

NFA031 Programmation avec Java: notions de base

CNAM - centre d'enseignement de Paris

session 1 - 12 février 2015

Aucun document n'est autorisé. Le barème est donné à titre indicatif.

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

```
int x;  
x = Terminal.lireInt()*3;
```

- 1a cette ligne de code est du Java correct
- 1b cette ligne de code est incorrecte et provoque une erreur du compilateur

Question 2

Pour parcourir un tableau d'entier **tab** jusqu'à la case contenant la valeur 100, on propose deux possibilités :

- choix 1 :

```
int i=0;  
while(i<tab.length && tab[i]!=100){  
    i++;  
}
```
- choix 2 :

```
int i=0;  
boolean trouve = false;  
while(i<tab.length && !trouve){  
    if (tab[i] == 100){  
        trouve = true;  
    }else{  
        i++;  
    }  
}
```

- 2a aucun des deux choix ne marche

- 2b seul le choix 1 marche
- 2c seul le choix 2 marche
- 2d les deux choix marchent

Question 3

```
int[] tab = {...};
boolean positifs = true;
for (int i=0; i<tab.length; i++){
    if (tab[i]<0){
        positifs = false;
        -- à compléter --
    }
}
```

Le code précédent sert à vérifier que toutes les valeurs du tableau sont positives. Le résultat de la vérification est stocké dans la variable `positifs`.

- 3a il ne faut pas de cas else pour le if : il faut juste fermer le if avec une accolade fermante.
- 3b il faut compléter le code avec ce cas else :

```
    }else{
        positifs=true;
    }
```

- 3c il faut compléter le code avec ce cas else :

```
    }else{
        positifs=Terminal.lireBoolean();
    }
```

Question 4

On veut écrire une méthode qui vérifie qu'un nombre `nb` est un multiple de 10. Cette méthode renvoie un résultat de type `boolean`

- 4a il faut absolument passer `nb` en paramètre
- 4b il faut absolument lire `nb` au clavier dans la méthode
- 4c il faut passer `nb` en paramètre et faire la lecture de `nb` au clavier dans la méthode.
- 4d on peut faire l'un ou l'autre, le passer en paramètre ou le lire au clavier, mais il ne faut pas faire les deux et il est préférable de le passer en paramètre.

Question 5

Une méthode peut créer deux tableaux de types différents et les renvoyer à la méthode appelante (par exemple la méthode `main`).

- vrai
- faux

Exercice 2 : exécution de programme (3 points)

Retracez l'exécution du programme suivant en indiquant l'évolution des valeurs des variables manipulées par ce programme. Vous pourrez faire cela au moyen d'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 Exam15{
2     public static void main(String [] args){
3         int x = 0;
4         String st = "x";
5         boolean b = true;
6         for (int i=1; i<=3; i++){
7             st = st + st ;
8             if (b){
9                 x = x + 10;
10            }
11            b = !b;
12        }
13        Terminal.ecrireStringln("valeur de x: " + x);
14        Terminal.ecrireStringln("valeur de st: " + st);
15        Terminal.ecrireStringln("valeur de b: " + b);
16    }
17 }
```

On rappelle que ! correspond à la négation booléenne (*non logique*) et que `if (b)` est équivalent à `if (b == true)`.

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

Question 1

Ecrivez un programme qui fasse un petit dessin de ce genre :

```
*
**
***
****
*****
****
***
**
*
```

Dans le programme le nombre maximal d'étoiles affichées sur une ligne est entré au clavier. Le dessin ci-dessus représente le cas où ce nombre vaut 5. Vous ferez attention de placer le(s) saut(s) de ligne au(x) bon(s) endroit(s).

Question 2

Ecrivez un programme qui calcule un carré de 10 lignes et 10 colonnes où chaque case contient le numéro de colonne suivi du numéro de ligne.

