

Programmation bas niveau - L2 informatique

Deuxième Contrôle Continu Ecrit - Correction

Vendredi 6 décembre 2013 (1 heure – aucun document autorisé)

Exercice 1 : Analyse de code (5 points)

Trouver, corriger et expliquer chacune des 5 erreurs du programme ci-dessous.

```
01 .class public Factorielle
02 .super java/lang/Object
03
04 .method public static facto_recuratif(I)I
05   .limit stack 3
06   .limit locals 1 ; (point bonus)
07   iload 0
08   ifgt facto_appel_recuratif ; le label s'intitule appel_recuratif
09   iconst_1
10   goto end
11 appel_recuratif:
12   aload 0 iload 0 ; la case 0 des vars locales contient un entier
13   dup
14   iconst_1
15   isub
16   invokestatic Factorielle/facto_recuratif(I)I
17   imul
18 end:
19   return ireturn ; la méthode renvoie un entier
20 .end method
21
22 .method public static main([Ljava/lang/String;)V
23   .limit stack 2
24
25   getstatic java/lang/System/out Ljava/io/PrintStream;
26   bipush 5
27   invokestatic Factorielle/facto_recuratif(5I)I ; 2 erreurs dans la
28 signature de la méthode appelée : le nom de la classe manquant et
29 l'argument : on donne le type uniquement (I), l'argument est déjà
30 placé sur la pile (c'est 5).
31   invokevirtual java/io/PrintStream/println(I)V
32
end:
return
.end method
```

Exercice 2 : Programmation (5 points)

Écrire un programme Jasmin effectuant la somme de tous les multiples de 4 compris entre 0 et 200, puis affichant le résultat.

```
.class public Multiple4
.super java/lang/Object

.method public static main([Ljava/lang/String;)V
  .limit stack 2
  .limit locals 4
  iconst_0
  dup
  istore 0 ; le compteur va de 4 en 4
  istore 1 ; le résultat : somme des multiples de 4 entre 0 et 200
start:
  iload 0
  sipush 200
  if_icmpgt end ; on vérifie qu'on a pas dépassé 200
  iload 0
  iload 1
  iadd
  istore 1
  iinc 0 4 ; on passe au multiple de 4 suivant
  goto start
end:
  getstatic java/lang/System/out Ljava/io/PrintStream;
  iload 1
  invokevirtual java/io/PrintStream/println(I)V
  return
.end method
```

Exercice 3 : Programmation (10 points)

Ecrire un programme Jasmin qui crée un tableau de 10 entiers, remplit ce tableau avec des chiffres compris entre 0 et 9 générés de manière aléatoire, puis affiche chacun des éléments du tableau ainsi rempli. Le programme devra obligatoirement comporter 3 méthodes au minimum : main, remplir_tableau et afficher_tableau.

Indication : la ligne de code suivante permet de générer un double aléatoire compris dans l'intervalle [0 , 1[:

`invokestatic java/lang/Math/random()D`

Rappel : l'instruction `d2i` permet de convertir un double en entier.

```
.class public RandomTab
.super java/lang/Object

.method public static remplir_tableau([I)V
    .limit stack 7
    .limit locals 3
    aload 0
    arraylength
    istore 1 ; taille du tableau
    iconst_0
    istore 2 ; compteur
boucle :
    iload 2
    iload 1
    if_icmpge end
    ; generation d'un nbre aleatoire
    aload 0
    iload 2
    invokestatic java/lang/Math/random()D
    ldc2_w 10.0 ; on manipule des doubles !
    dmul
    d2i
    iastore
    iinc 2 1
    goto boucle
end :
    return
.end method

.method public static afficher_tableau([I)V
    .limit stack 7
    .limit locals 3
    aload 0
    arraylength
    istore 1 ; taille du tableau
    iconst_0
    istore 2 ; compteur
boucle :
    iload 2
    iload 1
    if_icmpge end
    getstatic java/lang/System/out Ljava/io/PrintStream;
    aload 0
    iload 2
    iaload
    invokevirtual java/io/PrintStream/print(I)V
    getstatic java/lang/System/out Ljava/io/PrintStream;
    ldc " "
    invokevirtual java/io/PrintStream/print(Ljava/lang/String;)V
    iinc 2 1
    goto boucle
end :
    return
.end method

.method public static main([Ljava/lang/String;)V
    .limit stack 6
    .limit locals 4
    bipush 10
    newarray int ; création d'un tableau de 10 entiers
    astore 1 ; sauvegarde du tableau
    aload 1
    invokestatic RandomTab/remplir_tableau([I)V
    aload 1
    invokestatic RandomTab/afficher_tableau([I)V
    return
.end method
```