

Année universitaire 2015 – 2016

## **Sujet UE NFA031: Programmation avec Java : notions de base**

Examen première session : 2/02/2017

Responsable : François Barthélemy

Durée : 3 heures

### **Consignes**

Aucun document n'est autorisé.

Calculatrice non autorisée

Les téléphones mobiles et autres équipements communicants (exemple : PC, tablettes, etc) doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve.

Sujet de 4 pages, celle-ci comprise.

Le barème est donné à titre purement indicatif. Il est susceptible de changer

→ *Vérifiez que vous disposez de la totalité des pages du sujet en début d'épreuve et signalez tout problème de reprographie le cas échéant.*

## Exercice 1 : exécution de programme (3 points)

Retracez l'exécution du programme suivant en indiquant l'évolution des valeurs des variables qu'il manipule. Vous pourrez utiliser un tableau comportant une colonne pour chaque variable, en précisant sur chaque ligne du tableau le numéro de ligne de code exécutée. *Nota bene* : Vous pouvez vous limiter aux instructions qui réalisent des affectations et aux conditions de la boucle.

```
1   int res=-1;
2   int [] tab={1,2,3,6,10};
3   boolean fin= false;
4   int i=0;
5   while(i<tab.length-1 && !(fin)){
6       if (tab[i+1]-tab[i]!=1){
7           res=i;
8           fin=true;
9       }
10      i=i+1;
11  }
12  System.out.println(res);
```

## Exercice 2 : programmes et boucles (3 points)

### Question 1

Ecrivez un programme qui saisit au clavier un nombre entier et qui affiche sur des lignes séparées tous les diviseurs exacts de ce nombre, excepté lui-même. Voici un exemple d'exécution :

```
Entrez le nombre: 15
1
3
5
```

### Question 2

Ecrivez un programme comportant deux boucles imbriquées qui affiche la chose suivante à l'écran :

```
0X
1 X
2  X
3  X
4   X
5    X
6     X
7      X
8       X
9        X
```

### Exercice 3 : programme avec des méthodes (3.5 points)

Pour afficher agréablement des opérations arithmétiques, il faut que les nombres soient alignés à droite, alors qu'un affichage basique avec `Terminal.ecrireInt` ou `System.out.println` les aligne à gauche.

Affichage courant	Affichage souhaité
125	125
+ 988	+ 988
—	—
1013	1013

#### Méthode

Ecrire une méthode qui prend en paramètre un int et renvoie une chaîne de caractères de longueur 8 contenant d'abord des espaces, puis le nombre passé en paramètre. On vous suggère de convertir le nombre en chaîne avec une concaténation de la forme " "+x. Vous pouvez connaître la longueur de cette chaîne avec la méthode `length` puis compléter la chaîne en ajoutant (en début de chaîne) le nombre d'espaces pour atteindre la longueur 8. Ex : si le nombre est 125, la chaîne renvoyée sera " 125" avec 5 espaces en début de chaîne.

#### Programme

Ecrivez un programme qui lit deux nombres au clavier et affiche l'addition de ces deux nombres avec un alignement correct, en utilisant la méthode écrite à la question précédente.

### Exercice 4 : programme avec tableau (4 points)

1. Ecrivez un bout de programme qui crée un tableau de 10 notes, lit les 10 notes au clavier et affiche le tableau.
2. Ecrivez un **autre** bout de programme que l'on peut mettre à la suite du premier et qui vérifie si toutes les notes du tableau sont comprises entre 0 et 20. **Il ne faut pas faire ce contrôle au moment de la lecture au clavier. Le contrôle se fera après avoir totalement rempli le tableau avec les notes lues (pas 1)**. Le résultat de la vérification doit être affiché (un seul message donnant le résultat pour la totalité du tableau et non pas un message par note).
3. Ecrivez un troisième bout de programme que l'on peut ajouter aux deux premiers et qui calcule la moyenne des 10 notes. Le résultat pourra être entier ou à virgule.

### Exercice 5 : programme avec méthodes et tableaux (6,5 points)

On va représenter le contenu d'un placard à provisions par deux tableaux : dans l'un on enregistre les noms (Strings) des provisions et dans l'autre, la quantité disponible (en grammes) pour cette provision. Les deux tableaux seront en relation au moyen de leurs indices. Pour un numéro de case n, on trouve le nom d'une provision (p.e : "farine") dans la case n du premier tableau et la quantité disponible de cette provision dans la case n du second tableau.

*Nota bene* : les méthodes demandées doivent prendre en paramètre toutes les données nécessaires à leur réalisation. Elles ne doivent réaliser ni lecture ni affichage. Elles doivent fonctionner quelque soit le contenu des tableaux passés en paramètre.

### Question 1

Ecrivez une méthode `affichePlacard(String[] tp, int [] tq)` permettant d'afficher les informations contenues dans le placard, provision par provision (pour chaque provision, son nom et la quantité disponible).

### Question 2

Ecrivez la méthode `indiceDe(String ingre, String[] tp)` qui prend en paramètre un nom d'ingrédient et qui renvoie son indice dans le placard s'il existe, et -1 sinon. *Nota bene* : pour comparer des chaînes de caractères vous pourrez utiliser la méthode `equals`. Si `s1` et `s2` sont des variables de type `String`, l'appel `s1.equals(s2)` renvoie `true` si ces deux chaînes sont identiques.

### Question 3

Ecrivez une méthode `obtenirQuant(String p, String[] tp, int [] tq)` qui retourne la quantité de cet ingrédient restant dans le placard. Si l'ingrédient n'existe pas dans le placard, cette méthode doit renvoyer un nombre négatif. Pour écrire cette méthode, vous devrez impérativement utiliser un appel adéquat de la méthode `indiceDe`.

### Question 4

Ecrivez une méthode `prelever(String p, int quant, String[] tp, int [] tq)` qui prélève la quantité `q` dans l'ingrédient `p` du placard. Si la quantité disponible est insuffisante cette méthode devra lever l'exception `RuntimeException`.