

## Séance 7: AGENDA

Université Paris Cité

### Objectifs:

- |                                     |                        |
|-------------------------------------|------------------------|
| — Tableaux unidimensionnels         | — Tableaux d'entiers   |
| — Tableaux de chaînes de caractères | — Parcours de tableaux |

Dans ce TP nous allons écrire des fonctions pour manipuler un agenda personnel.

Placez les fichiers dans le bon répertoire. Pour ce TP, vous devez compléter le fichier `agenda.java` et avoir dans votre répertoire le fichier `agenda.dat` qui contient l'agenda personnel.

L'agenda contient au plus un évènement par jour. Nous allons le représenter par un tableau `t` de chaînes de caractères. La chaîne de caractère `t[i]` est la description de l'évènement qui se produit le  $i + 1$ -ème jour de l'année. Par exemple, `t[0]` est la chaîne de caractères "Jour de l'An". S'il n'y a aucun évènement particulier au  $i + 1$ -ème jour, `t[i]` est la chaîne de caractères vide "". L'agenda ne concerne que l'année 2016, qui est une année bisextile, donc `t` est un tableau de taille 366.

### Exercice 1 (Afficher l'agenda jour après jour, ☆)

1. Dans la fonction principale `main`, chargez l'agenda sauvegardé dans le fichier `agenda.dat` dans un tableau `t` à l'aide de la fonction `loadAgenda` :

```
1 String [] t = loadAgenda("agenda.dat");
```

puis affichez tout l'agenda, jour par jour, en rappelant le numéro du jour en début de ligne.

#### Contrat:

Le programme affiche

```
0 Jour de l'An
1
2
...
86 Pâques
...
```

2. Modifiez votre programme pour qu'il n'affiche que les jours où il y a un évènement.

#### Contrat:

Le programme affiche

```
0 Jour de l'An
86 Pâques
87 Lundi de Pâques
```

...

3. Structurez votre code : créez une procédure `showAgenda` qui prend en argument un tableau contenant un agenda et qui l'affiche. Modifiez la fonction principale `main` pour appeler la procédure `showAgenda`.
4. Déclarez un tableau

```
1 String [] wDays = {"Lun", "Mar", "Mer", "Jeu", "Ven", "Sam", "Dim"};  
2
```

puis utilisez-le pour que votre programme affiche le jour de la semaine en début de ligne.

**Contrat:**

Le programme affiche

```
Ven 0 Jour de l'An  
Dim 86 Pâques  
Lun 87 Lundi de Pâques  
...
```

5. Affichez le nombre d'évènements rentrés dans l'agenda avant d'afficher l'agenda.

**Contrat:**

Le programme affiche

```
Il y a 12 évènements dans l'agenda.  
Ven 0 Jour de l'An  
Dim 86 Pâques  
Lun 87 Lundi de Pâques  
...
```

6. Modifiez la procédure `showAgenda` pour qu'elle prenne en arguments un tableau de chaînes de caractères `t` contenant l'agenda, un entier `startDay` contenant le jour de départ, et un booléen `reverse`. Si `reverse` vaut `true`, la procédure affiche les évènements planifiés avant le jour `startDay`, dans l'ordre chronologique inverse. Sinon, elle affiche les évènements planifiés après le jour `startDay` dans l'ordre chronologique.<sup>1</sup>

**Contrat:**

Un appel à `affiche(t,0,false)` affiche l'agenda comme à la question précédente. Un appel à `affiche(t,87,true)` affiche le lundi de Pâques, puis Pâques, puis le jour de l'An.

□

## Exercice 2 (Affichage de la date, \*\*)

On veut maintenant remplacer le numéro du jour dans l'année par une date lisible. On a déjà vu la conversion d'un numéro de jour en une date dans un TP précédent, mais on va suivre une autre approche. La première étape est d'écrire une fonction `monthOfDayNumber` qui prend en argument le numéro d'un jour et renvoie le mois dans lequel tombe ce jour.

