IAD : TP2.1 — Implémentation machine

BUT1INFO1R1.01
version 2021-2022

Table des matières

1. Environnement de travail

Tout comme dans le TP1, réaliser les actions suivantes pour configurer votre environnement de travail.

1. Lancer IntelliJ.
2. Créer et nommer (nous vous suggérons `TP2_NumeroDeGroupe_Nom_Prenom`) un nouveau projet où vous souhaitez placer vos sources java.

2. Validité d'une date

2.1. Feuille de route

Nous avons vu à la séance précédente les opérations qui nous permettent de déterminer si une date est valide, ainsi qu'une structuration en trois modules (ou sous-programmes, ou programmes, ou fonctions) qui facilite notre compréhension et rend notre pseudo-code lisible et efficace.

Mais il ne s'agissait que d'un pseudo-code, écrit en langage algorithmique. Il nous reste trois étapes à franchir pour aboutir à un programme complet :

- Traduire le pseudo-code algorithmique en langage Java
- S'occuper de la saisie au clavier de la date par l'utilisateur, ainsi que de l'affichage du résultat (la date est-elle valide ou non) à l'écran
- Faire le lien entre les deux

2.2. Identification des années bissextiles

On se propose dans un premier temps d'expérimenter Java en se concentrant sur le calcul des années bissextiles.

Le compilateur Java n'est pas très ouvert d'esprit : on ne peut lui parler qu'en terme de classes. Pour le contenter, créons donc une nouvelle classe **`DatesEtCompagnie.java`** et incluons-y un squelette de programme qui détermine si une année est bissextile, ou non.

Exemple 1. DatesEtCompagnie.java

```
import java.util.Scanner ;

public class DatesEtCompagnie {


    /**
     * @param annee une année
     * @return vrai si l'année est bissextile, faux sinon
     */
    public static boolean estBissextile(int annee) {
        // A vous
        return false ;
    }
}
```

Compléter la fonction **`estBissextile`**.

L'opération RESTE DE LA DIVISION ENTIERE s'écrit **`%`** en **`java`**

- L'opérateur ET LOGIQUE s'écrit **`&&`** en **`java`**
- L'opérateur OU LOGIQUE s'écrit **`||`** en **`java`**
- L'opérateur de COMPARAISON D'EGALITE s'écrit **`==`** en **`java`**
- L'opérateur de COMPARAISON D'INEGALITE s'écrit **`!=`** en **`java`**

Vérifier en exécutant votre programme qu'il respecte le jeu d'essai suivant :

Tableau 1. Jeu d'essais

Donnée	Résultat
1900	faux
2000	vrai
2008	vrai
2013	faux
2016	vrai

2.3. Calcul du nombre de jours dans un mois donné

De la même manière ajoutez la fonction suivante et complétez-la pour la faire fonctionner comme prévu dans la première séance.

Exemple 2. Calcul du nombre de jours dans un mois donné

```
/**
 *  Détermine le nombre de jours dans un mois d'une année donnée
 *  @param mois
 *  @param année
 *  @return le nombre de jours dans le mois
 */
public static int nbJours(int mois, int annee) {
        // A vous
    return 0 ;
}
```

2.4. Validité

Il ne reste plus qu'à inclure et compléter la fonction suivante dans votre classe pour finaliser le calcul de la validité d'une date.

Exemple 3. Calcul de la validité d'une date

```
/**
 *  Détermine si une date est valide
 *  @param jour
 *     jour de la date
 *  @param mois
 *     mois de la date
 *  @param annee
 *     année de la date
 *  @return vrai si la date est valide, faux sinon
 */
public static boolean estValide(int jour, int mois, int annee)
{
    // Pour l'instant, on fait simplement en sorte que la classe compile.
    // cette fonction ne déterminera pas encore la validité
    return true ;
}
```

Une fois que vous considérez cette fonction comme terminée, validez votre programme sur le jeu d'essai.

3. Interactions avec l'utilisateur et calcul

On retourne au problème initial de validité de date, qui se servira de la fonction **`estValide`** précédemment écrite et supposément fonctionnelle.

Choix de résolution

À la lumière de la séance précédente, nous choisissons de décomposer la résolution du (de "raffiner le") problème de la manière suivante :

- Étape E1 : saisie des entrées utilisateur
- Étape E2 : évaluation de la validité de la date
- Étape E3 : affichage du résultat.

Nous vous proposons l'algorithme général suivant :

Algorithme général

```
DEBUT
  // Déclaration des variables
  entier jour, mois, annee
  booleen saisieValide

  // Etape E1
  // Saisie d'une date
  LIRE (clavier) jour, mois, annee

  // Etape E2
  // Vérifier que la date saisie est valide
  // ❶

  // Etape E3
  // Traitement
  SI ( saisieValide == VRAI ) ALORS
     // ECRIRE (écran) "La date : ", jour, "/", mois, "/", annee
     // ECRIRE (écran) "est une date valide"
  SINON
    ECRIRE (écran) "La date du ", jour, "/", mois, "/", annee
    ECRIRE (écran) "n'est pas une date valide"
  FINSI
FIN
```

❶ cette partie est précisée (raffinée) dans ce qui suit.

