

On souhaite écrire un composant permettant de calculer le jour de la semaine à partir d'une date donnée (par exemple, on souhaite obtenir « mardi » pour le « 6 octobre 2009 »).

## 1 Spécification

### 1.1 L'interface

Le composant demandé devra implémenter l'interface `ConvertisseurDateEnJour` qui ne contient qu'une seule méthode :

```
/**
 * Converti une date du calendrier grégorien en le jour de la semaine
 * correspondant.
 * @param jour le jour dans le mois (entre 1 et 31).
 * @param mois le mois (entre 1 et 12).
 * @param annee l'année.
 * @throws InvalidGregorianCalendarException
 *         si la date est avant le 15 octobre 1582 ou si les paramètres
 *         ne correspondent pas à une date.
 * @return le jour de la semaine correspondant entre 1 et 7.
 *         1 correspondant à dimanche.
 */
public int getDay(int jour, int mois, int annee)
    throws InvalidGregorianCalendarException;
```

### 1.2 Le composant

Afin de déterminer le jour de la semaine à partir de la date, le composant utilisera la formule de Zeller (cf. section 1.3). Pour cette raison, le composant se nommera `Zeller`.

Le composant sera conçu selon le « design pattern » singleton, aussi il contiendra la méthode `getInstance` qui retournera l'instance unique correspondant :

```
public static Zeller getInstance();
```

Le composant et l'interface feront partie du même « package » : `calendrier`.

### 1.3 La formule de Zeller

La formule de Zeller (établie par l'allemand Zeller en 1885) permet de trouver le jour de la semaine d'une date donnée dans le calendrier grégorien.

Pour obtenir le jour de la semaine, on calcule tout  $k$  selon la formule suivante :

$$k = q + [2, 6(m + 1)] + \left\lfloor \frac{s}{4} \right\rfloor - 2s + d + \left\lfloor \frac{d}{4} \right\rfloor,$$

où

- q est le quantième,
- m est le rang du mois dans l'année, sauf pour janvier et février qui sont considérés respectivement comme les 13<sup>e</sup> et 14<sup>e</sup> mois de l'année précédente
- s est la partie séculaire de l'année
- d est constitué des deux derniers chiffres de l'année

Si on associe 0 à samedi, 1 à dimanche et ainsi de suite jusqu'à vendredi ; le jour de la semaine est alors le reste de la division euclidienne par 7 de  $k$  (en considérant que le reste doit être positif).

## 2 Le travail à effectuer

### 2.1 Les tests

Le premier travail à réaliser est de définir un ensemble de tests à l'aide de la spécification.

Vous devez réaliser deux jeux de tests :

- le premier jeu de tests ne devra prendre en compte que la spécification de l'interface ;
- le second enrichira le premier jeu de tests en tenant compte de la spécification du composant.

Vous devez fournir les feuilles de scénarios de tests (disponibles sur la liste de diffusion) qui pourront permettre à une autre équipe d'implémenter ces tests.

### 2.2 L'implémentation

Implémentez tout d'abord une classe `TestCalendrier` permettant de réaliser les tests à l'aide de Junit. Les tests devront être placés dans un répertoire distinct de celui contenant le composant à tester, le « package » sera par contre le même que celui du composant.

Vous pourrez ensuite implémenter le composant et le tester à l'aide de votre jeu de test.

À la fin du TP vous devez envoyer les documents que vous avez rempli ainsi que les sources et les pseudo-codes réalisés à votre chargé de TP.