TP 5 : tableaux

Informatique Fondamentale (IF121)
17–21 novembre 2003

Exercice 1 : *Arguments d'un programme*

L'argument de la méthode `main` est le tableau des arguments du programme. Par exemple, si on exécute le programme `Toto` par

```
java Toto bonjour tout le monde
```

la méthode `main` de `Toto` reçoit un tableau à 4 éléments : `"bonjour"`, `"tout"`, `"le"` et `"monde"`.

Écrire un programme `Echo` qui affiche ses arguments, séparés par des espaces.

1 Tableaux de nombres

Exercice 2 : *Lecture d'un tableau de nombres*

Écrire une méthode `litTableau` qui lit un entier n, puis qui lit un n nombres réels qu'elle place dans un tableau, et qui renvoie ce tableau.

Exercice 3 : *Norme, produit scalaire*

Il est naturel de représenter le vecteur de coordonnées $(x_0, ..., x_{n-1})$ (dans un espace vectoriel de dimension n) par le tableau dont ces n nombres sont les éléments.

(a) Écrire une méthode `norme` qui calcule la norme d'un vecteur. On rappelle que la norme du vecteur $\vec{x}$ de coordonnées $(x_0, ..., x_{n-1})$ est donnée par

$$\|\vec{x}\| = \sqrt{\sum_{i=0}^{n-1} x_i^2} = \sqrt{x_0^2 + x_1^2 + \cdots + x_{n-1}}$$

(b) Écrire une méthode qui calcule le produit scalaire de deux vecteurs, dont on rappelle l'expression :

$$\vec{x} \bullet \vec{y} = \sum_{i=0}^{n-1} x_i y_i$$

Exercice 4 : *Norme du sup*

On appelle « norme du sup » du vecteur $\vec{x}$ de coordonnées $(x_0, ..., x_{n-1})$ le nombre réel positif donné par

$$\|\vec{x}\|_\infty = \max\{|x_0|, |x_1|, \ldots, |x_n|\}$$

Écrire une méthode qui calcule la norme du sup d'un vecteur.

Exercice 5 : *Concaténation*

Écrire une méthode `concatene` qui prend comme arguments deux tableaux `t1` et `t2` et qui construit et renvoie un tableau `t` contenant les éléments de `t1` suivis des éléments de `t2`.

Exercice 6 : *Polynômes*

Soit un polynôme à coefficients réels et de degré n :

$$P(X) = \sum_{i=0}^{n} a_i x^i = a_n X^n + a_{n-1} X^{n-1} + \cdots + a_1 X + a_0$$

(a) Écrire une méthode `puissance` qui calcule la puissance $n^{\text{ème}}$ d'un nombre réel. (Cf. TP 4, ex. 1)

(b) À l'aide de cette méthode `evaluePolynome`, écrire une méthode qui évalue un polynôme P en un point x : cette méthode prend en argument le tableau des coefficients de P et le nombre x, et renvoie le nombre $P(x)$.

(c) Si l'on suit le déroulement de l'exécution de la méthode `evaluePolynome`, on observe qu'elle fait beaucoup de multiplications inutiles (quand on calcule x^{i+1}, on commence par recalculer x_i). Un moyen d'éviter cela est la *méthode de Horner*. Elle consiste à remarquer qu'on peut écrire

$$P(X) = (((\ldots(a_n X + a_{n-1})X + \ldots)X + a_2)X + a_1)X + a_0$$

Écrire une méthode `horner` qui évalue un polynôme en un point par cette technique.

2 Tableaux triés

Exercice 7 : *Vérification de tri*

Écrire une méthode qui prend comme argument un tableau de nombres et qui vérifie si ce tableau est trié en ordre croissant.

Exercice 8 : *Fusion de deux tableaux triés*

Écrire une méthode qui prend comme arguments deux tableaux de nombres `t1` et `t2`, chacun trié en ordre croissant, et qui construit un nouveau tableau `t` trié contenant les mêmes éléments. Si un nombre apparaît k_1 fois dans `t1` et k_2 fois dans `t2`, il doit apparaître $k_1 + k_2$ fois dans `t`. Par exemple, la fusion de $(3, 3, 6, 7)$ avec $(1, 2, 3, 5, 6)$ est $(1, 2, 3, 3, 3, 5, 6, 6, 7)$.

3 Statistiques

Exercice 9 : *Constitution d'un histogramme*

Un examen est noté sur 10, par points entiers (c'est-à-dire que les notes possibles sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 et 10). À fins statistiques, on veut savoir combien d'élèves ont chaque note. On suppose qu'il y a e étudiants, et qu'on dispose d'un tableau `notes` tel que `notes[i]` est la note du $i^{\text{ème}}$ étudiant. Construire un tableau `stats` tel que `stats[n]` contienne le nombre d'étudiants qui ont eu la note `n`.

4 Chaînes de caractères

Une chaîne de caractères ressemble fortement à un tableau qu'on ne peut pas modifier. Pour accéder au `i`$^{\text{ème}}$ caractère de la chaîne `s`, on utilise `Deug.charAt(s,i)`. La longueur de la chaîne `s` est donnée par `Deug.length(s)`.

Exercice 10 : *Palindrome*

Écrire un programme qui lit un mot et qui teste si ce mot est un palindrome, c'est-à-dire si on obtient le même mot en le lisant à l'envers. Par exemple, « kayak », « non », « ressasser », « selles » sont des palindromes ; « ânonna », « ressasse », « salles » n'en sont pas.

Exercice 11 : *Recherche de « en »*

Écrire un programme qui lit une ligne de texte et qui compte le nombre de fois que cette ligne contient la lettre « e » suivie immédiatement de la lettre « n ». Par exemple, la phrase précédente contient 2 fois le fragment de texte « en », et celle-ci 4 fois.