

Partiel de CI2 (Concepts Informatiques)
Licence Informatique - Première année

Samedi 16 Mars 2013

Durée : 3 heures

Aucun document ou support autre que le sujet ou les copies d'examen n'est autorisé - La copie ou les brouillons du voisin ne sont pas des supports autorisés.

Positionnez impérativement vos mobiles en mode « avion ».

1 Variables, Appels et Passage d'arguments

Argh.java

```
1 import java.util.*;
2 public class Argh {
3     public static void affiche(int []t) {
4         System.out.print(t+"[");
5         for (int i=0; i<t.length-1; i++)
6             System.out.print(t[i]+",");
7         System.out.println(t[t.length-1]+"]");
8     }
9     public static void affiche(int [][]t) {
10        System.out.println("-----"+t);
11        for (int i=0; i<t.length; i++) affiche(t[i]);
12    }
13    public static void affiche(int [] [] []t) {
14        System.out.println(">>>>>"+t);
15        for (int i=0; i<t.length; i++) affiche(t[i]);
16    }
17    public static boolean fonction1(int [][]t) {
18        if (t==null) return false;
19        for (int i=0; i<t.length; i++)
20            if (t.length!=t[i].length) return false;
21        return true;
22    }
```

```

23 public static boolean fonction1(int [][] t) {
24     if (t==null) return false;
25     for (int i=0; i<t.length; i++)
26         if (t.length!=t[i].length) return false;
27     return true;
28 }
29
30 public static void fonction(int [][] t) {
31     if (t==null || !fonction1(t)) return;
32     int D = t.length;
33     for (int l=0; l<t.length/2; l++) {
34         for (int c=1; c<D-1-1; c++) {
35             int tmp = t[l][c];
36             t[l][c] = t[c][D-1-1];
37             t[c][D-1-1] = t[D-1-1][D-1-c];
38             t[D-1-1][D-1-c] = t[D-1-c][l];
39             t[D-1-c][l] = tmp;
40         }
41     }
42 }
43
44 public static void fonction(int [] [] [] t) {
45     if (t==null || !fonction1(t)) return;
46     int D = t.length;
47     for (int l=0; l<t.length/2; l++) {
48         for (int c=1; c<D-1-1; c++) {
49             int tmp = t[l][c];
50             t[l][c] = t[c][D-1-1];
51             t[c][D-1-1] = t[D-1-1][D-1-c];
52             t[D-1-1][D-1-c] = t[D-1-c][l];
53             t[D-1-c][l] = tmp;
54         }
55     }
56 }
57
58 public static void main(String args[]) {
59     Scanner in = new Scanner(System.in);
60     int [] var1 = new int[in.nextInt()];
61     for (int i=0; i<var1.length; i++) var1[i] = in.nextInt();
62     System.out.println("var1="+var1);
63     affiche(var1);
64
65     int [] [] var2 = new int[2][2];
66     int valeur = 0;
67     for (int i=0; i<var2.length; i++)
68         for (int j=0; j<var2[i].length; j++)
69             var2[i][j] = ++valeur;
70     affiche(var2);
71     fonction(var2);
72     affiche(var2);

```

```

73 int [][][] var3 = new int[2][2][3];
74 valeur = 0;
75 for (int i=0; i<var3.length; i++)
76     for (int j=0; j<var3[i].length; j++)
77         for (int k=0; k<var3[i][j].length; k++)
78             var3[i][j][k] = ++valeur;
79 affiche(var3);
80 fonction(var3);
81 affiche(var3);
82 }
83 }

