

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

Mardi 14 Mai 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 ». Attention : le sujet fait 4 pages

## 1 Variables, Appels et Passage d'arguments

Ex1.java

```
1 class Bidule {
2     public int machin;
3     public Bidule(int m) {
4         machin = m;
5     } }
6
7 public class Ex1 {
8     public static void f(Bidule b) {
9         b.machin = b.machin+10;
10    }
11    public static int f(int b) {
12        b = b+10;
13        return b;
14    }
15    public static int [] f(int []b) {
16        b = new int[1];
17        b[0] = b[0]+10;
18        return b;
19    }
20    public static void main(String []args) {
21        Bidule b1 = new Bidule(1);
22        System.out.println(b1);
```

```

23     f(b1);
24     System.out.println(b1);
25
26     Bidule b2 = new Bidule(2);
27     System.out.println(b2.machin);
28     f(b2);
29     System.out.println(b2.machin);
30     f(b2.machin);
31     System.out.println(b2.machin);
32
33     int i = 3;
34     System.out.println(i);
35     f(i);
36     System.out.println(i);
37
38     int []t1 = new int[1];
39     t1[0] = 4;
40     System.out.println(t1+" "+t1[0]);
41     f(t1);
42     System.out.println(t1+" "+t1[0]);
43 } }

```

1. Quelles sont les lignes de la classe **Ex1** où sont déclarées des variables ?
2. Quelles sont les fonctions déclarées dans la classe **Ex1** ?
3. Quels sont les affichages produits à l'exécution du programme ?
4. Si l'on remplace la ligne 41 par `t1 = f(t1);`, cela change t-il l'affichage correspondant ? Si oui comment ? Si non pourquoi ?

## 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.

## Ex2.java

```
1 import java.util.*;
2
3 public class Ex2 {
4     public static int v;
5     public static int []t = new int[10];
6     public static Scanner in = new Scanner(System.in);
7
8     public static int f(int n,int i) {
9         return (n%5)*i;
10    }
11    public static void main(String []args) {
12        for (v=0; v<t.length; v++) {
13            t[v] = f(v,in.nextInt());
14        }
15
16        for (v=0; v<t.length; v++)
17            System.out.println("t["+v+"]="+t[v]);
18    }
19 }
```

1. Si l'utilisateur tape 1 2 3 4 5 6 7 8 9 10 au clavier, quels sont les affichages produits à l'exécution ?
2. Quelle sera la taille minimale de la mémoire qui devra être employée dans le programme traduit ? Précisez l'unité.
3. Précisez où les variables du programme d'origine seront stockées dans le programme traduit ?
4. Traduisez le programme comme étudié en cours ?

### 3 Réursion

Soit la fonction de Zibonacci définie sur les entiers naturels par :

$$Z(n) = \begin{cases} 1, & \text{si } n \leq 2, \\ Z(n-3) + Z(n-1), & \text{sinon.} \end{cases}$$

1. Combien vaut  $Z(6)$  ?
2. Écrivez un programme Java de nom `Zibonacci.java` qui contient une fonction de nom `zibonacci` qui calcule **récurivement et naïvement** la fonction  $Z$ .
3. Dessinez l'arbre des appels lors du calcul de `zibonacci(6)`.
4. Utilisez la technique de la fonction à mémoire afin d'économiser les appels visiblement redondants lors du calcul naïf et écrivez le programme correspondant.

suite...

## 4 Récursion, Récursion terminale et Élimination de la récursion terminale

Soit le programme suivant :

Ex4.java

```
1 public class Ex4 {  
2     public static int f(int n) {  
3         if (n==0) return 1;  
4         if (n%2==0) return 2*f(n-2);  
5         else return 3*f(n-1);  
6     }  
7     public static void main(String []args) {  
8         for (int i=0; i<10; i++) {  
9             System.out.println("f("+i+")="+f(i));  
10        }  
11    }  
12 }
```

1. Quelle est la valeur de  $f(9)$  ?
2. La fonction  $f$  n'est pas récursive terminale, pourquoi ?
3. Transformez la fonction  $f$  afin de la rendre récursive terminale.
4. Transformez la version récursive terminale en une version itérative.