à la date d'anniversaire entrée
      // au clavier
      // Attention ! l'indice du mois debute a 0 (0 pour janvier, 1 pour
      // fevrier,...)
      GregorianCalendar monProchainAnniversaire = new GregorianCalendar(
            aujourdhui.get(GregorianCalendar.YEAR), Integer
                  .parseInt(moisDateSaisie) - 1, Integer
                  .parseInt(jourDateSaisie));
      String message;// La chaÃ@ne qui sera fournie à showMessageDialog
      // Calcul de la diff\tilde{\mathtt{A}}©rence de jours entre les deux dates
      long nbJoursRestant = monProchainAnniversaire
            .get(GregorianCalendar.DAY_OF_YEAR)
            - aujourdhui.get(GregorianCalendar.DAY_OF_YEAR);
      if (nbJoursRestant == 0)
         message = "Bon anniversaire !";
      else {
         // Si l'anniversaire est deja passe, la date d'anniversaire passe a
         // l'annee prochaine
         if (monProchainAnniversaire.before(aujourdhui)) {
            // On ajoute un an à la date d'anniversaire
            monProchainAnniversaire.set(GregorianCalendar.YEAR, aujourdhui
                  .get(GregorianCalendar.YEAR) + 1);
            // GregorianCalendar monAnniAnneeProchaine = monAnniversaire;
            int nbJoursRestantPourAnneeEnCours;
            // rapide calcul du nombre de jours restant
```

```
nbJoursRestantPourAnneeEnCours = monProchainAnniversaire
               .get(GregorianCalendar.DAY_OF_YEAR)
               - aujourdhui.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(
               aujourdhui.get(GregorianCalendar.YEAR),
               GregorianCalendar.DECEMBER, 31);
        nbJoursRestantPourAnneeEnCours = prochain31Decembre
               .get(GregorianCalendar.DAY_OF_YEAR)
               aujourdhui.get(GregorianCalendar.DAY_OF_YEAR);
         nbJoursRestant = monProchainAnniversaire
               .get(GregorianCalendar.DAY_OF_YEAR)
               + nbJoursRestantPourAnneeEnCours;
         // Autre Tests
         long aujourdhuiLong = aujourdhui.getTime().getTime();
         long monAnniversaireLong = monProchainAnniversaire.getTime()
               .getTime();
         long difference = aujourdhuiLong - monAnniversaireLong;
         Date differenceDate = new Date(difference);
         GregorianCalendar differenceCalendar = new GregorianCalendar();
         differenceCalendar.setTime(differenceDate);
         System.out.println("La difference :"
               + formatJour.format(differenceDate.getTime()) + "<");</pre>
         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(aujourdhui))
  }// if( nbJoursRestant == 0 )
  JOptionPane.showMessageDialog(null, message);
  System.exit(0);
}}
```

5 Hiérarchie d'héritage

```
Calenda
    + ERA: int = 0
+ YEAR: int = 1
+ MONTH: int = 2
    + WEEK_OF_YEAR: int = 3
+ WEEK_OF_MONTH: int = 4
+ DATE: int = 5
+ DAY_OF_MONTH: int = 5
      + DAY OF YEAR: int = 6
+ DAY_OF_WEEK: int = 7
+ DAY_OF_WEEK_IN_MONTH: int = 8
+ AM_PM: int = 9
     + HOUR : int = 10
+ HOUR OF DAY : int = 11
  + HOUR OF DAY: int = 11

+ MINUTE: int = 12

+ SECOND: int = 13

+ MILLISECOND: int = 14

+ ZONE OFFSET: int = 15

+ DST OFFSET: int = 15

+ FIELD COUNT: int = 17

+ SUNDAY: int = 1

+ MONDAY: int = 3

+ WEDNESDAY: int = 4

+ THURSDAY: int = 5
    + THURSDAY : int = 5
+ FRIDAY : int = 6
+ SATURDAY : int = 7
+ JANUARY : int = 0
     + FEBRUARY : int = 1
    + MARCH : int = 2
+ APRIL : int = 3
  + MAY: int = 4
+ JUNE: int = 5
+ JULY: int = 6
+ AUGUST: int = 7
+ SEPTEMBER: int = 8
      + OCTOBER : int = 9
    + NOVEMBER : int = 10
+ DECEMBER : int = 11
+ UNDECIMBER : int = 12
    + AM : int = 0
 + AM: int = 0