```

1. Quelles sont les variables présentes dans le programme? Précisez leur nature exacte (type primitif ou référence vers quel type, etc), la ligne où elle sont définies. La liste devra être exhaustive.
2. Combien de fonctions le programme contient-il? Pour chaque fonction, précisez le type de ses arguments (en utilisant une phrase du genre «l'argument x est une variable de type nombre flottant», nous ne voulons pas ici de définition Java).
3. La compilation de ce programme indique une erreur en ligne 49. Sauriez-vous la corriger? Dans les questions suivantes, on suppose que l'erreur a été corrigée.
4. Que produit à l'affichage l'appel de la ligne 63, si l'utilisateur a saisi au clavier la ligne 2 3 4 après le lancement du programme?
5. Que produit à l'affichage l'appel de la ligne 70 (vous devez faire des suppositions sur les adresses correspondant aux variables)?
6. Que produit à l'affichage l'appel de la ligne 72 (vous devez faire des suppositions sur les adresses correspondant aux variables)?
7. Que réalise donc la fonction dont le code commence en ligne 30?
8. Que produit à l'affichage l'appel de la ligne 79 (vous devez faire des suppositions sur les adresses correspondant aux variables)?
9. Que produit à l'affichage l'appel de la ligne 81 (vous devez faire des suppositions sur les adresses correspondant aux variables)?
10. Que réalise donc la fonction dont le code commence en ligne 44?

2 Traduction de programme

Dans cet exercice, il s'agit de traduire un programme Java en un autre programme Java de forme particulière étudiée en cours : le programme principal ne devra pas contenir d'autres variables qu'un entier de nom `instructionCourante` (ou un symbole plus « court » `ic` conviendra!), un tableau d'entiers de nom `memoire` et accessoirement une pile (dont il faudra détailler les caractéristiques si vous en utilisez une, on rappelle qu'en cours nous avons codé la pile des appels directement dans la mémoire!). Le code sera simplement constitué d'une boucle `while` contenant un unique `switch` avec autant de cas que nécessaires. Dans le programme traduit, il est formellement interdit d'employer toute autre construction de haut-niveau : pas de boucle `for`, ni d'autre boucle `while`, ni d'autre `switch`, etc.

On impose à ce qu'un appel de fonction utilise la pile de la façon suivante (comme il a été fait en cours) :

- l'appel s'occupe d'empiler ce qui sera nécessaire au bon fonctionnement du code appelé;

- sont empilés dans l'ordre : la valeur de retour (s'il y en aura une), l'adresse de retour, les paramètres dans l'ordre de leur déclaration;
- la fonction appelée s'occupe de dépiler tout ce dont on aura plus besoin au retour, c'est-à-dire, tout **sauf** la valeur de retour;
- c'est le code appelant qui s'occupe de dépiler la mémoire associée à la valeur de retour.

P.java

```

1 import java.util.*;
2
3 public class P {
4     public static int i;
5     public static int j;
6     public static int []t = new int[10];
7     public static Scanner in = new Scanner(System.in);
8
9     public static void affiche(int i) {
10        System.out.println("J'affiche la valeur "+i);
11    }
12    public static int f(int i,int j) {
13        return i+j;
14    }
15    public static void main(String []args) {
16        i = in.nextInt();
17        affiche(i);
18
19        for (i=0; i<t.length; i++) {
20            j = f(2*i,1);
21            t[i] = j;
22        }
23
24        for (i=0; i<t.length; i++) affiche(t[i]);
25    }
26 }

```

1. Quelles sont les variables présentes dans le programme? Précisez leur nature (type primitif ou non), la ligne où elles sont définies, leur taille (exprimée en nombre d'entiers), le type de mémoire (statique, pile, tas) dans lequel elles seront stockées.
2. Quelle est la taille minimum (exprimée en nombre d'entiers) nécessaire pour que ce programme puisse être exécuté? On ne compte ici que les variables (pas le code). En déduire la définition de la variable de nom **memoire** qui sera utilisée pour simuler la mémoire de l'ordinateur qui exécutera le programme après sa traduction.
3. Traduire tel que vu en cours, le code correspondant à la fonction **affiche**. Cette traduction aura pour numéro de première instruction (**case**) 100.
4. Traduire tel que vu en cours, le code correspondant à la fonction **f**. Cette traduction aura pour numéro de première instruction (**case**) 200.
5. Traduire, le code principal **main**. Cette traduction aura comme numéro de première instruction (**case**) 1.