le cnam

Année universitaire 2015 – 2016

Sujet UE NFA031: Programmation avec Java : notions de base

Examen deuxième session : 21/04/2016

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 6 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 : 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

En java, on peut exécuter un programme qui a provoqué une erreur à la compilation.
- 1a. vrai
- 1b. faux

Question 2

Dans un programme java, pour savoir si un nombre entier stocké dans une variable x est un nombre pair (divisible par 2), que faut-il faire ?
- 2a. tester si x/2 vaut 0
- 2b. tester si x%2 vaut 0
- 2c. tester si x%0 vaut 2
- 2d. afficher x et demander à l'utilisateur s'il est pair

Question 3

```
public static void main(String[] args){
  double[] tab1;
  int[] tab2 = new int[5];
}
```

Quel est le nombre de tableaux dans la mémoire à la fin de l'exécution de ce programme ?
- 3a. 0
- 3b. 1
- 3c. 2
- 3d. 10

Question 4

```
String reponse;
Terminal.ecrireStringln("Etes-vous heureux?");
reponse = Terminal.lireString();
```

Pour tester avec un if si la personne a répondu oui à la question posée, il faut utiliser la condition suivante :
- 4a. `reponse="oui"`
- 4b. `reponse=="oui"`
- 4c. une autre condition

Question 5

```
public static int plusUn(int x){
  int res = x+1;
  return res;
}
public static void main(String[] args){
  int x = 2;
  int y = 3;
  x = plusUn(2);      // appel 1
  x = plusUn(x);      // appel 2
  x = plusUn(y);      // appel 3
  x = plusUn("123"); // appel 4

}
```

- 5a. seul l'appel 1 est correct
- 5b. seul l'appel 2 est correct
- 5c. seuls les appels 1 et 2 sont corrects
- 5d. seuls les appels 1, 2 et 3 sont corrects
- 5e. les appels 1, 2, 3 et 4 sont tous les quatre corrects

Exercice 2 : exécution de programme (3 points)

Question 1

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 static void main(String[] args) {
2       boolean[] tab={true, true, false, true};
3       int res=0;
4       int p=1;
5       for (int i=tab.length-1; i>=0; i--){
6          if (tab[i]){
7                  res= res+ p;
8          }
9          p=p*2;
10      }
11      System.out.println(res);;
12 }
```

Question 2

Qu'affiche ce programme ?

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

Question 1

Ecrivez un programme qui saisit au clavier un nombre entier et qui affiche sur des lignes séparées toutes les puissances de ce nombre inférieures ou égales à 100.

Voici un exemple d'exécution :

```
Entrez le nombre: 3
1
3
9
27
81
```

Question 2

Ecrivez un programme qui saisit 7 nombres à virgule au clavier, qui vérifie que tous ces nombres sont des notes comprises entre 0 et 20, et, si c'est bien le cas, affiche la moyenne des notes.

Contraintes : dans tous les cas, les 7 notes doivent être lues. Le programme ne s'arrête pas si un nombre non compris entre 0 et 20 est entré. Vous utiliserez une boucle et **vous n'utiliserez pas de tableau**.

Si en dépit de la consigne vous utilisez un tableau, vous ne serez noté que sur la moitié des points attribués à l'exercice (ce qui est toujours mieux que de ne rien faire).

Question 3

Ecrivez un programme qui produit l'affichage suivant en utilisant une ou plusieurs boucles.

```
*
=***
==******
===********
====**********
=====************
======**************
=======****************
```

Exercice 4 : programme et tableau (5 points)

On va représenter les comptes bancaires au Panama d'un honnête contribuable au moyen de deux tableaux : le contiendra les numéros des comptes, le second tableau contiendra les soldes des comptes. Les deux tableaux seront coordonnés au moyen de leurs indices. Pour un numéro de case n, on trouve le numéro d'un compte dans la case n du premier tableau et le solde de ce même compte dans la case n du second tableau.

Nota bene : il n'est pas demandé d'écrire de méthode. Vous pouvez écrire une suite d'instruction à mettre dans la méthode main ou une méthode pour répondre à chaque question.

Question 1

Ecrivez les déclarations et instructions permettant de représenter trois comptes avec deux tableaux :
- le compte numéro 501 avec 10 000 dollars
- le compte 87878 avec 5 000,50 dollars
- le compte 88888 avec 100 000 dollars

Question 2

Ecrivez le code permettant d'afficher les informations contenues dans les deux tableaux compte par compte (pour chaque compte, son numéro et son solde). Ce code doit fonctionner quel que soit le nombre de comptes et pas seulement pour l'exemple donné à la question 1.

Question 3

Ecrivez le code permettant de transférer de l'argent d'un compte à un autre. Les entrées à prendre en considération sont les numéros des deux comptes et le montant à transférer. Vous gérerez les erreurs en levant une ou plusieurs exceptions.

Question 4

Ecrivez le code permettant d'ajouter un nouveau compte dans le programme.

Question 5

Ecrivez le code permettant de retirer un compte présent dans le programme.

Exercice 5 : tableaux à places vides (5 points)

On veut écrire des méthodes permettant de manipuler des tableaux d'entiers où l'on peut ajouter et enlever des éléments sans avoir à re dimensionner le tableau, ni avoir à ré-arranger ses élements. Dans les tableaux d'entiers que l'on considère, les valeurs que l'on pourra ajouter sont toujours positives ou nulles. Par exemple, on peut avoir au départ un tableau `t= {1,0,2,0,2,45,2,9}`. Si par exemple, on supprime toutes les occurrences de la valeur 2 dans ce tableau, on mettra -1 dans les cases où se trouve cette valeur. Par la suite, ces cases seront considérées comme étant vides. Aprsè suppresion on obtient le nouveau tableau : `t= {1,0,-1,0,-1,45,-1,9}`. Si par la suite on souhaite ajouter une élément, il suffira de chercher une place vide, c.a.d., contenant une valeur négative, pour y insérer le nouvel élément. Écrivez les méthodes suivantes.

1. écrivez une méthode `nbOccupees` qui calcule le nombre de places non vides d'un tableau.
2. écrivez une méthode `elimTous` qui élimine d'un tableau toutes les occurrences d'un élément passé en paramètre.
3. écrivez une méthode `ajout` qui ajoute dans un tableau un élément passé en paramètre. Votre méthode doit retourner un booléan qui indique si l'ajout à pu être effectué.
4. écrivez une méthode `elementsAGauche` qui ré-arrange les éléments d'un tableau de sorte que toutes les cases non vides se trouvent à gauche des cases vides.

5. écrivez une méthode `compacter` qui retourne un nouveau tableau contenant uniquement les cases non vides du tableau passé en paramètre.

Nota bene : Les méthodes demandées ne doivent faire ni lecture au clavier ni affichage à l'écran. Toutes ces méthodes prennent au minimum un tableau en paramètre. On ne vous demande pas d'écrire de méthode main.