le cnam

Année universitaire 2015 – 2016

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

Examen première session : 04/02/2016

Responsable : Virginia APONTE

Durée : 3 heures

Aucun document autorisé. Tout support électronique est interdit : pas d'ordinateur, de tablette, de liseuse...

**Les téléphones mobiles et autres équipements communicants doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve.**

Sujet de 6 pages, celle-ci comprise.

*Le barême est donné à titre indicatif ; il est susceptible de modifications.*

*→ 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 : QCM (2,5 points)

**Attention : les bonnes réponses rapportent des points mais les réponses fausses sont pénalisées par le retrait de points.** Il y a une seule bonne réponse par question.

### Question 1

```
double m = 4;
int x = m+1;
```

- 1a. ce code produit une erreur de compilation à la ligne 1.
- 1b. ce code ne produit pas d'erreur de compilation, et x vaut. 5
- 1c. ce code produit une erreur de compilation à la ligne 2.

### Question 2

```
double r = 5/2;
```

- 2a. la valeur de r est 2.0.
- 2b. ce code produit une erreur de compilation.

### Question 3

Pour tester si un tableau est trié en ordre strictement croissant on propose la solution suivante. Le résultat de la vérification est stocké dans la variable `ordre`.

```
int[] tab ={....};
boolean ordre = true;
for (int i=0; i<tab.length; i++){
   if (tab[i]>=tab[i+1]){
      ordre = false;
   }
}
```

Quel énoncé convient pour décrire le comportement de ce code :

- 3a. ce code ne produit pas d'erreur, et determine bien si le tableau est trié en ordre strictement croissant.
- 3b. ce code produit une erreur à l'exécution si le tableau n'est pas trié.
- 3c. ce code produit une erreur à l'exécution si le tableau est trié.
- 3d. ce code ne produit pas d'erreur mais ne sert pas à determiner si le tableau est en ordre croissant.

### Question 4

```
static void m(int[] t){
   t[0] = 100;
}
static void r(){
   int[] tab = {1,2,3};
```

```
    m(tab);
    // point analyse
}
```

Lors de l'exécution de ce code, que peut-on dire au point d'analyse signalé par un commentaire ?

- 4a. le tableau `tab` contient {`1,2,3`}.
- 4b. le tableau `tab` contient {`100,2,3`}.
- 4c. il y a une erreur car l'appel à la méthode m doit se faire avec un tableau de nom `t` et non `tab`.

### Question 5

On veut écrire une méthode `insertion` afin d'ajouter dans un tableau t une nouvelle case contenant un entier x. Le tableau résultat contient tous les éléments de t ainsi qu'une case de plus avec la valeur x. Voici deux entêtes possibles pour cette méthode :

- choix 1 : `static void insertion(int x, int[] t)`
- choix 2 : `static int[] insertion(int x, int[] t)`

- 5a. seul le choix 1 convient
- 5b. seul le choix 2 convient
- 5c. aucun des deux ne convient.
- 5d. les deux choix conviennent.

## Exercice 2 : 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  public class Exo2 {
2      public static void main(String[] args) {
3          int n = 42;
4          int bt = 0;
5          String r="";
6          while (n>0){
7              bt= n%2;
8              n=n/2;
9              r=bt+r;
10         }
11         Terminal.ecrireStringln("Valeur de r: "+ r);
12         Terminal.ecrireStringln("Valeur de n: "+ n);
13         Terminal.ecrireStringln("Valeur de bt: "+ bt);
14     }
15 }
```

On rappelle que `a%b` correspond à l'opération *modulo*, qui calcule le reste de la division entière de a par b. Exemple : la division entière 11/3 donne 3 en résultat (et non pas un nombre à virgule, puisqu'il s'agit d'une division entière). Le reste de 11/3 est 2. Donc `11 % 3 == 2`. Notez que les divisions effectuées par ce code **sont des divisions entières**.

## Exercice 3 : programmes et boucles (4,5 points)

### Question 1

Ecrivez un programme qui saisit au clavier une chaîne de caractère et affiche sur la première ligne cette chaîne de caractère, sur la deuxième ligne, deux fois la chaîne, sur la nième ligne n fois la chaîne entrée, et ainsi de suite jusqu'à ce que la longueur de la ligne soit supérieure à 40. La méthode `length()` renvoie la longueur d'une chaîne de caractère.

Voici un exemple d'exécution :

