

**Tester vos connaissances, les examens se tiendront du 17/01 au 05/02**

**NB : Choisissez la ou les bonnes réponses. Une réponse est rejetée si elle n'est pas complète ou une réponse incorrecte est sélectionnée Chaque question vaut 01 point.**

**1. Java est un langage orienté objet qui :**

- ☐ S'exécute uniquement sur Windows
- ☐ Multiplateforme
- ☐ Permet de créer des applications Web
- ☐ Permet de créer des applications mobiles

**2. Lequel/Lesquels de ces mots clefs permettent de déclarer un nombre à virgule?**

- ☐ int
- ☐ double
- ☐ Short
- ☐ float

**3. Après les initialisations: String mot = "toto"; Boolean test = false;**

- ☐ La variable **mot** contient "toto"
- ☐ la variable **test** contient -1
- ☐ La variable **test** contient false
- ☐ La variable **test** contient une référence à false

**4. Dans une boucle (for, while, do... while)**

- ☐ On utilise **continue** pour sortir de la boucle et passer à la suite de la méthode
- ☐ On utilise **break** pour sortir de la boucle et passer à la suite de la méthode
- ☐ On utilise **return** pour sortir de la méthode
- ☐ **return** est interdit à l'intérieur d'une boucle

**5. L'instruction break est autorisée**

- ☐ Dans un bloc **if**
- ☐ Dans un bloc **for**
- ☐ Dans un bloc **switch**
- ☐ Dans un bloc **while**

**6. Quelle est la différence entre l'exécution d'un bloc while et celle d'un bloc do {...} while**

- ☐ Le bloc **while** peut être exécuté zéro fois
- ☐ Le bloc **do... while** est exécuté au moins une fois
- ☐ Le bloc **while** ne s'exécute que selon une condition vérifiée

**7. Après la séquence int x=2; int y=x; x=3;**

- ☐ y vaut 1
- ☐ y vaut 3
- ☐ Aucune de ces réponses

**8. Avec le bloc try / catch suivant**

```
try { int nombre = Integer.parseInt(texte) ; } // Integer.parseInt(texte) convertit un texte en valeur entière
```

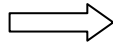
```
catch (NumberFormatException ex) { System.out.println... }
```

ajouter(nombre);

- ☐ ajouter() est exécuté seulement si **texte** peut être converti en nombre
- ☐ ajouter() est toujours exécuté après le **try / catch**
- ☐ ajouter() peut ne pas être exécuté si **texte** ne peut pas être converti en nombre, à condition de terminer le **catch** par un return

9. Cet extrait de code est sensé placer 1 dans m si a est plus grand que b, et 0 sinon. Fonctionne-t-il ? Si non, corrigez-le (a, b et m sont considérés comme déjà déclarés).

```
if (a > b) { m=1; } else (a>b) { m=0; }
```



10. J'ai écrit un programme dans le fichier Main.java. Quelles sont les commandes à écrire (dans l'invite de commande Windows) afin de compiler mon programme puis de l'exécuter ?

11. Combien d'instances de la classe A crée le code suivant ?

```
A x,u,v;  
x=new A();  
A y=x;  
A z=new A();
```

- ☐ Aucune
- ☐ Cinq
- ☐ Trois
- ☐ Deux

12. Que fait le programme suivant à la console ?

```
class Computer  
{  
    public static int mymethod (int a, int b)  
    {  
        return a + b;    }  
}  
public class Test  
{  
    public static void main (String[] args)    {  
        Computer pc = new Computer();  
        int result = pc. mymethod (9, 5);  
        System.out.println (result);    }    }
```

- ☐ Une erreur de compilation apparaît
- ☐ Afficher 14
- ☐ Afficher 45

13. Un fichier Java, peut contenir :

- ☐ Plusieurs classes et/ou plusieurs méthodes
- ☐ plusieurs méthodes main
- ☐ Une seule classe

14. Que va afficher le code à la console ?

```
int x = 5;  
double y = 3.5;  
if (x > 5 && y > x || x + 3 < y + 1)  
    System.out.println ("A");  
else if (x < y + 1 && x + y <= 12)    System.out.println ("B");  
else    System.out.println ("C");
```

- ☐ A
- ☐ B

- ☐ C
- ☐ Rien du tout, le code ne compile pas !

**15. Que fait le programme suivant ?**

```
import java.util.Scanner ;
class MyFile {
int mymethod()
{ Scanner clavier = new Scanner(System.in);
  int a = clavier.nextInt();  int m=a;
  for (int j = 2; j <= 5; j++)
    { a = clavier.nextInt();
      if (m>=a) {m=a;}; }
return m; } }
```

```
class Example {
public static void main(String [] args) {
MyFile exemple = new MyFile();
System.out.println(exemple.mymethod()); } }
```

- ☐ Affiche la plus grande valeur d'une suite de nombre
- ☐ Affiche la plus petite valeur d'une suite de nombres
- ☐ Ne fait rien
- ☐ Fait la permutation de valeurs

**16. Que fait cette instruction Java ? `Date today = new Date();`**

- ☐ Déclaration d'une nouvelle variable
- ☐ Création d'un objet (instance de classe)
- ☐ Invocation d'une méthode

**17. Que va afficher cette instruction à la console ? `System.out.println ("4" + "5" + " enfants");`**

- ☐ 9 enfants
- ☐ 45 enfants
- ☐ N'affiche rien
- ☐ 0 enfants

**18. Ce code : `<APPLET code="exemple.class"> </APPLET>`**

- ☐ Représente une applet java qui est intégrée dans un fichier HTML
- ☐ Le fichier exemple.class est un fichier java compilé
- ☐ Est intégré dans un fichier exemple.java
- ☐ Est intégré dans un fichier HTM et représente une applet java

**19. La méthode : `public void mymethode(int a) { .....}`**

- ☐ Ne renvoie pas de résultat
- ☐ Possède dans son corps le mot return
- ☐ Est une méthode qui n'est pas protégée
- ☐ Cette déclaration est erronée et manque le type du résultat renvoyé par la méthode

**20. Un fichier exemple.java qui contient plusieurs classes, pour l'exécuter, on doit :**

- ☐ Avoir une méthode main dans chaque classe
- ☐ Mettre une méthode main dans la classe exemple
- ☐ Mettre une seule méthode main dans uniquement la classe exemple