Conservatoire National des Arts et Métiers
Centre Régional de Basse Normandie
Informatique Avancé, NFP121

Questionnaire à choix multiple sur Java 1.5.
Série 4

Durée : 30 minutes

Documents autorisés : Aucun

Préambule :
Prenez bien le temps de lire les énoncés. Les questions peuvent être traitées dans un ordre quelconque.
Vous devez entourer clairement les numéros correspondant à vos réponses directement sur le support qui doit être rendu pour la correction. Il n'est pas nécessaire d'argumenter vos choix.

---

1. Quelle est la valeur par défaut d'un attribut de classe de type « boolean » :
    1. 0
    2. 1
    3. true
    4. false
    5. n'est pas initialisée

2. Quelle est la valeur par défaut d'une variable java de type « boolean » :
    1. 0
    2. 1
    3. true
    4. false
    5. n'est pas initialisée

3. Quelle est la taille mémoire occupée par une valeur de type int en Java :

1. 64 bits
2. 32 bits
3. 16 bits
4. 8 bits

4. Quelle est la taille mémoire occupée par une valeur de type long en Java :

1. 64 bits
2. 32 bits
3. 16 bits
4. 8 bits

5. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test {
    static void method(Object obj){
        System.out.println("Object");
    }
    static void method(String str){
        System.out.println("String");
    }
    public static void main(String args[]){
        method(null);
    }
}
```

1. affiche : Object
2. affiche : String
3. lance une RuntimeException
4. ne compile pas

6. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test{
    static void method(StringBuffer obj){
        System.out.println("StringBuffer");
    }
    static void method(String str){
        System.out.println("String");
    }
    public static void main(String args[]){
        method(null);
    }
}
```

1. affiche : StringBuffer
2. affiche : String
3. lance une RuntimeException
4. ne compile pas

7. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test{
      public void m1() {
            int m = 5;
            int i = 0;
            while (i < m) {
                  i++;
                  if (i == 3) {
                        break;
                  }
                  System.out.print(i);
            }
      }

    public static void main(String args[]){
        new Test().m1();
    }
}
```

1. affiche : 12345
2. affiche : 1245
3. affiche : 12

8. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test{
      public void m1() {
            int m = 5;
            int i = 0;
            while (i < m) {
                  i++;
                  if (i == 3) {
                        continue;
                  }
                  System.out.print(i);
            }
      }

    public static void main(String args[]){
        new Test().m1();
    }
}
```

1. affiche : 12345
2. affiche : 1245
3. affiche : 12

9. Dans le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class Test{
      public void m1() {
            List<Integer> l = new
ArrayList<Integer>();
            for (int i = 0; i < 1000; i++) {
                  l.add(i);
            }
            // boucle1
            int i = 0;
            while (i < l.size()) {
                  Integer integer = l.get(i);
                  i++;
            }
            //boucle2
            Iterator<Integer> it = l.iterator();
            while (it.hasNext()) {
                  Integer integer = it.next();

            }
      }

    public static void main(String args[]){
        new Test().m1();
    }
}
```

1. les deux boucles sont strictement équivalentes
2. la boucle 1 est préférable à la boucle 2
3. la boucle 2 est préférable à la boucle 1

10. Dans le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

import java.util.ArrayList;
import java.util.List;

public class Test{
      public void m1() {
            List<Integer> l = new
ArrayList<Integer>();
            for (int i = 0; i < 1000; i++) {
                  l.add(i);
            }
            // boucle1
            int i = 0;
            while (i < l.size()) {
                  Integer integer = l.get(i);
                  i++;
            }
            //boucle2
            for (Integer integer : l) {
            }
      }

    public static void main(String args[]){
        new Test().m1();
    }
}
```

1. les deux boucles sont strictement équivalentes
2. la boucle 1 est préférable à la boucle 2
3. la boucle 2 est préférable à la boucle 1

11. Les bonnes méthodes pour démarrer et arrêter les Threads sont :

    1. run, stop
    2. start, stop
    3. run, suspend
    4. start, suspend

12. Le mot clé synchronized permet de :

    1. Le mot clé synchronized permet de demander à un thread d'attendre la terminaison d'un thread particulier
    2. Le mot clé synchronized permet de poser un verrou sur un objet pour gérer une exclusivité entre plusieurs threads.
    3. La déclaration d'un attribut sous la forme « synchronized public int attribut1 ; » est légale.