```
00 10 20 30 40 50 60 70 80 90
01 11 21 31 41 51 61 71 81 91
02 12 22 32 42 52 62 72 82 92
03 13 23 33 43 53 63 73 83 93
04 14 24 34 44 54 64 74 84 94
05 15 25 35 45 55 65 75 85 95
06 16 26 36 46 56 66 76 86 96
07 17 27 37 47 57 67 77 87 97
08 18 28 38 48 58 68 78 88 98
09 19 29 39 49 59 69 79 89 99
```

Question 3

Modifiez le programme pour que le numéro de colonne soit représenté par une lettre avec A pour la première colonne.

```
A0 B0 C0 D0 E0 F0 G0 H0 I0 J0
A1 B1 C1 D1 E1 F1 G1 H1 I1 J1
A2 B2 C2 D2 E2 F2 G2 H2 I2 J2
A3 B3 C3 D3 E3 F3 G3 H3 I3 J3
A4 B4 C4 D4 E4 F4 G4 H4 I4 J4
A5 B5 C5 D5 E5 F5 G5 H5 I5 J5
A6 B6 C6 D6 E6 F6 G6 H6 I6 J6
A7 B7 C7 D7 E7 F7 G7 H7 I7 J7
A8 B8 C8 D8 E8 F8 G8 H8 I8 J8
A9 B9 C9 D9 E9 F9 G9 H9 I9 J9
```

Exercice 4 : programme et tableau (4 points)

On veut utiliser un tableau pour stocker les prénoms des enfants nés dans une maternité au cours d'une année.

Question 1

Donnez le code pour créer un tableau de 100 prénoms, lire au clavier tous les prénoms et ensuite, afficher les 100 prénoms.

Question 2

On ne connaît pas à l'avance le nombre de prénoms à stocker. On va donc créer un tableau de 1000 cases et une case inutilisée contiendra soit la valeur `null` soit un signe de ponctuation qui ne peut pas être un

prénom. Vous choisirez l'une ou l'autre possibilité librement (ce choix ne recèle aucun piège).

Donnez le code pour créer le tableau, le remplir de prénoms lus au clavier jusqu'à ce qu'il n'y ait plus de prénoms, puis affiche tous les prénoms.

Question 3

Donnez le code permettant de calculer et conserver en mémoire le nombre d'enfants portant chacun des prénoms présents dans le tableau. Ou autrement dit, le nombre d'occurrences de chaque prénom dans le tableau.

Exercice 5 : méthodes pour le scrabble (6 points)

Le scrabble est un jeu de lettres qui consiste à déposer jetons représentant des lettres pour écrire des mots sur un plateau de jeu. Dans cet exercice, nous considérerons que les jetons ont tous une lettre sans accent ni cédille. On représente les jetons d'un joueur au moyen d'un tableau de 7 caractères que l'on suppose toujours rempli avec des lettres.

- Ecrivez une méthode qui teste si les jetons d'un joueur permettent d'écrire entièrement un mot donné au choix par une chaîne de caractère ou par un tableau de caractère (c'est vous qui choisissez au moment où vous faites l'exercice).
- Chaque lettre a une certaine valeur. Les lettres fréquentes comme le E ont la valeur 1, les lettres les plus rares comme X ou K ont la valeur 10. Les autres une valeur intermédiaire entre 1 et 10. Comment peut-on stocker dans la mémoire de l'ordinateur la valeur de toutes les lettres et comment peut-on retrouver la valeur d'une lettre donnée ? Vos explications doivent être courtes et peuvent comporter du code java.
- En utilisant le procédé que vous avez décrit pour stocker les valeurs des lettres, écrivez une méthode qui renvoie la valeur d'une lettre passée en paramètre.
- Ecrivez une méthode qui calcule le score d'un mot, c'est-à-dire la somme des valeurs des lettres qui le composent. Par exemple le mot *axe* a comme score 12, qui est la somme de la valeur de a (1), de celle de x (10) et de celle de e (1).
- Ecrivez une méthode qui compte combien de lettres il manque pour écrire un mot donné avec les jetons d'un joueur.
- Ecrivez une méthode qui détermine et renvoie les lettres qui manquent pour écrire un mot donné avec les jetons d'un joueur.

Les méthodes demandées ne doivent faire ni lecture au clavier ni affichage à l'écran.