```
> java NFois
Entrez une chaîne: nfa031
nfa031
nfa031nfa031
nfa031nfa031nfa031
nfa031nfa031nfa031nfa031
nfa031nfa031nfa031nfa031nfa031
nfa031nfa031nfa031nfa031nfa031nfa031
```

### Question 2

Ecrivez un programme qui affiche un carré de 10 lignes et 10 colonnes où chaque case contient le numéro de la ligne courante.

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

### Question 3

Ecrivez un programme avec une ou deux boucles qui affiche le triangle suivante :

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7
```

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

Un logiciel de gestion de tâches permet de stocker une liste tâches à réaliser. Une tâche a un nom (String) et une échéance ou statut, donnée par un caractère : A pour une tâche à faire aujourd'hui, D si elle est à faire demain, P si elle à faire plus tard et F pour une tâche déjà faite. On veut utiliser des tableaux pour stocker les noms des tâches ainsi que les caractères correspondant à l'échéance/statut de chaque tâche. Cela peut se faire avec deux tableaux différents. Vous pouvez également choisir d'utiliser un tableau à deux dimensions de taille Nx2 où N est le nombre de tâches. Dans ce cas, la première colonne pourra stocker le nom d'une tâche, et la deuxième son échéance/statut sous forme de chaîne.

*Nota bene* : Pour chaque question vous devez produire un morceau de méthode main qui répond à la question posée. Il n'est pas démandé d'écrire des méthodes, mais cela n'est nullement interdit.

### Question 1

Donnez le code pour créer et initialiser les variables permettant de répresenter les tâches et leurs échéances/statut, avec les données suivantes. Les écheances/statut sont données entre parenthèses.

1. Analyse (F)
2. Tester projet (A)
3. Acheter 2 baguettes (A)
4. Soutenance (D)
5. Révisions examen (P)
6. Rdv Oftalmo (P)
7. Inscrire NFA032 ? (P)

### Question 2

On veut lire au clavier les tâches ayant été réalisées dernièrement. Produisez le code permettant de lire les numéros de ces tâches. Votre programme doit modifier les statuts correspondants en F.

### Question 3

Donnez le code permettant d'afficher la liste complète des tâches, avec leurs échéances/statut. Pour chaque tâche vous afficherez son rang, son nom, et entre parenthèses son statut.

### Question 4

Donnez le code permettant d'afficher un bilan de tâches réalisées et à faire. Vous devez afficher le nombre de tâches réalisées, le nombre de tâches à faire aujourd'hui, le nombre à faire demain puis le nombre de tâches à faire plus tard. Enfin, vous afficherez le taux de productivité donné par un nombre à virgule. Ce taux est calculé en multipliant le nombre de tâches réalisées par 100 et en divisant le tout par le nombre total de tâches.

## Exercice 5 : temperatures mensuelles (6 points)

On veut écrire un programme qui traite les températures d'un mois stockées dans un tableau.

1. écrivez une méthode qui calcule la moyenne des températures d'un mois.
2. écrivez une méthode qui crée et renvoie un nouveau tableau de températures. Le nombre de jours du mois est passé en paramètre. Le tableau est créé et renvoyé mais il ne contient pas de températures : **il ne s'agit donc pas de saisir les températures au clavier**.
3. écrivez une méthode qui **utilise la méthode précédente** pour créer un tableau de températures, mais cette fois, c'est le numéro du mois qui est passé en paramètre (1 pour janvier, 12 pour décembre). Pour cette question, on négligera le fait qu'il existe des années bissextiles et on considèrera que février compte 28 jours.
4. écrivez une méthode qui pour un mois donné, calcule le nombre de périodes d'au moins trois jours consécutifs avec des températures négatives. Par exemple, si les températures du mois sont les suivantes :

| jour | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| température | 5.1 | 2.3 | -0.2 | -0.5 | 0.6 | 1.2 | -0.5 | -1 | -2.3 | -1.8 | -2.9 | 0.5 | 1.1 | 5.1 | 4.2 |

| jour | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| température | 0.2 | -1.1 | -1.2 | -1.3 | 0.2 | 1.2 | -0.5 | 1.1 | 5.1 | 2.3 | -0.2 | -1.5 | -1.3 | -1.5 | -1.1 |

   il y a trois périodes d'au moins trois jours consécutifs avec températures négatives : du 7 au 11, du 17 au 19 et du 26 au 30. Dans un cas comme celui-ci, la méthode renvoie donc 3.

*Nota bene :* Les méthodes demandées ne doivent faire ni lecture au clavier ni affichage à l'écran. On ne demande pas de méthode main.