

IPST-CNAM
Programmation JAVA
NFA 001
Mercredi 22 Juin 2011

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

2ème Session NFA 001

(COURS)

La durée de l'examen est de 2 heures et 30 minutes. Les documents sont autorisés.. Le barème est indicatif.

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

1. Probleme 1 [20mn] [2 points]

Ecrire en JAVA, la méthode suivante :

```
public void ajouterTabOrd(int[] t1, int nb1, int[] t2, int nb2)
t1 est un tableau d'entiers ordonnés par ordre croissant.
nb1 est le nombre d'éléments utiles de t1
t2 est un tableau d'entier quelconque
nb2 est le nombre d'éléments utiles de t2.
```

Cette méthode ajoute dans t1 tous les éléments de t2 en conservant l'ordre croissant des éléments.

Elle déclenche une exception si le tableau t1 n'est pas assez grand pour contenir les éléments de t2.

NB: Ne pas utiliser la méthode prédéfinie *sort* de Java.

2. Problème 2 [30mn] [4 points]

Afin d'optimiser la recherche d'un mot (chaîne), on stocke tous les mots de même longueur **N** dans un même tableau, soit **tabRn** ces tableaux. Ces tableaux sont stockés dans un autre tableau, soit **tabT** ce tableau.

Le tableau des mots de longueur **N** est stocké en **tabT[N-1]**.

Le tableau **tabT** est de dimension 100 (on suppose qu'il n'existe pas de mot dont la longueur est supérieure à 100), et chacun des tableaux **tabRn** sont de dimensions 200 (on suppose que le tableau n'est jamais complètement plein).

Les mots sont rangés dans **tabRn** de gauche à droite. Les cases non occupées ont la valeur **null**. Cela veut dire que, on ajoute un mot dans un tableau **tabRn** en recherchant la 1ère valeur **null** afin d'y mettre le mot.

Au départ, le tableau **tabT** contient que des valeurs **null**. Sa déclaration est donc :

```
String[][] tabT = new String[100][];
```

1/ Faire la méthode qui ajoute un nouveau mot dans **tabT**.

2/ Faire la méthode qui recherche un mot donné dans **tabT**.

Tournez la page →

3. Problème 3 [60mn] [8 points]

Soit un fichier texte ("CompteBancaire.txt") contenant des lignes dont le format est le suivant :

DEBIT *type somme*

CREDIT *type somme*

PRLVT *origine somme*

où *type* est "CB" (transaction par Carte Bleue) ou "CH" (transaction par Chèque)

où *somme* est une valeur double toujours positive

où *PRLVT* est une *somme* (positive) prélevée sur le compte et dont l'origine est *origine*.

La première ligne du fichier est nécessairement de la forme :

SOLDE *solde*

où *solde* est un double positif ou négatif qui est le solde initial du compte

Exemple de fichier :

```
SOLDE 1000.00
PRLVT SFR 25.00
DEBIT CB 12.5
DEBIT CB 7.5
PRLVT EDF 200.0
CREDIT CH 150.0
CREDIT CH 350.0
PRLVT EDF 300.0
PRLVT SFR 25.00
DEBIT CH 1000.00
```

Exemple d'exécution avec ce fichier:

```
Solde final = -70.0
Detail CB = -20.0
Detail CH = -500.0
PRLVT SFR = 50.0
PRLVT EDF = 500.0
```

Faire le programme Java qui affiche :

- le nouveau solde du compte bancaire
- la somme des transactions de carte bleue
- la somme des transactions des chèques.
- les sommes cumulés par *origine* des prélèvements.

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

4. Problème 4 [40mn] [6 points]

Faire en JAVA, le programme complet qui permet de réviser les tables de multiplication.

Le programme demande quelles tables, de 2 à 9, il faut réviser et demande le nombre de multiplication à trouver.

Ensuite, le programme boucle et propose une multiplication dont chaque opérande est tiré aléatoirement : le premier opérande de 2 à 9 et le deuxième opérande une des tables choisies. Le programme attend une saisie. Si la valeur saisie est juste alors il affiche "JUSTE" sinon "FAUX" avec le résultat.

A la fin, le programme affiche la note correspondante au nombre de valeurs justes sur le nombre proposé, ramené à une note sur 20.

Pour réaliser ce programme, vous utilisez la méthode *int aleat(n)* qui retourne une valeur aléatoire comprise entre 0 et n-1 (inclus).

Vous utiliser aussi la classe `Terminal` vue en cours.

(Fin du sujet)