

IPST-CNAM
Programmation JAVA
NFA 001
Mercredi 23 Février 2011

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

1^{ère} Session NFA 001

(CORRECTION)

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]

```
public class Probleme1
{
    public static void main(String args[])
    {
        // Saisis des valeurs du programme
        //
        // Saisi de la somme d'argent
        Terminal.ecrireString("Somme : ");
        double somme = Terminal.lireDouble();

        // Saisi du taux
        Terminal.ecrireString("Taux : ");
        double taux = Terminal.lireDouble();

        // Saisi du nombre d'annee
        Terminal.ecrireString("Nb annee : ");
        double nbAnnee = Terminal.lireDouble();

        // Calcul du credit pour chaque année
        double credit = somme;
        for(int i=0;i<nbAnnee;i++)
        {
            int an = i+2011; // Annee
            // Calcul du credit de l'annee
            credit=credit*(1+taux/100);
            Terminal.ecrireStringln("Annee "+an+" : "+credit);
        }
    }
}
```

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]

```
import java.util.*;

public class Probleme2
{
    // Programme principal
    //
    static public void main(String args[])
    {
        Terminal.ecrireString("Saisir la phrase: ");
        String phrase = Terminal.lireString(); // Saisi
        String[] mots = isolerMots(phrase); // Traitement
        for(String m:mots) // Affichage
            Terminal.ecrireStringln(m);
    }

    // Méthode qui isole de maniere unique chacun des mots d'une phrase
    //
    static String[] isolerMots(String phrase)
    {
        // On isole les mots
        StringTokenizer strtok = new StringTokenizer(phrase);
        int n = strtok.countTokens();

        // Tableau qui contiendra les mots
        // au maximum ce tableau est de la dimension des mots
        // de la phrase
        String[] tabMots = new String[n];
        int nbMots=0;

        // On parcourt les mots de la phrase et on l'ajoute dans le
        // tableau de mot s'il n'existe pas deja
        //
        while(strtok.hasMoreTokens())
        {
            String mot = strtok.nextToken();

            // On recherche si'il n'existe pas deja
            boolean trouve=false;
            for(int i=0;i<nbMots;i++)
            {
                if (mot.equals(tabMots[i]))
                {
                    trouve=true;
                    break;
                }
            }
            // S'il n'existe pas on l'ajoute
        }
    }
}
```

```
        if (! trouve)
        {
            tabMots[nbMots]=mot;
            nbMots++;
        }
    }

    // On construit un nouveau tableau contenant que les mots uniques
    //
    String[] tab = new String[nbMots];
    for(int i=0;i<nbMots;i++)
        tab[i] = tabMots[i];

    return(tab);
}
}
```

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]

```
public class Probleme3
{
    static public void main(String args[])
    {
        // On stocke dans un tableau les entiers en entree du programme
        //
        int[] tab = new int[args.length];
        for(int i =0;i<args.length;i++)
        {
            try{
                // Conversion du parametre en entier
                tab[i]=Integer.parseInt(args[i]);
                if (tab[i]<0) // Si negatif alors positif
                    tab[i]=tab[i]*(-1);
            }catch(Exception ex)
            // Un des paramtere n'est pas un entier
            {
                Terminal.ecrireStringln("pas un entier :
"+args[i]);
                tab[i]=0; // On remplace la valeur par 0
            }
        }

        // On parcourt le tableau avec une valeur max courante
        //
        int max=-1; // max courant
        for(int i=0;i<tab.length;i++)
        {
```

```
        // Si la valeur courante n'est pas superieur à la valeur
        // max courante alors on revient à la ligne et
        // on affiche la valeur
        // sinon on affiche la valeur sans aller à la ligne.
        if (tab[i]<max)
        {
            Terminal.ecrireStringln("");
            Terminal.ecrireString(tab[i]+" ");
            max=tab[i];
        }
        else
        {
            Terminal.ecrireString(tab[i]+" ");
            max=tab[i];
        }
    }
}
```

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

4. Problème 4 [1h00] [8 points]

```
import java.util.*;

public class Probleme4
{
    static public void main(String args[])
    {
        // Lecture du fichier des notes de la classe
        //
        StringBuffer strbuf = Terminal.lireFichier("Classe.txt");
        String[] lignes = strbuf.toString().split("\n");

        // QUESTION 1
        // =====
        Terminal.ecrireStringln("QUESTION 1 :");
        Terminal.ecrireStringln("-----");
        // Saisie du nom et du prenom de l'eleve
        //
        Terminal.ecrireString("Nom :");
        String nom = Terminal.lireString();
        Terminal.ecrireString("Prenom :");
        String prenom = Terminal.lireString();

        // Parcours des lignes du fichier
        //
        for(String l : lignes)
        {
            // Si la ligne est celle de l'eleve alors on affiche sa
note
            String[] infos = l.split(";");
            if( nom.equals(infos[0])&&
                prenom.equals(infos[1]))
            {
                Terminal.ecrireStringln(infos[2]+":"+infos[3]);
            }
        }

        // QUESTION 2
        // =====
        Terminal.ecrireStringln("QUESTION 2 :");
        Terminal.ecrireStringln("-----");

        // Determiner la liste des matières
        //
        String[] matieres = new String[100]; // Tableau des matieres
        int nbMatiere=0; // Nombre de matiere

        // Parcours des lignes du fichier
        //
        for(String l : lignes)
        {
            // Extraction de la matière
            String[] infos = l.split(";");
            String matiere = infos[2];

            // Si la matiere n'existe pas deja alors on l'ajoute
            // dans le tableau des matieres
            //
            boolean trouve=false;
            for(int i=0;i<nbMatiere;i++)
```

```
        if (matieres[i].equals(matiere))
            trouve=true;
        if(!trouve)
        {
            matieres[nbMatiere] = matiere;
            nbMatiere++;
        }
    }

    // Pour chaque matiere on parcours les lignes du fichier
    // et on calcule la somme et le nombre de note de chaque matiere
    //
    double[] sommes    = new double[nbMatiere];
    int[]    nbMoyenne = new int[nbMatiere];

    for(int i=0;i<nbMatiere;i++)
    {
        for(String l : lignes)
        {
            String[] infos = l.split(";");
            double note = Double.parseDouble(infos[3]);
            if( matieres[i].equals(infos[2]))
            {
                sommes[i]=sommes[i]+note;
                nbMoyenne[i]=nbMoyenne[i]+1;
            }
        }
        // Calcul de la moyenne de la matiere et affichage
        //
        double moyenne = sommes[i]/nbMoyenne[i];
        Terminal.ecrireStringln(matieres[i]+":"+moyenne);
    }
}
}
```

Exemple d'une exécution :

java Probleme4

QUESTION 1 :

Nom :**Lafont**

Prenom :**Pierre**

Histoire:13.5

Geographie:12.0

Francais:8

Math:18

QUESTION 2 :

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 de la correction)