+ PM: int = 1

# fields[]: int

# isSet[]: boolean

# time: long

# isTimeSet: boolean

# areFieldsSet: boolean
 + Calendar()
# Calendar(arg0 : TimeZone, arg1 : Locale)
+ getInstance() : Calendar
+ getInstance(arg0 : TimeZone) : Calendar
+ getInstance(arg0 : Cocale) : Calendar
+ getInstance(arg0 : TimeZone, arg1 : Locale) : Calendar
+ getAvailableLocales() : Locale[]
# computeTime() : void
# computeTime() : void
# getTime() : Date
+ setTime(arg0 : Date) : void
# getTimeInMillis(q) : long
# setTime(arg0 : Date) : void
# getTimeInMillis(q) : long) : void
+ getArg0 : int) : int
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              GregorianCalendar
# get1 ImenNulls(): long
# setTimenNulls(): long)
# setTimenNulls() argo : long): void
+ get(arg0 : int) : int
# internalGet(arg0 : int) : int
+ set(arg0 : int, arg1 : int) : void
+ set(arg0 : int, arg1 : int, arg2 : int) : void
+ set(arg0 : int, arg1 : int, arg2 : int, arg3 : int, arg4 : int) : void
+ set(arg0 : int, arg1 : int, arg2 : int, arg3 : int, arg4 : int) : void
+ set(arg0 : int) : void
+ clear() : void
+ clear() : void
+ set(arg0 : int) : boolean
# complete() : void
+ equals(arg0 : Object) : boolean
+ hashCode() : int
+ before(arg0 : Object) : boolean
+ arter(arg0 : Object) : boolean
+ arter(arg0 : int, arg1 : int) : void
+ roll(arg0 : int, arg1 : int) : void
+ roll(arg0 : int, arg1 : int) : void
+ setTimeZone(arg0 : TimeZone) : void
+ getTimeZone() : TimeZone
+ setLenient(arg0 : boolean) : void
+ setTimeZone() : TimeZone
+ setEirent(arg0 : boolean) : void
+ setEirent(arg0 : boolean) : void
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    (from util)
                                                                                                                                                                                                                                                                                                                                                                                                             + BC : int = 0
+ AD : int = 1
                                                                                                                                                                                                                                                                                                                                                                                                             + GregorianCalendar()
+ GregorianCalendar(arg0: TimeZone)
+ GregorianCalendar(arg0: Locale)
+ GregorianCalendar(arg0: TimeZone, arg1: Locale)
+ GregorianCalendar(arg0: int, arg1: int, arg2: int)
+ GregorianCalendar(arg0: int, arg1: int, arg2: int, arg3: int, arg4: int)
+ GregorianCalendar(arg0: int, arg1: int, arg2: int, arg3: int, arg4: int, arg5: int)
+ SetfCregorianChange(arg0: Date): valid
                                                                                                                                                                                                                                                                                                                                                                                                            + GregorianCalendar(arg0 : int, arg1 : int, 
+ setGregorianChange(arg0 : Date) : void 
+ getGregorianChange() : Date 
+ isLeapYear(arg0 : int) : boolean 
+ equals(arg0 : Object) : boolean 
+ hashCode() : int 
+ add(arg0 : int, arg1 : int) : void 
+ roll(arg0 : int, arg1 : int) : void 
+ roll(arg0 : int, arg1 : int) : void 
- aretMinigum(arg0 : int) : int
                                                                                                                                                                                                                                                                                                                                                                                                               + getMinimum(arg0 : int) : int
                                                                                                                                                                                                                                                                                                                                                                                                           + getMaximum(arg0 : int) : int
+ getGreatestMinimum(arg0 : int) : int
+ getLeastMaximum(arg0 : int) : int
+ getActualMinimum(arg0 : int) : int
      + isLenient() : boolean
            isLenient(): boolean
setFirstDayOftWeek(arg0:int): void
getFirstDayOftWeek(): int
setMinimalDaysInFirstWeek(arg0:int): void
                                                                                                                                                                                                                                                                                                                                                                                                         + getActualMaximum(arg0 : int) : int

# computeFields() : void

# computeTime() : void
         - qetMinimalDaysInFirstWeek(): int
            getMinimum(arg0 : int) : int
getMaximum(arg0 : int) : int
getGreatestMinimum(arg0 : int) : int
            getLeastMaximum(arg0 : int) : int
getActualMinimum(arg0 : int) : int
getActualMaximum(arg0 : int) : int
clone() : Object
        + toString() : String
```