

Examen de NFA031 (S2)

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 ecrireStringln(String s)` : écrit une `String` et saute à la ligne ;
- `public static void ecrireIntln(int i)` : écrit un entier et saute à la ligne ;
- `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 ;
- `public static int lireDouble()` : fonction qui lit et retourne un double tapé au clavier ;

Exercice 1 4 points

Soit le programme :

```
public class P2 {
    public static void main(String args[]) {
        int t1[] = {7,1,6,1,9,15};
        int t2[] = {1,3,6,20,9};
        for (int i= 0; i < t1.length; i++) {
            int j= 0;
            while (j < t2.length && t1[i] > t2[j]) {
                j++;
            }
            if (j < t2.length && t1[i] == t2[j]) {
                Terminal.ecrireIntln(t1[i]);
            }
        }
    }
}
```

Qu'affiche le programme ?

Correction

1
6
1

Exercice 2 4 points

Écrire un programme qui demande 100 entiers, et affiche “gagné” si l'utilisateur a saisi deux valeurs successives identiques (et “perdu” sinon).

Correction : une solution parmi d'autre. On peut aussi envisager d'écrire gagné et perdu à chaque fois...

```
public class Cent {
    public static void main(String a[]) {
        boolean gagne= false;
        int precedent= Terminal.lireInt(); // premier entier...
        for (int i= 0; i < 99; i++) { // 99 suivants
            int c= Terminal.lireInt();
            if (c== precedent) {
                gagne= true;
            }
            precedent= c;
        }
        if (gagne) {
            Terminal.ecrireString("gagné");
        } else {
            Terminal.ecrireString("perdu");
        }
    }
}
```

Exercice 3 5 points

Question 3.1 2 points

Écrire la fonction `nombreDOccurrence`, qui prend comme arguments un tableau de char `t` et un char `a`, et renvoie le nombre de fois où on a trouvé `a` dans `t`.

Correction :

```
public static int nombreDOccurrence(char t[], char a) {
    int cpt= 0;
    for (int i= 0; i < t.length; i++) {
        if (t[i]== a) cpt++;
    }
    return cpt;
}
```

Question 3.2 3 points

Écrire la fonction `caractereLePlusFrequent`, qui prend comme argument un tableau de char `t` et renvoie le caractère qui se rencontre le plus fréquemment dans `t`. Dans le cas où plusieurs caractères sont possibles, vous renverrez le premier que l'on rencontre dans le tableau.

Exemple : `{ 'c', 'a', 'd', 'a', 'b', 'e', 'a', 'b', 'a' }` : le caractère le plus fréquent dans ce tableau est 'a' (4 occurrences).

Exemple : `{ 'c', 'f', 'f', 'd', 'c' }` : on rencontre 'c' et 'f' deux fois chacun. On renvoie donc 'c', qui se trouve en premier dans le tableau.

Correction :

```

public static char caractereLePlusFrequent(char t[]) {
    int meilleurNb= 0;
    char result= ' ';
    for (int i= 0;i< t.length; i++) {
        int nb= nombreDOccurrence(t, t[i]);
        if (nb > meilleurNb) {
            meilleurNb= nb;
            result= t[i];
        }
    }
    return result;
}

```

Exercice 4 7 points

Dans une école, les étudiants passent un certain nombre de modules. Ils ont une note par module, et obtiennent le diplôme si :

- ils ont au moins 10 de moyenne générale (les modules ont tous le même coefficient);
- ils ont au moins 8 dans chaque module (une note strictement inférieure à 8 est éliminatoire).

Les noms des étudiants sont représentés par un tableau de Strings. Les notes par un tableau à deux dimensions, notes[][].

notes[i][j] représente la note numéro j de l'étudiant numéro i; le nom de celui-ci est nom[i].

```

public class E2 {

    public static void main(String args[]) {
        int nbreEtudiants;
        double notes[][];
        String noms[];
        noms= saisirNoms();
        notes= saisirNotes(noms);
        Terminal.ecrireString("entrez le nom d'un des etudiants");
        String etud= Terminal.lireString();
        if (passe(etud, noms, notes)) {
            Terminal.ecrireStringln(etud + " a eu son diplome");
        } else {
            Terminal.ecrireStringln(etud + " n'a pas eu son diplome");
        }
    }

    public static String [] saisirNoms(){
        // COMPLETER.
        return null;
    }

    public static double [][] saisirNotes(String noms[]){
        // COMPLETER.
        return null;
    }

    public static boolean passe(String nomEtudiant, String noms[], double notes[][]){
        // COMPLETER.
        return false;
    }

    public static int indiceEtudiant(String nomUnEtudiant, String [] noms) {
        // COMPLETER
        return -1;
    }
}

```

```
}
```

Question 4.1 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.

Question 4.2 1 point

Écrire la méthode `saisirNom`, qui demande combien d'étudiants il y a, saisit et retourne leurs noms.

Question 4.3 2 point

Écrire la méthode `saisirNotes`, qui demande combien de modules il y a, puis saisit et retourne les notes de tous les étudiants dans ces modules.

Question 4.4 3 point

Écrire la méthode `passé`, qui permet de savoir si un étudiant dont on donne le nom a obtenu le diplôme (c'est-à-dire a 10 de moyenne sur l'ensemble des modules, et 8 ou plus à chaque module).

Correction :

```
public class E2 {

    public static void main(String args[]) {
        int nbreEtudiants;
        double notes [][];
        String noms[];
        noms= saisirNoms ();
        notes= saisirNotes(noms);
        Terminal.ecrireString("entrez le nom d'un des etudiants");
        String etud= Terminal.lireString ();
        if (passé(etud, noms, notes)) {
            Terminal.ecrireStringln(etud + " a eu son diplome");
        } else {
            Terminal.ecrireStringln(etud + " n'a pas eu son diplome");
        }
    }

    public static String [] saisirNoms(){
        Terminal.ecrireString("nombre de noms:");
        int nb= Terminal.lireInt ();
        String result[]= new String[nb];
        for (int i= 0; i < result.length; i++) {
            result[i]= Terminal.lireString ();
        }
        return result;
    }

    public static double [][] saisirNotes(String noms[]){
        double [][] notes;
        Terminal.ecrireString("nombre de modules:");
        int nbMod= Terminal.lireInt ();
        notes= new double[noms.length][nbMod];
        for (int i= 0; i < noms.length; i++) {
            for (int j= 0; j < nbMod; j++) {
                Terminal.ecrireString("note de " + noms[i] +
```

```

        "dans le module num. " + j+ ":"");
    notes[i][j]= Terminal.lireDouble();
    }
}
return notes;
}

public static boolean passe(String nomEtudiant, String noms[], double notes[][] ) {
    int pos= indiceEtudiant(nomEtudiant, noms);
    boolean sup8= true;
    double moyenne= 0;
    for (int i= 0; i < notes[pos].length; i++) {
        double v= notes[pos][i];
        if (v < 8) sup8= false;
        moyenne+= v;
    }
    moyenne= moyenne/ notes[pos].length;
    return sup8 && (moyenne >= 10.0);
}

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;
}
}
}

```