**Contrat:**

Par exemple `monthOfDayNumber(0)` renvoie 1 car le jour 0 est le premier janvier, et janvier est le mois 1. De même, `monthOfDayNumber(86)` renvoie 3 car le jour 86 est le 27 mars.

Un enseignant (un peu fatigué par les corrections de copies) a écrit la fonction suivante.

```
1 public static int monthOfDayNumber (int dayNumber) {
```

1. [BONUS] Évitez autant que possible d'avoir du code redondant. Vous pouvez utiliser par exemple une fonction pour l'affichage d'un évènement. Vous pouvez aussi écrire une seule boucle dont le déroulement dépend de la variable `reverse`.

```

2  int [] daysInMonth = {31,29,31,30,31,30,31,31,30,31,30,31};
3  int res = 1;
4  for (int i = 0; i<12; i++){
5      if (dayNumber > daysInMonth[i]) {
6          dayNumber = dayNumber - daysInMonth[i];
7          res = res + 1;
8      }
9  }
10 return res;
11 }

```

1. Utilisez cette fonction pour afficher le mois de chaque évènement.

**Contrat:**

*Le programme affiche*

*Il y a 12 évènements dans l'agenda.  
 Ven 0 1 Jour de l'An  
 Dim 86 3 Pâques  
 Lun 87 3 Lundi de Pâques  
 Dim 121 4 Fête du travail ...*

2. Si vous regardez bien, il y a un problème avec la fête du travail. La fonction monthOfDayNumber a des erreurs ! Corrigez-la.
3. En vous inspirant beaucoup de la fonction monthOfDayNumber corrigée, créez une fonction dateOfDayNumber qui renvoie un tableau de deux entiers, le premier entier étant le numéro du jour dans le mois, et le second le numéro du mois dans l'année.

**Contrat:**

*Par exemple dateOfDayNumber(0) renvoie le tableau {1,1}, et dateOfDayNumber(86) renvoie le tableau {27,3}.*

4. Utiliser la fonction dateOfDayNumber pour afficher la date au format habituel.

**Contrat:**

*Le programme affiche*

*Il y a 12 évènements dans l'agenda.  
 Ven 1/1 Jour de l'An  
 Dim 27/3 3 Pâques  
 Lun 28/3 Lundi de Pâques  
 ...*

□

### Exercice 3 (Modifier l'agenda, \*\*)

1. Après avoir chargé l'agenda, et avant de l'afficher, ajoutez dans l'agenda votre anniversaire. Vérifiez qu'il s'affiche à la bonne place.
2. Ajoutez dans l'agenda l'évènement Bac sur chacune des entrées du 15 au 22 juin inclus.

**Contrat:**

*Le programme affiche*

*...  
 Mer 15/6 Bac  
 Jeu 16/6 Bac*

*Ven 17/6 Bac*

...

3. Ajoutez dans l'agenda toutes les dates de séances de TP mini-projet avec leur numéro.

**Contrat:**

*En supposant que votre TP de mini-projet a lieu le lundi, le programme affiche*

...

*Lun 12/9 TP 1 IP1*

*Lun 19/9 TP 2 IP1*

*Lun 26/9 TP 3 IP1*

...

*Lun 28/11 TP 12 IP1*

□

#### Exercice 4 (Question bonus : vue annuelle, \*\*\*)

Écrivez une fonction qui prend en paramètre un tableau représentant un agenda et qui affiche mois par mois, et semaine par semaine, les dates contenant des événements, en les faisant apparaître dans la colonne du jour de la semaine correspondant.

**Contrat:**

*En supposant que l'agenda est celui des questions précédentes, la fonction affiche*

Janvier

```
-----  
Lun Mar Mer Jeu Ven Sam Dim  
      01 ... ..  
... ..  
... ..  
... ..  
... ..
```

Février

```
-----  
Lun Mar Mer Jeu Ven Sam Dim  
... ..  
... ..  
... ..  
... ..  
...
```

Mars

```
-----  
Lun Mar Mer Jeu Ven Sam Dim  
      ... ..  
... ..  
... ..  
... .. 27  
28 ... ..
```

etc.

□