Cet algorithme général décrit une solution au problème posé, il en existe d'autres.

1. Créer l'librairie de la fonctions suivante et l'ajouter à votre classe **`DatesEtCompagnie`**.

```
    public static void saisieCalculAffichageValidite() {
        /*************
        /* E1: Saisie */
        /*************

        /***************
        /* E2: Validité */
        /***************

        /*****************
        /* E3: Affichage */
        /*****************

    }
```

2. Réviser les boîtes algorithmiques en dessinant celle de la fonction **`saisieCalculAffichageValidite`**.

3.1. Étape E1 : Saisie

On vous fournit la fonction suivante :

```
    /**
     * Fonction aidant à saisir un entier.
     * A priori, pas besoin de la modifier.
     * @return entier saisi par l'utilisateur
     */
    public static int saisieEntier() {
        System.out.println("Saisir un entier");
        Scanner clavier = new Scanner(System.in) ;
        int nombreSaisi = clavier.nextInt() ;
        return nombreSaisi ;
    }
```

1. Dessinez la boîte algorithmique de cette fonction, et réfléchissez à la manière dont vous allez pouvoir l'utiliser dans la fonction **`saisieCalculAffichageValidite`**.
2. Incluez cette fonction dans votre classe **`DatesEtCompagnie`** pour pouvoir l'utiliser. La classe `«Scanner»` se trouve dans une librairie à part, et il faut indiquer en début de fichier à Java où la trouver. Ajoutez `import java.util.Scanner;` en début de fichier.
3. Complétez l'étape E1 de la fonction **`saisieCalculAffichageValidite`**. Rappel : cette étape doit faire saisir à l'utilisateur une date.

3.2. Étapes E2 et E3 : Calcul, puis affichage

Terminer **`saisieCalculAffichageValidite`** en complétant l'étape E2, et en écrivant la dernière étape E3 d'affichage du résultat.

3.3. À ce stade…

A ce stade vous devez être satisfait·e de votre codage car tous les cas de votre jeu d'essais fonctionnent comme prévu.

4. Dates : pour aller plus loin…

4.1. … dans le système solaire

1. Retour sur les années bissextiles, en restant sur Terre : on a vu que grâce aux années bissextiles, on évite le décalage des jours dans l'année. A partir de l'algorithme de calcul des années bissextiles ci-dessus, estimer la durée moyenne d'une année terrestre, exprimée en jour.
2. Partons sur Mars : on peut estimer que l'année tropique martienne est de 668,5921 jours. Définir une règle pour (un algorithme sur) les années bissextiles martiennes pour éviter les décalages. (Vous pouvez inventer des mois à vous le désirez.)

4.2. Un programme pour prévoir son âge

Une personne imbue d'elle-même (vous) souhaite écrire un programme qui lui permette de connaître quel âge elle aura ce cet année à une date donnée.

1. Ajouter le squelette du programme ci-dessous dans votre classe `DatesEtCompagnie`.

```
    /**
     * Determine l'âge de VOTRE_NOM à une date donnée.
     *
     */
    public static int ageDeVOTRE_NOM(int jour, int mois, int annee,
            int jourNaissance, moisNaissance, anneeNaissance,
    age;

        /* Initialisation de votre date de naissance */

        /* Calcul de votre âge à la date indiquée */

        return age;
    }
```

2. Remplacer dans le code les deux occurrences de `VOTRE_NOM` par votre vrai nom.
3. Déterminer *à la main* l'âge que vous aviez le 1 janvier 2014 et l'âge que vous aviez le 31 décembre 2014.

La création du jeu de tests est l'occasion de mettre en lumière les raisonnements qui vous permettront de répondre à la question posée. Votre algorithme sera une formalisation de ces raisonnements.

4. Écrire et programmer l'algorithme qui résout le problème posé.
5. Passer le jeu de tests ci-dessous.

Tableau 2. Jeu de tests pour le calcul de l'âge

Données	Résultat attendu
Votre date de naissance	0
1/1/2014	calculé précédement
31/12/2014	calculé précédement
Votre prochain anniversaire	Votre age + 1
La veille de votre prochain anniversaire	Votre age
Un jour du mois précédent votre prochain anniversaire	Votre age
Le lendemain de votre prochain anniversaire	Votre age + 1
Un jour du mois suivant votre prochain anniversaire	Votre age + 1

5. Avant de partir

1. Enregistrez votre programme **`DatesEtCompagnie.java`** sur **`webetud2`**
2. N'oubliez pas de vous déconnecter