

Examen de NFA031 (S2)

26 Juin 2013

Durée : 3 heures. Documents, calculettes, portables, ordinateurs interdits. Barème indicatif.

Précision : pour les entrées/sorties, on utilise dans ces exercices la classe `Terminal`, déjà écrite, qui dispose (entre autres) des méthodes suivantes :

- `public static void ecrireChar(char c)` : écrit un caractère (`char`) ;
- `public static void ecrireString(String s)` : écrit une `String` ;
- `public static void sautDeLigne()` : saute une ligne (écrit le caractère `\n`) ;
- `public static String lireString()` : fonction qui lit et retourne une `String` tapée au clavier ;
- `public static int lireInt()` : fonction qui lit et retourne un entier tapé au clavier ;

Exercice 1 4 points

Soit le programme :

```
public class P1 {
    public static void f(char t[], int i, int j) {
        char tmp= t[i];
        t[i]= t[j];
        t[j]= tmp;
    }

    public static void a(char t[]) {
        for (int i= 0; i< t.length; i++) {
            Terminal.ecrireChar(t[i]);
        }
        Terminal.sautDeLigne();
    }

    public static void main(String args[]) {
        char t[]= {'a', 'b', 'c', 'd'};
        for (int i= 0; i < t.length - 1; i++) {
            for (int j= 1; j < t.length - i; j++) {
                f(t, j-1, j);
                a(t);
            }
        }
    }
}
```

Question 1.1 1 point

Que fait la méthode `f` ?

Correction : elle échange le contenu des cases `t[i]` et `t[i+1]`.

Question 1.2 3 points

Qu'affiche le programme ?

Correction :

```
bacde
bcade
bcdae
bcdea
cbdea
cdbea
cdeba
dceba
decba
edcba
```

Exercice 2 4 points

Écrire une fonction qui prend comme argument un tableau de double à **deux dimensions**, et qui renvoie le max des éléments de ce tableau.

Correction

- en-tête : 0,5 point pour un en-tête correct;
- les deux bornes maximales des boucles (`t.length` et `t[i].length`) valent 0,5 point chacune.
- initialisation de `m` 0,5 point ;
- 2 point pour le reste

```
public static double max(double[][] t) {
    double m= t[0][0];
    for (int i= 0; i < t.length; i++) {
        for (int j= 0; j < t[i].length; j++) {
            if (m < t[i][j]) {
                m= t[i][j];
            }
        }
    }
    return m;
}
```

Exercice 3 5 points

Question 3.1 3 points

Écrire une fonction qui renvoie le nombre de fois où la suite de caractères 'a', 'b' se rencontre dans un tableau de caractères passé en paramètre.

Par exemple, dans le tableau `{ 'a', 'b', 'c', 'a', 'a', 'b', 'a' }` on rencontre 2 fois la séquence.

Correction

```
public static int nbAB(char t[]) {
    double nb= 0;
    char prec= ' ';
    for (int i= 0; i < t.length; i++) {
        if (prec== 'a' && t[i] == 'b') nb++;
    }
}
```

```

        prec= t[i];
    }
    return nb;
}

```

Question 3.2 2 points

Écrire une fonction qui prend comme argument un tableau de char, et renvoie un booléen, vrai si et seulement si on trouve un 'a', puis un 'b' dans le tableau, sans que les case ne soient nécessairement contiguës. On renverra true pour des tableaux comme : {'a', 'b', 'c', 'a', 'a', 'b', 'a'}, {'a', 'c', 'b', 'a'} et false pour {'b', 'c', 'a', 'a'} ou {'h', 'c', 'd', 'e'}.

Correction

```

public static boolean contientAB(char t[]) {
    boolean aVu= false;
    boolean abVu= false;
    for (int i= 0; i < t.length; i++) {
        if (t[i] == 'a') aVu= true;
        else if (t[i]== 'b' && aVu) abVu= true;
    }
    return abVu;
}

```

Exercice 4 7 points

On veut écrire un logiciel de gestion de notes pour une matière. Chaque étudiant a trois notes : la note d'examen, la note de partiel, et la note finale, qui est le max de la note d'examen et de la moyenne entre partiel et examen.

On devra donc saisir deux note pour chaque étudiant, la troisième étant calculée.

Pour stocker ces informations, on utilise

- un tableau de Strings pour les noms des étudiants ;
- un tableau à deux dimensions pour les notes. Il y a une ligne par étudiant, et 3 colonnes, une par note.

Le nom qui est dans la ligne i du tableau des noms est celui de l'étudiant dont les notes seront stockées dans la ligne i du tableau des notes.

