

INFO-H-200 : Programmation orientée objet

TP 2 - Objets et méthodes

Professeur : Hugues Bersini

<http://cs.ulb.ac.be/public/teaching/infoh200>

Année académique 2016-2017

Objets en Java

Exercice 2.1

Ecrire une classe `Date`. Ses attributs privés sont trois entiers : `day`, `month` et `year`. Chaque attribut est accessible via un accesseur en lecture et un accesseur en écriture :

- `public int getDay()`
- `public void setDay(int day)`
- `public int getMonth()`
- `public void setMonth(int month)`
- `public int getYear()`
- `public void setYear(int year)`

Les accesseurs en écriture doivent garantir l'intégrité de l'objet (la date doit rester cohérente).

Cette classe possède les méthodes publiques suivantes :

- `public Date(day, month, year)`, le constructeur.
- `public void increment()`, qui passe la date au jour suivant.
- `public int dayOfYear()` qui renvoie le numéro du jour dans l'année.
- `public int dayOfWeek()` qui renvoie le numéro du jour dans la semaine. Utilisez la congruence de Zeller (voir plus loin).
- `public String toString()` qui transforme la date en chaîne de caractère. Par exemple, "Lundi 2 novembre 2009".

Ajouter les méthodes qui vous semblent utiles comme par exemple pour calculer si l'année est bissextile, le nombre de jours d'un mois, etc.

Tester la classe `Date` dans une classe `TestDate` qui possède une méthode `main()`.

Exercice 2.2

Ajouter à l'exercice précédent une classe `Person` qui possède les attributs `String firstName`, `String lastName` et `Date dateOfBirth`, un constructeur `Person(firstName, lastName, day, month, year)` et les méthodes `String toString()` et `int getAge()`. Pour ce dernier point, voici comment récupérer la date du jour :

```
Calendar now = Calendar.getInstance();
int day = now.get(Calendar.DAY_OF_MONTH);
int month = now.get(Calendar.MONTH) + 1;
int year = now.get(Calendar.YEAR);
```

Tester la classe dans la méthode `main` de la classe `TestDate`.

Annexe : Congruence de Zeller

Pour déterminer le jour de la semaine où commence le mois, vous utiliserez la congruence de Zeller¹. Pour le calendrier grégorien², actuellement utilisé dans la majeure partie du monde, la congruence de Zeller est la suivante :

$$h = \left(q + \left\lfloor \frac{(m+1)26}{10} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor + 5J \right) \bmod 7$$

où

- h est un entier représentant jour de la semaine (0 = samedi, 1 = dimanche, 2 = lundi, ...),
- q est un entier représentant le jour du mois (de 1 à 31),
- m est un entier représentant le numÈro du mois (3 = mars, 4 = avril, ... Janvier et février étant considérés comme les mois 13 et 14 de l'année précédente),
- J est un entier représentant $\lfloor year/100 \rfloor$ (par exemple 20 pour l'année 2008),
- K est un entier représentant l'année dans le siècle, c'est à dire $year \bmod 100$ (par exemple 8 pour l'année 2008) et
- $\lfloor x/y \rfloor$ représente le résultat de division entière de x par y .

En C++, cela donne le code suivant :

```
int dayOfWeek(int dayInMonth, int month, int year)
{
    if(month==1 or month==2)
    {
        month += 12;
        year--;
    }
    int century = year/100;
    int yearOfCentury = year%100;
    int dayOfWeek = (dayInMonth +
                     + ((month+1)*26)/10)
                     + yearOfCentury
                     + (yearOfCentury/4)
                     + (century/4)
                     + 5*century) % 7;
    return (dayOfWeek+5)%7;
}
```

Annexe : Années bissextiles

Le mois de février des années bissextiles contient 29 jours au lieu de 28. Depuis l'instauration du calendrier grégorien, sont bissextiles les années divisibles par 4 mais non divisibles par 100 ou les années divisibles par 400. Ainsi, 2008 est une année bissextile.

