

IF121 - TP4

3 Novembre 2004

* facile, ** moyen, *** difficile

1 Rappel syntaxe

1.1 Méthodes

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>*)

Exemple : carré(5), Deug.length(str), ...

Les deux programmes suivants sont équivalents :

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

Attention !! Une méthode peut

- prendre plusieurs paramètres
 - de même type : double max (double x, double y)
 - de types différents : Deug.substring (String s, int début, int fin)
- 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("*****");
}
```

1.2 For

La structure générale d'une boucle **for** est :

```
for (<initialisation>; <condition>; <expression> ) <instruction>;
ou
for (<initialisation>; <condition>; <expression> ) {
<Instruction>;
<Instruction>;
...
}
```

- *initialisation* est évaluée avant d'entrer dans la boucle.
- *condition* est évaluée au début de chaque itération pour déterminer si la boucle doit être poursuivie. Elle doit être de type booléen.
- *expression* est évaluée à chaque itération et vise à mettre à jour la variable de contrôle.

Exemple : `for (i = 0 ; i < 10 ; i++) Deug.println("coucou");`
 Cette boucle aura pour effet d'afficher 10 fois de suite *coucou*.

1.3 While

La structure générale d'une boucle **while** est :

```
while ( <condition> ) <instruction>;
ou
while ( <condition> ) {
<Instruction>;
<Instruction>;
...
}
```

condition est évaluée au début de chaque itération pour déterminer si la boucle doit être poursuivie. Elle doit être de type booléen.

Exemple :

```
i = 0;
while( i < 10) {
    Deug.println("coucou");
    i++;
}
```

Cette boucle aura pour effet d'afficher 10 fois de suite *coucou*.

2 Méthodes

Exercice 1 : *Plus grand entier **

Ecrire une fonction qui prend en argument 2 entiers et qui renvoie le maximum.

Exercice 2 : *Plus grand entier (bis) **

Ecrire une procédure qui prend en argument 3 entiers, et qui affiche le maximum, en utilisant la fonction de l'exercice précédent.

3 Itérations simples

Exercice 3 : *Affichage des entiers de 0 à N* *

Ecrire un programme qui lit un entier $N > 0$ et qui affiche tous les entiers de 0 à N .

Exercice 4 : *Sommes* **

Ecrire un programme qui lit un entier $N > 0$ et qui calcule :

- la somme $1 + 2 + \dots + N$.
- la somme $1^3 + 2^3 + \dots + N^3$.

Exercice 5 : *Factorielle* **

Ecrire un programme qui lit un entier $N > 0$ et qui calcule $N!$.

Exercice 6 : *Maximum de N entiers* **

Ecrire un programme qui lit un entier $N > 0$, et ensuite lit N entiers strictement positifs, et affiche leur maximum.

Exercice 7 : *Fibonacci* ***

Ecrire un programme qui lit un entier $N > 0$ et qui calcule $fib(N)$ où :

$$fib(N) = \begin{cases} 1 & \text{si } N = 0 \\ 1 & \text{si } N = 1 \\ fib(N - 1) + fib(N - 2) & \text{sinon} \end{cases}$$

Exercice 8 : *Algorithme d'Euclide* ***

Etant donnés 2 nombres a et b , écrire un programme qui calcule $pgcd(a, b)$. L'algorithme est basé sur les observations suivantes :

- si $b|a$ alors $pgcd(a, b) = b$.
- si $a = bq + r$, avec q et r entiers, alors $pgcd(a, b) = pgcd(b, r)$.