TP 4 : premières boucles

Informatique Fondamentale (IF121)
10–14 novembre 2003

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.

Rappel. Sauf si l'énoncé le précise explicitement, une méthode n'est pas censée lire ou afficher quoi que ce soit.

1 Suites numériques

Exercice 1 : *Puissance d'un nombre*

Écrire une méthode qui calcule la puissance $n^{\text{ème}}$ d'un nombre, ou n est un entier positif. Par exemple, `puissance(2.5, 3)` vaut $15,625$.

Modifier la méthode précédente pour traiter le cas où n est un entier quelconque. Par exemple, `puissance(2.5, -3)` vaut $0,064$; `puissance(2.5, 0)` vaut 1.

Exercice 2 : *Factorielle*

On rappelle que la fonction factorielle est définie sur les entiers positifs de la façon suivante :

$$\begin{aligned} &\text{factorielle}(0) = 1 \\ &\text{factorielle}(n) = n \times \text{factorielle}(n-1) && \text{si } n \geqslant 1 \end{aligned}$$

Écrire une méthode qui calcule la factorielle d'un entier.

Exercice 3 : *Une suite*

On définit la suite $(x_n)_{n\in\mathbb{N}}$ de la manière suivante : $x_0 = a$ et $x_{n+1} = 4x_n(1-x_n)$. Écrire une méthode qui calcule x_n en fonction de a et de n.

Exercice 4 : *Approximation de $\sqrt{x}$*

Étant donné un réel strictement positif a, on définit la suite réelle $(x_n)_{n\in\mathbb{N}}$ de la manière suivante :

$$x_0 = a \qquad \text{et} \qquad x_{i+1} = \frac{x_i^2 + a}{2x_i}$$

On admet que cette suite converge vers $\sqrt{a}$.

Écrire un programme qui lit un réel positif a et un entier n et qui renvoie une valeur approchée `b` de $\sqrt{a}$ en utilisant l'approximation x_n. Afficher aussi `b*b` pour vérifier.

Exercice 5 : *Une série*

Écrire une méthode qui calcule la somme des carrés des n premiers entiers. Par exemple, `sommeCarres`$(5) = 1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$.

Exercice 6 : *Une autre série*

Écrire une méthode qui calcule la somme des carrés des n premiers entiers *impairs*. Par exemple, `sommeCarresImpairs`$(5) = 1^2 + 3^2 + 5^2 + 7^2 + 9^2 = 165$.

2 Suites numériques (encore)

Exercice 7 : *Deux suites*

On définit une suite double :

$$\begin{aligned} u_0 &= 1 & \qquad u_{n+1} &= (u_n + v_n)/2 \\ v_0 &= 2 & \qquad v_{n+1} &= \sqrt{u_{n+1} v_n} \end{aligned}$$

On admet que les suites (u_n) et (v_n) sont adjacentes de limite $\sqrt{27}/\pi$. Écrire un programme qui lit un entier n et affiche l'approximation du nombre π obtenue à partir de v_n.

Note : la méthode `Math.sqrt` permet de calculer la racine carrée d'un nombre.

Exercice 8 : *Suite de Fibonacci*

On définit la suite de Fibonacci de la manière suivante :

$$F_0 = F_1 = 1$$
$$F_{n+2} = F_n + F_{n+1}$$

Écrire une méthode qui calcule le $n^{\text{ème}}$ nombre de Fibonacci (c'est-à-dire F_n).

Exercice 9 : *Somme des puissances $k^{\text{èmes}}$*

Écrire une méthode qui, étant donnés deux entiers k et n, calcule la somme des puissances $k^{\text{èmes}}$ des n premiers entiers. Par exemple, `sommePuissances`$(3, 5) = 1^3 + 2^3 + 3^3 + 4^3 + 5^3 = 225$. (On réutilisera la méthode `puissance` de l'exercice 1.)

3 Affichage

Exercice 10 : *Compter*

Écrire un programme qui lit un entier n, puis qui affiche tous les entiers de 1 à n (un par ligne).

Exercice 11 : *Un rectangle*

(a) Écrire un méthode `repetePrint` qui prend comme argument un entier n et un caractère, et qui affiche n fois de suite (sans passer à la ligne) le caractère en question.

(b) À l'aide de la méthode `repetePrint`, écrire un programme qui affiche (sous forme de texte) un rectangle de côtés donnés. On utilisera les caractères `-`, `|` et `+`. Par exemple :

```
> java Rectangle
Largeur : 10
Hauteur : 5
+--------+
|        |
|        |
|        |
+--------+
```

4 Traitement de données

Exercice 12 : *Calcul de moyenne*

Écrire un programme qui lit un entier n, puis qui lit n nombres, et affiche la moyenne de ces nombres. Exemple :

```
> java Moyenne
Combien de nombres ? 4
Nombre 1 : 3
Nombre 2 : 5
Nombre 3 : 1
Nombre 4 : 55
La moyenne est 16.
```

Exercice 13 : *La ligne la plus longue*

Écrire un programme qui lit un entier n, puis qui lit n lignes, et affiche la ligne la plus longue. Exemple :

```
> java LigneLaPlusLongue
Combien de lignes ? 5
[1] Ceci
[2] est
[3] un
[4] essai
[5] ...
Ligne la plus longue : essai
```

Qu'affiche votre programme s'il y a des ex-aequos ?

5 Manipulation de texte

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

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.

Exercice 14 : *Somme des chiffres*

La méthode `Deug.intToString` permet d'obtenir la représentation décimale d'un nombre entier. Par exemple `Deug.intToString(42)` est la chaîne `"42"`.

Étant donné un caractère `c` représentant un chiffre, on peut obtenir le nombre entier correspondant par `c - '0'`. Par exemple, l'extrait de programme suivant affiche `6` :

```
char c = '3';
int n = 2 * (c - '3');
Deug.println(n);
```

Écrire une méthode qui calcule la somme des chiffres d'un nombre. Par exemple, `sommeChiffres(1789)` renvoie 25.

Exercice 15 : *Compter les voyelles*

Écrire un programme qui lit une ligne et affiche le nombre de voyelles qu'elle contient.