

IPST-CNAM
Programmation JAVA
NFA 001
Mercredi 23 Février 2011

Avec document
Durée : **2 h30**
Enseignant : LAFORGUE Jacques

1^{ère} Session NFA 001

(COURS)

La durée de l'examen est de 2 heures et 30 minutes. La consultation de tous les documents fournis par l'enseignant (support de cours, exercices, solutions) est autorisée. Le barème est indicatif.

La réponse doit être sur papier. Vous devez écrire les codes commentés en Java.

1. **Probleme 1 [15mn] [2 points]**

Une banque vous permet de déposer une certaine somme d'argent sur un compte rémunéré. Le compte est rémunéré à un taux annuel donné.

Faire le programme Java qui affiche le crédit du compte pour chaque année sur 10 ans à partir de l'année 2011.

La somme d'argent, le taux et le nombre d'année sont des valeurs saisies au clavier.

Exemple :

Somme déposé = 2000 euros

Taux = 3%

Au bout d'1 an le compte est crédité de $2000 * (1 + 3/100) = 2060$ euros

Au bout de 2 ans le compte est crédité de $2060 * (1+3/100) = 2121,80$ euros

Au bout de 3 ans le compte est crédité de $2121,80 * (1+3/100) = 2185,454$ euros

...etc..

NB: Utilisez la classe Terminal vue en cours pour saisir et afficher les informations.

Exemple d'une exécution :

java Probleme1

Somme : **2000**

Taux : **3**

Nb annee : **10**

Annee 2011 : 2060.0

Annee 2012 : 2121.8

Annee 2013 : 2185.454

Annee 2014 : 2251.01762

Annee 2015 : 2318.5481486000003

Annee 2016 : 2388.1045930580003

Annee 2017 : 2459.7477308497405

Annee 2018 : 2533.5401627752326

Annee 2019 : 2609.5463676584895

Annee 2020 : 2687.832758688244

2. Problème 2 [30mn] [4 points]

1/ Faire la méthode Java : **String[] isolerMots(String phrase)** qui retourne un tableau de chaîne contenant chacun des mots de la phrase. Si le mot existe plusieurs fois dans la phrase, il ne doit exister qu'une fois dans le tableau retourné.

Dans la phrase, les mots peuvent être séparés par plusieurs blancs.

2/ Ecrire le programme principal qui saisie la phrase à l'écran, appelle la méthode *isolerMots* et affiche le tableau à l'écran (utilisez la boucle énumérative Java).

Exemple d'une exécution :

```
java Probleme2
```

```
Saisir la phrase: un deux un cinq trois deux quatre six trois
```

```
un
```

```
deux
```

```
cinq
```

```
trois
```

```
quatre
```

```
six
```

3. Problème 3 [45mn] [6 points]

Etant donné un tableau *tab* d'entiers, on définit une monotonie comme étant toute partie de taille maximale de *tab* dont les éléments sont triés par ordre croissant..

Par exemple dans le tableau {3, 5, 1, 4, 6, 2, 9}, nous avons trois monotonies : {3, 5}, {1, 4, 6}, {2, 9}.

Faire le programme Java qui affiche toutes les monotonies de la série de nombre en entrée du programme, 1 ligne à l'écran par monotonie, et un caractère blanc entre chaque entier.

Si une des valeurs en entrée du programme n'est pas un entier, la valeur est remplacée par 0 et un message d'avertissement est écrit à l'écran.

Si la valeur en entrée est négative alors elle est convertie en positif.

Rappel: la méthode `Terminal.ecrireString(String s)` écrit la chaîne *s* à l'écran sans revenir à la ligne et la méthode `Terminal.ecrireStringln(String s)` écrit la chaîne *s* à l'écran en revenant à la ligne.

Rappel : la méthode `int n = Integer.parseInt(String s)` permet de convertir une chaîne en entier et retourne une exception si la chaîne n'est pas un entier.

Exemple d'une exécution :

```
java Probleme3 1 2 3 x -5 6 2 12 5 7 2
```

```
pas un entier : x
```

```
1 2 3
```

```
0 5 6
```

```
2 12
```

```
5 7
```

```
2
```

Tournez la page →

4. Problème 4 [1h00] [8 points]

Soit un fichier texte ("Classe.txt") contenant des lignes dont le format est le suivant :
<nom>;<prenom>;<matière>;<note moyenne>

Exemple :

```
Lafont;Pierre;Histoire;13.5  
Lafont;Pierre;Geographie;12.0  
Dupont;Paul;Histoire;7  
Dupont;Paul;Geographie;9  
Dupont;Paul;Francais;13  
Dupont;Paul;Math;11  
Lafont;Pierre;Francais;8  
Lafont;Pierre;Math;18
```

1/ Faire le programme Java qui affiche toutes les notes par matière d'un élève dont les nom et prénom sont saisis.

Rappel : StringBuffer Terminal.lireFichier(String nomFichier) permet de lire le fichier et retourne son contenu dans une StringBuffer.

2/ Faire le programme Java qui calcule la moyenne de la classe pour chacune des matières. On ne sait pas à priori quelles sont les matières contenu dans le fichier. Cela veut dire qu'il faut commencer par identifier les matières contenues dans le fichier.

Exemple d'une exécution :

1/

```
java Probleme41  
Nom : Lafont  
Prenom : Pierre  
Histoire:13.5  
Geographie:12.0  
Francais:8  
Math:18
```

2/

```
java Probleme42  
Histoire:10.25  
Geographie:10.5  
Francais:10.5  
Math:14.5
```

NB: Dans les exemples d'exécution, les valeurs saisies sont en gras italique.

(Fin du sujet)