TP 3 : Méthodes

Informatique Fondamentale (IF121)

27–31 octobre 2003

L'objectif de ce TP est d'apprendre à se servir des méthodes (parfois aussi nommées « fonctions »). Comme les TP précédents, il fera l'objet de plusieurs séances.

Il est vivement recommandé de sauvegarder vos programmes sur disquette afin de pouvoir les réutiliser lors des séances prochaines.

1 Rappel de syntaxe

Pour **définir une méthode** :

```
static <type> <nom de la méthode> (<type du paramètre> <nom du paramètre>) {
   <Instruction>;
   <Instruction>;
   ...
}
```

Exemple :

```
static double carré (double x) {
   return x*x;
}
```

Pour **faire appel à une méthode** : `<nom de la méthode>(<une valeur correspondante>)`

Exemples : `carré(5)`, `Deug.length(str)`, ...

Les deux programmes suivants sont équivalents :

```
import fr.jussieu.script.Deug;
class Carre {
   static double carré (double x) {
      return x*x;
   }

   static void main (String[] args) {
      Deug.print("Donnez un nombre réel : ");
      double y = Deug.readDouble();


      Deug.println("Son carré est égal à "
         + carré(y));
   }
}
```

```
import fr.jussieu.script.Deug;
class Carre {




   public static void main (String[] args) {
      Deug.print("Donnez un nombre réel : ");
      double y = Deug.readDouble();
      double x = y;
      double resultat = x*x;
      Deug.println("Son carré est égal à "
         + resultat);
   }
}
```

Une méthode peut

- prendre plusieurs paramètres
  - de même type : `double max (double x, double y)`

    $\Longrightarrow$ `max(4.0, 3.0)`

- de types différents : `Deug.subString (String s, int début, int fin)`
  $\Longrightarrow$ `Deug.subString("Bonjour", 3, 7)`
- ne pas prendre de paramètres : `Deug.readInt()`
- retourner une valeur : *tous les exemples ci-dessus*
- ne pas retourner de valeur :

```
static void afficherEnGros (String str) {
   Deug.println("************************");
   Deug.println("* " + str + " *");
   Deug.println("************************");
}
```

2 Lecture d'un programme

Exercice 1
Pour chacun des programmes suivants, indiquer (de tête) ce qu'il affiche. Attention, un programme incorrect s'est glissé dans le lot.

```
import fr.jussieu.script.Deug;
class P1 {
    static int f (int x) {
        x = x + 2;
        return x;
    }
    static void main (String[] args) {
        int x = 5;
        Deug.println(f(x));
    }
}
```

```
import fr.jussieu.script.Deug;
class P2 {
    static int f (int y) {
        y = y + 2;
        return y;
    }
    static void main (String[] args) {
        int x = 5;
        Deug.println(f(x));
    }
}
```

```
import fr.jussieu.script.Deug;
class P3 {
    static int f (int y) {
        int x;
        x = x + 2;
        return x;
    }
    static void main (String[] args) {
        int x = 5;
        Deug.println(f(x));
    }
}
```

```
import fr.jussieu.script.Deug;
class P4 {
    static int f (int y) {
        int x = y + 2;
        return y;
    }
    static void main (String[] args) {
        int x = 5;
        Deug.println(f(x));
    }
}
```

```
import fr.jussieu.script.Deug;
class P5 {
    static int f (int x) {
        x = x + 2;
        return x;
    }
    static void main (String[] args) {
        int x = 5;
        Deug.println(f(x) + f(x));
    }
}
```

```
import fr.jussieu.script.Deug;
class P6 {
    static int f (int x) {
        x = x + 2;
        return x;
    }
    static void main (String[] args) {
        int x = 5;
        Deug.println(f(f(x)));
    }
}
```

Exercice 2
Que pensez vous du bout de code suivant ?

```
static double produit (double x,double y) {
    double x = Deug.readDouble();
    double y = Deug.readDouble();
    double prod = x * y;
    return prod;
}
```

Exercice 3
Que pensez vous du code suivant ?

```
class test{
    static double produit (double x,double y) {
        double prod = x * y;
        return prod;
    }
    static void main (String[] args){
        double x = Deug.readDouble();
        double z = Deug.readDouble();
        double y = produit(x,z);
        Deug.println(y);
    }
}
```

3 Exercices de base

Remarque. En TP, lorsqu'un exercice demande d'« écrire une méthode », il faudra aussi la tester en l'incluant dans un programme complet qui l'appelle avec des arguments appropriés.

