

DEV1 – Laboratoires Java I

TD 7 – Mise en pratique Le calendrier

Dans ce TD vous utiliserez les notions vues précédemment afin de réaliser une petite application permettant d'afficher sur le terminal le calendrier d'un mois et d'une année donnés.

Votre programme demandera à l'utilisateur d'entrer au clavier un mois et une année au format MM (ou M) AAAA (par exemple 6 2018 ou 12 2015) et afficher le calendrier de ce mois. Par exemple pour le mois de juin 2018 le programme affichera :

```
=====
                Juin 2018
=====
Lu  Ma  Me  Je  Ve  Sa  Di
                01  02  03
04  05  06  07  08  09  10
11  12  13  14  15  16  17
18  19  20  21  22  23  24
25  26  27  28  29  30  31
=====
```

Nous allons vous guider tout au long de ce TD afin de développer les différentes méthodes nécessaires à cette petite application.

Créer un package `g12345.dev1.calendrier` et une classe `Calendrier`.

Exercice 1

Mois

Écrivez la méthode `String nomMois(int mois)` qui reçoit le numéro d'un mois et retourne son nom en français.

Par exemple si la méthode reçoit 1 elle retourne "Janvier" et si elle reçoit 6 elle retourne "Juin".

Nous supposons pour le moment que la méthode reçoit toujours un nombre entre 1 et 12. Nous traiterons les erreurs au TD suivant.

Exercice 2

Titre

Écrivez la méthode `void afficherTitre(int mois, int année)` qui reçoit un mois et une année et affiche le titre.

Par exemple si la méthode reçoit 6 et 2018 elle affiche :

```
=====
                Juin 2018
=====
```

Cette méthode fera un appel à la méthode `nomMois` de l'exercice précédent.

Exercice 3

Méthode principale

Écrivez la méthode principale de votre programme qui demande à l'utilisateur le mois et l'année et affiche le titre du mois correspondant.

Par exemple si l'utilisateur entre 6 et 2018 votre programme affiche :

```
=====
                Juin 2018
=====
```

Exercice 4

Entête

Écrivez la méthode `void afficherEntête()` qui affiche l'entête :

```
Lu  Ma  Me  Je  Ve  Sa  Di
```

Complétez votre méthode principale afin qu'elle affiche l'entête juste après le titre.

Exercice 5

Afficher le mois

Écrivez la méthode `void afficherMois(int décalage, int nombreJours)` Cette méthode reçoit le décalage de départ et le nombre de jours dans le mois et affiche les numéros de jours du mois en prenant soin de faire les sauts de ligne opportuns.

Par exemple si elle reçoit 4 et 31 cette méthode affiche :

```

                                01  02  03
04  05  06  07  08  09  10
11  12  13  14  15  16  17
18  19  20  21  22  23  24
25  26  27  28  29  30  31
```

Et si elle reçoit 1 et 28 elle affiche :

```

                01  02  03  04  05  06
07  08  09  10  11  11  13
14  15  16  17  18  19  20
21  22  23  24  25  26  27
28
```

Exercice 6

Années bissextiles

Pour connaître le nombre de jour dans un mois il est nécessaire de savoir si l'année est bissextile ou non¹. En effet, le mois de février de ces années contient 29 jours au lieu de 28.

La règle est simple, une année est bissextile si l'une des deux conditions suivantes est vraie

- ▷ si elle est divisible par 4 et non divisible par 100, ou
- ▷ si elle est divisible par 400.

Par exemple, les années 2000 (car divisible par 400) et 2008 (car divisible par 4 mais pas par 100) sont bissextiles. Par contre, les années 1900, 2018 ou encore 2003 ne le sont pas.

1. http://en.wikipedia.org/wiki/Leap_year

Écrivez la méthode `boolean estBissextile(int année)` qui reçoit une année en paramètre et retourne vrai si cette année est une année bissextile et faux sinon.

Exercice 7

Nombre de jours dans un mois

Écrivez la méthode `int nombreJours(int mois, int année)` qui reçoit un mois et une année et retourne le nombre de jours dans ce mois.

Par exemple si le méthode reçoit 1 et 2018 elle retourne 31 car le mois de janvier a toujours 31 jours. Si elle reçoit 2 et 2018 elle retourne 28 car février 2018 a 28 jours puisque l'année 2018 n'est pas bissextile.

Mettez à jour votre application afin d'afficher le nombre correct de jours dans votre calendrier.

Exercice 8

Le premier jour du mois

Il nous reste à connaître le décalage à effectuer en début de mois. Pour cela il faut connaître le premier jour du mois : lundi, mardi,... Si le premier jour du mois est un lundi il n'y aura pas de décalage, si c'est un mardi il y aura un décalage de 1, etc.

Il existe une formule permettant de calculer le jour correspondant à une date (jour, mois, année) donnée. Cette formule s'appelle la congruence de Zeller².

$$h = \left(q + \left\lfloor \frac{(m+1)13}{5} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor + 5J + 5 \right) \bmod 7$$

- ▷ h est un entier représentant le jour de la semaine (0 = lundi, 1 = mardi, 2 = mercredi, ...),
- ▷ q est un entier représentant le jour du mois (de 1 à 31),
- ▷ m est un entier représentant le numéro du mois (3 = mars, 4 = avril, ... janvier et février étant considérés comme les mois 13 et 14 de l'année précédente. Donc janvier 2008 sera considéré comme 13 2007),
- ▷ 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 .

Ainsi, pour le 3 mars 2008, $q = 3$, $m = 3$, $J = 20$ et $K = 8$. Le résultat h de la congruence de Zeller est $((3 + 10 + 8 + 2 + 5 + 100 + 5) \bmod 7) = (133 \bmod 7) = 0$ et donc lundi.

Écrivez la méthode `int numéroJour(int jour, int mois, int année)` qui retourne la valeur de la congruence de Zeller correspondant au jour reçu en paramètre.

Terminez votre application en y insérant le décalage obtenu avec cette méthode `numéroJour`

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