Dans le code suivant, l'utilisateur entre les noms de tous les étudiants. Ensuite, commence la phase de saisie des notes. On entre le nom de l'étudiant dont on veut saisir les notes, ou "q" pour quitter.

On demande ensuite la note d'examen et la note de partiel de l'étudiant dont on vient d'entrer le nom.

Quand on termine la saisie des notes en tapant "q", on calcule les notes finales, puis on affiche les noms des étudiants et les détails de leur notes.

```

public class E1 {

    public static void main(String args[]) {
        int nbreEtudiants;
        double notes[][];
        String noms[];
        Terminal.ecrireString("nombre d'etudiants");
        nbreEtudiants= Terminal.lireInt();
        noms= new String[nbreEtudiants];
        notes= new double[nbreEtudiants][3];
        // lire les noms d'etudiant
        Terminal.ecrireString("Entrez les noms des etudiants");
        for (int i= 0; i < nbreEtudiants; i++) {

```

```

        noms[i]= Terminal.lireString ();
    }
    String reponse;
    do {
        Terminal.ecrireString("entrez un nom d'etudiant ('q' pour terminer)");
        reponse= Terminal.lireString ();
        // On ne doit pas demander de note si la réponse est "q"
        if (! reponse.equals("q")) {
            demanderLesNote(reponse , noms , notes);
        }
    } while (! reponse.equals("q"));
    calculerNotesFinales(notes);
    afficherNotes(noms , notes);
}

public static int indiceEtudiant(String nomUnEtudiant , String [] noms) {
    COMPLÉTEZ
}

public static void demanderLesNote(String nomUnEtudiant , String [] noms , double notes [][]) {
    COMPLÉTEZ
}

public static double lireUneNote () {
    COMPLÉTEZ
}

public static void calculerNotesFinales(double notes [][]) {
    COMPLÉTEZ
}

public static void afficherNotes(String noms[], double notes [][]) {
    COMPLÉTEZ
}
}

```

Question 4.1 1 point

Écrire la méthode `lireUneNote`, qui demande un réel entre 0 et 20. Si l'utilisateur entre une valeur hors limites, vous devez la redemander, et ainsi de suite...

On ne se souciera pas de l'ergonomie de la fonction. À la limite, elle peut n'afficher aucun texte.

Correction

```

public static double lireUneNote () {
    double note;
    boolean noteIncorrecte;
    do {
        Terminal.ecrireString("Entrez une note entre 0 et 20: ");
        note = Terminal.lireDouble ();
        noteIncorrecte= (note < 0 || note > 20);
        if (noteIncorrecte) {
            Terminal.ecrireStringln("note hors limites. Recommencez");
        }
    } while (noteIncorrecte);
    return note;
}

```

Question 4.2 1 point

Écrire la méthode `indiceEtudiant`, qui cherche la position d'un étudiant dont on donne le nom dans le tableau des noms. Si l'étudiant n'est pas trouvé, retourner -1.

Correction

```
public static int indiceEtudiant(String nomUnEtudiant, String[] noms) {
    for (int i= 0; i < noms.length; i++) {
        if (nomUnEtudiant.equals(noms[i])) return i;
    }
    return -1;
}
```

Question 4.3 2 points

Écrire la méthode `demanderLesNote`, qui demande les notes d'un étudiant dont le nom est passé en paramètre.

Correction

```
public static void demanderLesNote(String nomUnEtudiant, String[] noms, double notes[][])
int pos= indiceEtudiant(nomUnEtudiant, noms); Terminal.ecrireString("Note examen de "+
nomUnEtudiant); double n1= lireUneNote(); Terminal.ecrireString("Note partiel "); double
n2= lireUneNote(); notes[pos][0]= n1 ; notes[pos][1]= n2 ;
```

Question 4.4 1.5 points

écrire la méthode `calculerNotesFinales` qui calcule les notes finales de tous les étudiants ;

Correction

```
public static void calculerNotesFinales(double notes[][]) {
    for (int i= 0; i < notes.length; i++) {
        double moy= (notes[i][0]+ notes[i][1])/2.0;
        notes[i][2]= Math.max(moy, notes[i][0]);
        // ou calcul à la main du max, évidemment...
    }
}
```

Question 4.5 1.5 points

Écrire la méthode `afficherNotes`, qui, pour chaque étudiant, affiche, sur une ligne, son nom et ses trois notes.

Correction

```
public static void afficherNotes(String noms[], double notes[][]) {
    for (int i= 0; i < notes.length; i++) {
        Terminal.ecrireString(noms[i]+ " :");
        for (int j=0; j < 3; j++) {
            Terminal.ecrireString(notes[i][j]+ " ");
        }
        Terminal.sautDeLigne();
    }
}
```