13. Parmi les affirmations suivantes, lesquelles sont vraies :

    1. Java permet de gérer nativement des programmes contenant plusieurs threads tournant au sein d'une même machine virtuelle.
    2. Java permet de gérer nativement des programmes contenant plusieurs processus tournant au sein d'une même machine virtuelle.

14. Parmi les affirmations suivantes, lesquelles sont vraies :

    1. La méthode sleep de la classe Thread a pour objectif d'endormir un thread.
    2. Une attente active est préférable à une attente passive.
    3. La méthode notify permet de réveiller un thread endormi par l'appel à sleep.

15. Parmi les affirmations suivantes, lesquelles sont vraies :

    1. La méthode wait de la classe Thread permet à un thread de libérer le verrou de l'objet synchronisé et de se placer dans la liste d'attente associée à l'objet.
    2. La méthode notifyAll permet de rendre exécutables tous les threads présents dans la liste d'attente d'un objet.
    3. Un thread correspondant à un traitement long et coûteux en CPU doit avoir une priorité faible.

16. Parmi les affirmations suivantes, lesquelles sont vraies :

1. Une méthode « synchronized public void m1() » pose un verrou sur l'objet correspondant à « this ».
2. Une méthode « synchronized public static void m2() » pose un verrou sur l'objet correspondant à la classe où elle est définie.
3. Une méthode « synchronized public void m1() » ne peut être exécutée au même moment par deux threads sur deux instances de la classe où elle est définie.

17. Parmi les affirmations suivantes, lesquelles sont vraies :

1. Les méthodes des composants AWT et Swing gèrent le multi-thread.
2. L'EDT (Event Dispatching Thread) doit traiter l'ensemble des traitements graphiques AWT ou Swing.
3. L'appel à SwingUtilities.invokeLater permet d'ajouter un événement à l'EDT pour réaliser un traitement.

18. Parmi les affirmations suivantes concernant java.exe, lesquelles sont vraies :

1. Le classpath peut contenir des fichiers jar et des répertoires
2. Un fichier jar est un fichier archive contenant les fichiers compilés java.
3. Lors du lancement d'un programme java, il faut préciser le nom du fichier .class que l'on souhaite exécuter.
4. Lors du lancement d'un programme java, il est possible de préciser quelle méthode on souhaite exécuter.

19. Parmi les affirmations suivantes, lesquelles sont vraies :

1. La technologie sax permet de parcourir un document xml
2. La technologie dom permet de parcourir un document xml.
3. La technologie sax permet de créer un document xml.
4. La technologie dom permet de créer un document xml.

20. Parmi les affirmations suivantes, lesquelles sont vraies :

1. La technologie dtd permet d'analyser un document xml à l'aide d'une grammaire.
2. Les balises d'un document xml peuvent se chevaucher. <a>eert est légal
3. Un élément doit encapsuler tous les autres
4. Un namespace sert à créer des synonymes au sein d'un fichier xml

21. Parmi les affirmations suivantes, lesquelles sont vraies :

1. Le prologue d'un fichier xml permet de préciser l'encoding du document
2. La déclaration d'une dtd peut être interne ou externe.
3. < !--bla bla --> est un commentaire dans un fichier xml

22. Parmi les affirmations suivantes, lesquelles sont vraies :

1. La technologie xml est normalisée.
2. La technologie xml est spécifique au langage java.
3. La technologie xml permet la représentation de données sous forme d'un arbre