Exercice 4

1. Écrire une méthode `max` qui calcule le maximum de deux nombres.
2. Grâce à `max`, écrire une méthode `max3` qui calcule le maximum de trois nombres, sans utiliser de `if`.

Exercice 5

1. Écrire une méthode `delta` qui, à partir des trois coefficients d'un trinôme du second degré à coefficients réels, calcule le discriminant de l'équation.
2. En se servant de la méthode `delta`, écrire une méthode déterminant le nombre et les solutions d'une équation de second degré. Pour cela on utilisera la méthode `Math.sqrt` qui calcule la racine carré d'un nombre de type double.

Exercice 6
Il est possible de comparer les caractères. Les caractères sont ordonnés. Les majuscules sont classées par ordre alphabétique. Il en est de même pour les minuscules. Ainsi, l'expression `'a'<'b'` est évaluée à `true` alors que l'expression `'R'<='E'` est évaluée à `false`.

A l'aide des éléments ci-dessus, écrire des méthodes qui renvoient

1. `true` si un caractère donné c est une lettre (miniscule ou majuscule).
2. `true` si la lettre transmise en paramètre est une majuscule et `false` si c'est une miniscule (on supposera que la méthode est toujours appelée correctement)

4 Les jours de la semaine

N'oubliez pas de *tester* vos méthodes au fur et à mesure !

Exercice 7
Écrire une méthode `nomDuJour` qui prend comme paramètre un nombre de 0 à 6 et qui renvoie le jour de la semaine correspondant sous forme d'une chaîne de caractères. On fera correspondre 0 à dimanche, 1 à lundi, etc.

Exercice 8
Rajouter une méthode `jourDansLAnnee` qui prend comme paramètre une date en 2003 sous forme de deux entiers — jour et mois — et renvoie le nombre de jours passés depuis le début de l'année.

Exercice 9
Modifier le programme pour qu'il lise en entrée une date en 2003 (par exemple 14 07) et affiche le jour de la semaine correspondant. (Sachez que 31/12/2002 était un mardi).

Exercice 10
Rajouter une méthode `compteBissextiles` prenant comme paramètres deux années arbitraires et qui renvoie le nombre d'années bissextiles entre les deux.

On rappelle qu'une année a est bissextile si et seulement si elle est multiple de 4 mais pas de 100, ou si elle est multiple de 400.

Exercice 11
En utilisant les méthodes ci-dessus écrire un programme qui prend en entrée n'importe quelle date après le 01/01/2003 et qui affiche le jour de la semaine correspondant.

Quel jour de la semaine sera le Noël 2006 ? Et votre 26$^{\text{ème}}$ anniversaire ?

Exercice 12
Modifier le programme afin de pouvoir aussi traiter des dates avant 2003.

Quel était le jour de la semaine quand vous êtes né ?

Exercice 13
Dans de nombreux exercices nous demandons à l'utilisateur de saisir une date. Pour cela on lui demande d'entrer le jour puis le mois et enfin l'année. Pour simplifier cela, on souhaite demander à ce dernier de rentrer une date sous la forme : jj/mm/aaaa.

À l'aide des fonctions ci-dessous, écrire des méthodes `jourDeDate`, `moisDeDate` et `anneeDeDate` permettant de récupérer le jour, le mois et l'année à partir d'une date au format jj/mm/aaaa. L'inclure dans le programme de l'exercice précédent.

Pour manipuler des chaînes de caractères, on dispose (entre autres) des méthodes suivantes :

`Deug.length(s)`	longueur (nombre de caractères) de la chaîne `s`
`Deug.charAt(s, p)`	caractère de la chaîne `s` à la position `p` (en partant de 0)
`Deug.subString(s, début, fin)`	sous-chaîne de `s`, composée des caractères des positions `début` incluse à `fin` exclue
`Deug.stringToInt(s)`	convertit la chaîne `s` en le nombre qu'elle représente (par exemple, convertit la chaîne `"42"` en l'entier 42)

Par exemple, `Deug.charAt(s, Deug.length(s)-1)` renvoie le dernier caractère de `s`, tandis que `Deug.subString(s, 3, Deug.length(s))` renvoie `s` privée de ses trois premiers caractères.