

1- Un langage de programmation orienté objet est :

- 1 un langage dont les traitements informatiques sont écrits dans une classe
- 2 un langage dont les traitements informatiques sont écrits en dehors de la classe

2- Dans un langage orienté objet, un principe fort est que les attributs non statiques sont alloués dans une instance d'une classe appelé un objet.

- 1 OUI
- 2 NON

3- Le paramètre -classpath ou la variable d'environnement CLASSPATH est utilisée pour désigner une liste de plusieurs path d'accès à des répertoires. Chacun de ces répertoires contient les fichiers .class ou

les packages utilisés dans la compilation ou dans l'exécution d'un programme JAVA.

- 1 OUI
- 2 NON

4- La compilation d'un fichier .java (usage de la commande javac)

- 1 crée un et un seul fichier .class
- 2 crée un fichier .jar
- 3 crée autant de fichier .class que de classes définies dans le fichier .java

5- Soit le fichier suivant C:\CodeJava\exercices\fr\cnam\util\Terminal.java. Le fichier Terminal.java contient en 1ère ligne : package exercices.fr.cnam.util;

Le répertoire C:\bin est vide. Dans C: on réalise la commande suivante : javac -d C:\bin C:\CodeJava\exercices\fr\cnam\util\Terminal.java

On obtient la création du fichier suivant : C:\bin\exercices\fr\cnam\util\Terminal.class

- 1 OUI
- 2 NON

6- Quand on compile un fichier .java contenant une classe, de nom Prog, qui contient une méthode main,

alors on obtient un fichier binaire Prog.exe que l'on peut ensuite exécuter sur le système d'exploitation.

1 OUI

2 NON

7- Soit la classe C1 dont tous les attributs sont privés et statiques. Soit la classe C2 appartenant au même

package que C1. Dans ce cas, les méthodes de C2 peuvent accéder directement aux attributs de C1

1 OUI

2 NON

8- Soit un fichier Prog.java contenant une classe publique Prog et une classe privée Prog2. Chacun de ces

classes contiennent une méthode public static void main(String... args)

1 On peut lancer une exécution avec la commande : java Prog

2 On peut lancer une exécution avec la commande : java Prog2

3 La classe Prog2 ne peut pas contenir de méthode main

9- En JAVA, une classe privé peut contenir plusieurs constructeurs

1 OUI X

2 NON

10- En JAVA, un objet est un pointeur

1 OUI

2 NON

11- Dans la classe String la méthode d'objet int indexOf(char c) permet de rechercher dans une String la

première occurrence du caractère c. Elle retourne la position de ce caractère.

1 OUI

2 NON

12- Soit le code suivant : `int[] tab_int; tab_int[0]=12; tab_int[1]=3;`

Ce code s'exécute correctement.

1 OUI

2 NON

13- La méthode equals de la classe Object permet de tester l'égalité de n'importe deux objets. Elle retourne VRAI si tous les attributs

sont égaux.

1 OUI

2 NON

14- Soit le code suivant :

```
public class Exemple {  
    public void main(String[] args) {  
        System.out.println(args[0]);  
    }  
}
```

Commande d'exécution : java Exemple toto

1 Ce programme ne se compile pas car il y a une erreur de syntaxe

2 L'exécution échoue car il y a une erreur d'exécution X

3 Ce programme s'exécute correctement et affiche : "toto"

15- Soit le code suivant est :

```
int[][] matrice = new Matrice[10][20];  
matrice[0] = new int[3];
```

On peut écrire ensuite les codes suivants :

1 System.out.println(matrice[0][0]);

2 System.out.println(matrice[0][3]);

3 System.out.println(matrice[1][1]);

16- Un attribut déclaré en final static est un attribut qui est protégé en écriture. Il est une constante du

programme.

1 OUI

2 NON

17- Soit le code suivant :

```
Livre l1 = new Livre(); Livre l2 = l1;  
l1.nom = "Face aux feux du soleil";  
ArrayList<Livre> livres = new ArrayList<Livre>(); livres.add(l1);  
l2.nom="Les cavernes d'acier";  
System.out.println(livres.get(0).nom);
```

Ce code affiche :

- 1 Les cavernes d'acier
- 2 Face aux feux du soleil

18- En JAVA, le type de retour d'une méthode peut être un type primitif (int, double, char,) ou un type

référence (objet, tableau) ou void

- 1 OUI
- 2 NON

19- En JAVA, un tableau (tab[]) ne peut contenir que des éléments de type primitif

- 1 OUI
- 2 NON

20- En Java, il est possible de modifier le contenu d'un tableau passé en paramètre d'une méthode

- 1 OUI
- 2 NON

21- S'il existe au moins un constructeur avec des paramètres et pas de constructeur sans paramètres alors le

constructeur par défaut est toujours accessible

- 1 OUI
- 2 NON

22- En JAVA, il est possible d'augmenter la taille d'un tableau.

- 1 OUI

2 NON

23- Le package qu'il faut importer pour utiliser la classe ArrayList est : java.lang

1 OUI

2 NON

24- En JAVA, pour créer un objet, on peut créer une méthode static de la classe dont le rôle est de créer un objet et de le retourner. Exemple :

```
public class Individu
{
    public String nom;
    public String prenom;
    public int age;
    static public Individu Individu(String nom, String prenom, int age)
    {
        Individu ind = new Individu();
        ind.nom=nom; ind.prenom=prenom; ind.age = age;
        return ind;
    }
}
```

Utilisation de cette méthode dans un programme Java : Individu ind =
Individu.Individu("LAFONT","Pierre",30);

1 OUI

2 NON

25- En JAVA, faire la déclaration : Individu ind;

consiste à créer un objet dont la classe d'appartenance est Individu. On peut alors utiliser les attributs de l'objet ind.

1 OUI

2 NON

26- La classe StringArrayList, de Java, permet de créer une collection de chaînes de caractère. Chaque élément de cette collection (un tableau) est une String.

1 OUI

2 NON

27- En JAVA, le code suivant permet de remplacer un caractère d'une chaîne de caractère par un autre :

```
String str = new String("abcdefg");  
str.replace(str,3,'x');  
System.out.println(str);
```

Ce code affiche : abcxefg

1 OUI

2 NON

28- Soit le code JAVA suivant :

```
public class C {  
    public int attribut;  
    public C(){attribut=10;}  
    public static void main(String[] args)  
    {  
        int attribut = 10;  
    }  
}
```

Ce code est correct

1 OUI

2 NON

29- Soit le code suivant :

```
Livre l1 = new Livre("LAFONT", "Pierre", 20);  
Exemple.changerNom(l1, "DUPONT");  
System.out.println(l1.getNom());
```

Avec

```
public class Exemple{  
    public static changerNom(Livre l, String nom)  
    { l.setNom(nom); }  
}
```

Ce code affiche :

```
1 DUPONT  
2 LAFONT  
2 EXCEPTION
```

30- Soit le code JAVA suivant :

```
ArrayList<String> tab1 = new ArrayList<String>(4);  
String[] tab2 = new String[4];  
tab1.add("UN"); tab1.add("DEUX");  
tab2[0]="UN"; tab2[1]="DEUX";  
boolean egal=true;  
for(int i=0;i<tab1.size();i++)  
    if ( ! (tab1.get(i).equals(tab2[i])) egal=false;  
System.out.println(egal);
```

```
1 Ce code affiche: true
```

```
2 Ce code affiche : false
```


