

Examen: n°455118 "Validation des acquis JAVA objet"

Copie n°12001416

Référence de l'examen: 250722-701-94324-455118

Exam effectué depuis l'adresse IP 176.165.222.40 en utilisant le login OTP matthieu21 / 4114

Commencé le 2025-07-22 16:33:02 et terminé le 2025-07-22 16:58:41

Légende

Correct

Devait être coché

Faux

Note : 15/20 (21 points / 28) - 28 questions

Barème: (Barème 1) Demi-point attribué si au moins la moitié des bonnes réponses sont cochées

(Examen ouvert le 2025-05-22 13:35:00)

1 Quelle est la différence entre une classe et un objet ?A Un objet est une instance d'une classeB Une classe est une instance d'un objet

(1 point / 1) Question à choix multiple

2 Soit la classe suivante : fr.pizzeria.model.Pizza. Dans quel package est cette classe ?A frB fr.pizzeriaC fr.pizzeria.modelD fr.pizzeria.model.Pizza

(1 point / 1) Question à choix multiple

3 Quelles sont les classes connues par défaut (ne nécessite pas d'import) ?A Les classes du même packageB Les classes du package java.utilC Les classes du package java.langD Aucune classe, il faut systématiquement importer une classe pour l'utiliser

(1 point / 1) Question à choix multiple

4 Soit le code suivant :

class Voiture {

 String a;
 static int b;

Lesquelles de ces affirmations sont correctes ?

A Les valeurs de l'attribut « a » seront différentes pour chaque objet créé à partir de la classe VoitureB La valeur de l'attribut « a » sera la même pour tous les objets créés à partir de la classe VoitureC Les valeurs de l'attribut « b » seront différentes pour chaque objet créé à partir de la classe VoitureD La valeur de l'attribut « b » sera la même pour tous les objets créés à partir de la classe VoitureE L'attribut « a » est appelé variable ou attribut d'instanceF L'attribut « a » est appelé variable de classeG L'attribut « b » est appelé variable ou attribut d'instanceH L'attribut « b » est appelé variable de classe

(1 point / 1) Question à choix multiple

5 Soit la classe suivante :

class Voiture {

int a;

}

Qu'affiche le code suivant ?

Voiture v = new Voiture();

System.out.println(v.a);

A 0

B null

C Une exception NullPointerException est lancéeD Le programme n'affiche rienE false

F

true

(0 point / 1) Question à choix multiple

6 Soit la classe suivante :

class Voiture {

boolean a;

}

Qu'affiche le code suivant ?

Voiture v = new Voiture();

System.out.println(v.a);

A 0B nullC Une exception NullPointerException est lancéeD Le programme n'affiche rienE falseF true

(0 point / 1) Question à choix multiple

7 Soit la classe suivante :

class Voiture {

String a;

}

Qu'affiche le code suivant ?

Voiture v = new Voiture();

System.out.println(v.a);

A 0B nullC Une exception NullPointerException est lancéeD Le programme n'affiche rienE falseF true

(0 point / 1) Question à choix multiple

8 Soit la classe suivante :

class Personne {

int nbPersonnes = 120 ;

}

Qu'affiche le code suivant ?

System.out.println(Personne.nbPersonnes);

A 120B nullC Le programme n'affiche rienD Une exception NullPointerException est lancéeE Le code ne compile pas

(0 point / 1) Question à choix multiple

9 Soit la classe suivante :

class Personne {

String nom = "Hugues";

}

Qu'affiche le code suivant ?

Personne p = new Personne();

System.out.println(p.nom);

A HuguesB nullC Le programme n'affiche rien

D

Une exception NullPointerException est lancée

Le code ne compile pas

(1 point / 1) Question à choix multiple

10 Soit la classe suivante :

```
class Marque {  
  
    static String pays = "FRANCE";  
  
}  
Qu'affiche le code suivant ?  
System.out.println(Marque.pays) ;  
A  FRANCE  
B  null  
C  Le programme n'affiche rien  
D  Une exception NullPointerException est lancée  
E  Le code ne compile pas
```

(1 point / 1) Question à choix multiple

11 Soit la classe suivante :

```
class Voiture {  
  
    String pays = "FRANCE";  
    public Voiture(String pays) {  
  
        this.pays = pays ;  
  
    }  
}
```

Est-ce que la ligne de code ci-dessous est une manière correcte d'instancier la classe Voiture ?

