

Contrôle de TP (M3)

Informatique Fondamentale (IF1)

jeudi 7 décembre 2006

À chaque exercice doit correspondre un fichier `.java`. Ces fichiers seront stockés dans un répertoire nommé `exam`. Lorsque vous aurez fini, venez nous voir pour que nous enregistrions vos fichiers.

Quelques conseils :

- commencez le programme par une zone de commentaires indiquant le nom de votre groupe (M3), votre nom, la date et le numéro de l'exercice, par exemple :

```
/* M3 - Francis Dupont - 7/12/2006 - exercice 12 */
```

- indentez correctement votre programme ;
- utilisez des noms de variables significatifs (évités les noms de variables d'une lettre) ;
- toute utilisation d'une fonction `Deug.read` doit être précédée d'une instruction `Deug.println` qui indique à l'utilisateur ce qu'on attend de lui.

Exercice 1.

Vitesse. Écrire un programme `Vitesse.java` qui demande à l'utilisateur combien de kilomètres il a parcouru lors de son dernier trajet en voiture et combien de temps (en heures) ce trajet a duré, et calcule la vitesse moyenne de la voiture durant ce trajet. Par exemple :

Distance ?

43

Durée ?

2

Vitesse moyenne : 21.5 km/h.

Exercice 2.

Nombre de mots. Écrire un programme `Mots.java` qui lit une phrase à l'aide de la fonction `Deug.readLine()` et compte le nombre de mots qu'elle contient. On considérera que deux mots sont toujours séparés soit par une espace, soit par une apostrophe.

Mot le plus long ().** Modifiez le programme précédent pour qu'il affiche aussi le nombre de lettres du mot le plus long de la phrase.

Exercice 3.

Miroir. Écrire un programme `Miroir.java` qui demande un mot à l'utilisateur et l'affiche à l'envers. Exemple : Miroir devient rioriM.

Palindrome¹. Écrire un programme `Palindrome.java` qui teste si un mot rentré par l'utilisateur est identique qu'on le lise de gauche à droite ou de droite à gauche. (Rappel : le test d'égalité sur les chaînes de caractères est effectué avec `Deug.equals`).

Exemple de palindrome : ressasser.

¹On désigne généralement par palindrome un mot dont l'ordre des lettres reste le même qu'on le lise de gauche à droite ou de droite à gauche.

Exercice 4.

Attention, les questions qui suivent sont liées et demandent d'écrire les fonctions d'une même classe nommée `Polynome`.

On considérera qu'un polynôme $P(x) = \sum_{i=0}^{n-1} p_i x^i$, à coefficients réels, est représenté par un tableau $[p_0, p_1, \dots, p_{n-1}]$ dont les éléments sont de type `double`. Par exemple, le tableau $[5, 4.2, 9, 3]$ représente le polynôme $P(x) = 5 + 4.2x + 9x^2 + 3x^3$.

Entrée de tableaux de réels. Écrivez une fonction `litTableau` déclarée par

```
public static double[] litTableau()
```

qui lit un entier n , crée un tableau a de taille n , puis lit n réels qu'elle stocke dans le tableau.

Évaluateur de polynômes. Écrivez une fonction `eval` déclarée par

```
public static double eval(double[] polynome, double x)
```

qui évalue le polynôme, dont les coefficients sont stockés dans le tableau `polynome`, au point `x`. Plus précisément, si on note p_0, p_1, \dots, p_{n-1} les éléments du tableau représentant le polynôme P , la fonction doit renvoyer $P(x)$, c'est-à-dire $\sum_{i=0}^{n-1} p_i x^i$.

Dérivateur de polynômes. Écrivez une fonction `derive` déclarée par

```
public static double[] derive(double[] polynome)
```

qui prend en argument un polynôme P représenté par le tableau `polynome` et renvoie le tableau correspondant au polynôme P' dérivé de P .

On rappelle que si $P(x) = \sum_{i=0}^{n-1} p_i x^i$, sa dérivée est $P'(x) = \sum_{i=0}^{n-2} (i+1)p_{i+1} x^i$.

Fonction principale. Écrivez une fonction `main` qui demande un tableau correspondant à un polynôme P et un réel x et affiche les résultats du calcul de $P(x)$ et $P'(x)$.

Exercice 5.

Devinez le nombre. Écrivez un programme `Devinette.java` qui fait deviner un nombre secret compris entre 0 et 100 à quelqu'un. Ce programme commencera par initialiser une variable `secret` par

```
int secret = (int)(Math.random() * 100);
```

ce qui donne une valeur différente à `secret` à chaque exécution du programme. Puis le programme doit faire deviner cette valeur à l'utilisateur en lui demandant d'entrer un nombre et en lui indiquant si celui-ci est supérieur ou inférieur à la valeur secrète jusqu'à ce qu'il trouve le bon nombre.

Par exemple :

Devinez le nombre :

42

Trop grand !

Devinez le nombre :

21

Trop bas !

Devinez le nombre :

23

Trop bas !

Devinez le nombre !

25

Bravo, vous avez gagné !

Plus difficile. Modifiez votre programme pour que l'utilisateur perde s'il n'a pas trouvé au bout de dix essais.