1. http://en.wikipedia.org/wiki/Zeller's_congruence
2. http://en.wikipedia.org/wiki/Gregorian_calendar

Exercice 2.1 et 2.2

Fichier 'Main.java'

```
package Example;

public class Main {
    public static void main(String args[]){
        Date today = new Date(5, 11, 2009);
        System.out.println(today);

        Person prof = new Person("Hugues", "Bersini", new Date(6, 11, 1965));
        System.out.println(prof);
    }
}
```

Fichier 'Date.java'

```
package Example;

public class Date {
    private static final Integer daysInMonths[]={31,28,31,30,31,30,31,31,30,31,30,31};
    private static final String monthInFrench[]={
        "Janvier", "Février", "Mars",
        "Avril", "Mai", "Juin", "Juillet", "Aout",
        "Septembre", "Octobre", "Novembre", "Decembre"
    };
    private static final String dayInFrench[]={
        "Lundi", "Mardi", "Mercredi",
        "Jeudi", "Vendredi", "Samedi",
        "Dimanche"
    };

    Integer day;
    Integer month;
    Integer year;

    public Date(Integer day, Integer month, Integer year){
        this.year = year;
        if(month<1 || month >12)
            this.month=1;
        else
            this.month=month;
        if(day<1 || day>daysInMonth())
            this.day=1;
        else
            this.day=day;
    }

    /**
     * increments the date of one day
     */
    public void increment() {
        if(lastDayOfMonth()) {
            day = 1;
            if(month == 12){
                month = 1;
                year++;
            } else {
                month++;
            }
        } else {
            day++;
        }
    }

    /**
     * @returns The number of days in the month this date belongs to.
     */
    private Integer daysInMonth() {
        return daysInMonths[month-1];
    }

    /**
     * @returns true if this is the last day of the month.
     */
    private Boolean lastDayOfMonth() {
        return day == daysInMonth();
    }

    /**
     * @returns the day number in the current year this date represents.
     */
    public Integer dayOfYear() {
        Integer dayOfYear=this.day;
        for(Integer i=0; i<month-1;i++)
            dayOfYear += daysInMonths[i];
        return dayOfYear;
    }

    /**
     * @returns 0 for monday 1 for tuesday and so on
     * see http://en.wikipedia.org/wiki/Zeller's\_congruence
     */
    public Integer dayOfWeek() {
        Integer m = this.month; // local copies because
```

```

    Integer y = this.year; // month and year can be modified
    if (m==1 || m==2)
    {
        m += 12;
        y--;
    }

    Integer century = y/100;
    Integer yearOfCentury = y%100;
    Integer dayOfWeek = (day +
        + ((m+1)*26)/10)
        + yearOfCentury
        + (yearOfCentury/4)
        + (century/4)
        + 5*century) % 7;

    return (dayOfWeek+5)%7;
}

public String toString() {
    return dayInFrench[dayOfWeek()]+" "+day+" "+
        monthInFrench[month-1]+" "+year+
        " le "+dayOfYear()+"-ième jour de l'année";
}
}

```

Fichier 'Person.java'

```

package Example;

import java.util.Calendar;

public class Person {
    String firstName;
    String lastName;
    Date dateOfBirth;

    public Person(String firstName, String lastName, Date dateOfBirth){
        this.firstName = firstName;
        this.lastName = lastName;
        this.dateOfBirth = dateOfBirth;
    }

    public String toString() {
        return firstName+" "+lastName+" né le "+dateOfBirth
            + ", il a "+age()+" ans.";
    }

    public Integer age() {
        Calendar now = Calendar.getInstance(); //Calendar from java.util
        Integer year = now.get(Calendar.YEAR);
        Integer month = now.get(Calendar.MONTH)+1; //in Calendar January is 0, February is 1, and so on
        Integer day = now.get(Calendar.DAY_OF_MONTH);
        System.out.println(year+"/"+month+"/"+day);
        Integer age = year - dateOfBirth.year;
        if(month < dateOfBirth.month ||
            (month == dateOfBirth.month && day < dateOfBirth.day))
            age--;
        return age;
    }
}

```