Voiture v = new Voiture();

A Oui

B Non

(1 point / 1) Question à choix multiple

12 Soit les classes suivantes :

```
package fr.bank.app ;  
public class A {  
  
    String nom = "Robert";  
  
    public String afficher() {  
  
        return this.nom;  
    }  
}
```

```
package fr.bank.app;  
public class B extends A {
```

```
    public String afficher() {
```

```
return "Bonjour je m'appelle "+super.afficher();
```

```
}
```

Qu'affiche l'exécution de la classe suivante ? :

```
package fr.bank;
import fr.bank.app.B;
public class Main {
    public static void main(String[] args) {
```

```
        B b = new B();
```

```
        System.out.println(b.afficher());
```

```
}
```

A Robert

B Le programme n'affiche rien

C Bonjour je m'appelle Robert

(1 point / 1) Question à choix multiple

13 A t-on le droit d'utiliser le modificateur « protected » sur une classe principale ?

Exemple :

```
protected class A {  
}
```

A Oui

B Non

(1 point / 1) Question à choix multiple

14 A t-on le droit d'utiliser le modificateur « private » sur une classe principale ?

Exemple :

```
private class A {  
}
```

A Oui

B Non

(1 point / 1) Question à choix multiple

15 Quel mot clé permet de créer une relation d'héritage entre deux classes ?

A extends

B implements

C Aucun des deux

(1 point / 1) Question à choix multiple

16 Quel mot clé permet de créer une relation d'héritage entre deux interfaces ?

A extends

B implements

C Aucun des deux

(1 point / 1) Question à choix multiple

17 Quel mot clé permet d'interdire l'héritage entre deux classes ?

A

- A final
B static
C void
D extends
E implements

(1 point / 1) Question à choix multiple

18 Est-il possible de créer une classe abstraite sans qu'elle ait une méthode abstraite ?

- A Oui
B Non

(0 point / 1) Question à choix multiple

19 Peut-on instancier une classe abstraite ?

- A Non
B Oui

(1 point / 1) Question à choix multiple

20 Peut-on définir un constructeur dans une classe abstraite ?

- A Oui
B Non

(1 point / 1) Question à choix multiple

21 La classe abstraite suivante est-elle correcte ?

public abstract class A {

 public abstract void afficher(){

 System.out.println("A") ;

}

- A Oui
B Non

(1 point / 1) Question à choix multiple

22 L'interface suivante est-elle correcte ?

public interface A {

 public void afficher(){

 System.out.println("A") ;

}

- A Oui
B Non

(0 point / 1) Question à choix multiple

23 Soit le code suivant :

class Voiture {

 static int b;

{}

Lesquelles de ces affirmations sont correctes ?

- A Les valeurs de l'attribut « b » seront différentes pour chaque objet créé à partir de la classe Voiture
B La valeur de l'attribut « b » sera la même pour tous les objets créés à partir de la classe Voiture
C L'attribut « b » est appelé variable ou attribut d'instance
D L'attribut « b » est appelé variable de classe

(1 point / 1) Question à choix multiple

24 Soit le code suivant :

```
List<String> list = new ArrayList<String>();  
list.add("A");  
list.add("B");  
list.remove(0);
```

Qu'affiche la ligne de code suivante ?

```
System.out.println(list.size());
```

- A 2
B 1
C 0

(1 point / 1) Question à choix multiple

25 Soit le code suivant :

```
HashMap<String, String> map = new HashMap<>();  
map.put(1, "Hello");  
map.put(2, "Bonjour");
```

Est-ce que le code ci-dessus compile ?

- A Non car la clé doit être de type String
B Oui car il est possible d'insérer dans une map des objets à partir d'un index.

(1 point / 1) Question à choix multiple

26 Quelle assertion est juste parmi celles proposées ci-dessous :

- A pour comparer les objets et les types primitifs il est conseillé d'utiliser ==
B pour comparer les objets et les types primitifs il est conseillé d'utiliser la méthode equals
C pour comparer les objets il est conseillé d'utiliser la méthode equals et pour les types primitifs ==
D pour comparer les objets il est conseillé d'utiliser la méthode compareTo et pour les types primitifs ==

(1 point / 1) Question à choix multiple

27 Quelle assertion est juste parmi celles proposées ci-dessous :

- A la méthode equals de la classe mère doit être redéfinie pour tester l'égalité de 2 instances.
B la méthode equals de la classe mère n'a pas besoin d'être redéfinie pour tester l'égalité de 2 instances. Elle teste l'égalité de tous les attributs de vos instances.

(0 point / 1) Question à choix multiple

28 Soir l'énumération suivante :

```
public enum Paiement {  
  
    public static final Paiement CHEQUE = new Paiement("Chèque");  
    public static final Paiement CB = new Paiement("CB");
```

```
    private String nom;  
    private Paiement(String nom){
```

```
        this.nom = nom;
```

```
    }  
    public String getNom(){
```

```
return this.nom;
```

```
}
```

```
}
```

Est-ce que l'énumération est correcte ?

A Non

B Oui

(1 point / 1) Question à choix multiple
