

TP Algorithmique programmation 1 NFA005

Session : 1
Date : Mardi 6 juin 2006
Horaire : de 18h30 à 21h30
Durée : 3 h

Documents manuscrits et photocopiés autorisés.

Les 2 exercices sont indépendants.
Le total des points est 22, ce qui est normal.
Le sujet comporte 3 pages.

1. STATEXAM

(notation : 8 points)

Le but de l'exercice est d'écrire un programme **StatExam** qui, étant donné un ensemble de notes (sur 20), affiche le nombre d'occurrences de chacune des notes ainsi que leur fréquence.

Exemple d'exécution :

```
$ java StatExam 10 12 15 12 8 12 15 3
```

```
note 3   : 1           12.5%
note 8   : 1           12.5%
note 10  : 1           12.5%
note 12  : 3           37.5%
note 15  : 2           25.0%
```

On écrira les méthodes suivantes :

1. Une méthode **creerTableau** qui fabrique un tableau de dimension 21 correspondant aux notes de 0 à 20, qui garnit ce tableau avec le nombre d'occurrences de chacune des notes du tableau de notes passé en paramètre et qui retourne une référence sur ce tableau.

Exemple : `creerTableau` appliqué au tableau suivant :

10	12	15	12	8	12	15	3
----	----	----	----	---	----	----	---

Retourne une référence sur un tableau contenant :

0	0	0	1	0	0	0	0	1	0	1	0	3	0	0	2	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2. Ecrire une méthode **afficherTableau** qui reçoit en paramètre la référence sur un tableau créé par un appel à la méthode **creerTableau** et qui produit l'affichage souhaité.
3. Tester votre programme en écrivant une fonction **main**.

2. GEOMETRIE

(notation : 14 points)

On souhaite écrire un ensemble de classes permettant de manipuler les figures géométriques du plan.

1. Ecrire une classe **Point** qui représente les points du plan sachant qu'un objet **Point** a deux attributs : ses coordonnées dans le plan (son abscisse et son ordonnée) qui sont des nombres réels.
La classe comportera un constructeur prenant deux nombres réels en arguments ainsi que des accesseurs et des modificateurs pour chacun des attributs.

2. Ecrire une interface **Figure** comportant les trois méthodes suivantes :
 - **appartient** qui indique si le **Point** en paramètre appartient ou non à la figure géométrique implémentée.
 - **aire** qui retourne l'aire de la figure géométrique implémentée.
 - **perimetre** qui retourne le périmètre de la figure géométrique implémentée.

3. Ecrire une classe **Cercle** implémentant **Figure**. Un objet **Cercle** a pour attribut un objet **Point** qui représente son centre et un nombre réel positif pour son rayon.

Indications : la distance $d(A,B)$ entre les points A de coordonnées (x_A, y_A) et le point B (x_B, y_B) est calculée par la formule : $\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$. Ainsi, le point P appartient au disque de centre C et de rayon R si $d(P,C) \leq R$.

La fonction Java **Math.sqrt(x)** permet de calculer \sqrt{x} .

4. On souhaite traiter les rectangles de côtés parallèles aux axes du plan. Un objet **Rectangle** est défini par un objet **Point** qui est son sommet en bas à gauche et par deux nombres réels positifs : sa longueur et sa largeur. Ecrire une classe **Rectangle** implémentant **Figure**.
5. On souhaite traiter les carrés de côtés parallèles aux axes du plan.
Un carré peut être vu comme un rectangle dont la longueur et la largeur sont égales.
Compte tenu de cette remarque, quel mécanisme du langage Java peut-on utiliser pour écrire une classe **Carre** exploitant la classe **Rectangle** écrite à la question précédente ?
Ecrire la classe **Carre**.
6. Ecrire un programme de test.