

Examen Blanc java

*Obligatoire

1. Adresse e-mail *

2. Quelle collection permet un accès indexé à ses éléments, mais ses méthodes sont synchronisées? *

Une seule réponse possible.

- ☐ HashSet
- ☐ Liste des tableaux
- ☒ Vecteur
- ☐ TreeMap

3. Quel type d'exception est levée lorsque nous exécutons un programme avec des objets qui ne sont pas de même type dans un ensemble trié? *

Une seule réponse possible.

- ☒ ClassCastException
- ☐ RuntimeException
- ☐ OutOfBoundsException
- ☐ Aucun

4. Combien de fois vous pouvez implementer l'interface java.lang.Comparable pour faire le tri? *

Une seule réponse possible.

- ☐ 0
- ☒ 1
- ☐ 2
- ☐ plusieurs

5. List et Set _____ Collection.

Une seule réponse possible.

- ☒ héritent de
- ☐ implémentent
- ☐ Les deux sont vrais.
- ☐ Aucune de ces réponses n'est vraie.

6. Pour garantir que les objets ajoutés dans HashMap sont unique il faut : *

Une seule réponse possible.

- ☐ Redéfinir la méthode equals
- ☐ Redéfinir la méthode hashCode
- ☒ Tout les réponses sont vrais
- ☐ Aucune de ces réponses n'est vraie.

7. Que signifie SAM dans le contexte de l'interface fonctionnelle?

Une seule réponse possible.

- ☐ Simple Abstract Markup
- ☐ Simple Active Markup
- ☒ Single Abstract Method
- ☐ Single Ambivalue Method

8. Quelle est la réponse exacte? *

```
1 interface Vehicle {  
2     void start();  
3     void run();  
4     void stop();  
5 }  
6 class Car implements Vehicle {  
7     public void start() {}  
8 }
```

Une seule réponse possible.

- ☒ Erreur de compilation
- ☐ Erreur d'exécution
- ☐ Une exception est levée
- ☐ Le code source est correct

9. Quelle affirmation est correcte concernant les interfaces en Java? *

Une seule réponse possible.

- ☐ L'interface est utilisée pour réaliser l'héritage multiple en java.
- ☐ L'objet d'une interface ne peut pas être créé.
- ☐ Une interface peut hériter une autre interface.
- ☒ Tout les réponses sont vrais.

10. Quel est la sortie du code suivant? *

```

1 interface A {
2     void show();
3 }
4 class B implements A {
5     public void show() {
6         System.out.println("Welcome To TestBlanc!");
7     }
8 }
9 public class Main {
10     public static void main(String[] args) {
11         A obj = new B();
12         obj.show();
13     }
14 }

```

Une seule réponse possible.

- ☐ L'objet d'une interface ne peut pas être créé.
- ☒ Welcome To TestBlanc!
- ☐ Erreur de compilation
- ☐ Tout les réponses sont fausse.

11. Soit le code suivant , c'est quoi le résultat ?

```

1 import java.util.*;
2 public class LetterASort {
3     public static void main(String[] args) {
4         ArrayList < String > strings = new ArrayList ();
5         strings.add("aAaA");
6         strings.add("AaA");
7         strings.add("aAa");
8         Collections.sort(strings);
9         for (String s: strings) {
10             System.out.print(s + " ");
11         }
12     }
13 }

```

Une seule réponse possible.

- ☐ Erreur de compilation
- ☐ aAa aAaAaAa
- ☐ AaA aAaaAaA
- ☒ AaAaAaA aAa
- ☐ Une exception est levée

12. Soit le code suivant , c'est quoi le résultat ? *

```
1 import java.util.*;
2 public class WrappedString {
3     private String s;
4     public WrappedString(String s) {
5         this.s = s;
6     }
7     public static void main(String[] args) {
8         HashSet < Object > hs = new HashSet < Object > ();
9         WrappedString ws1 = new WrappedString("aardvark");
10        WrappedString ws2 = new WrappedString("aardvark");
11        String s1 = new String("aardvark");
12        String s2 = new String("aardvark");
13        hs.add(ws1);
14        hs.add(ws2);
15        hs.add(s1);
16        hs.add(s2);
17        System.out.println(hs.size());
18    }
19 }
```

Une seule réponse possible.

- ☐ 2
- ☐ 3
- ☐ 4
- ☐ Erreur de compilation
- ☒ Une exception est levée

13. Choisir la réponse correcte *

```
1 public class Score implements Comparable < Score > {  
2     private int wins,  
3     losses;  
4     public Score(int w, int l) {  
5         wins = w;  
6         losses = l;  
7     }  
8     public int getWins() {  
9         return wins;  
10    }  
11    public int getLosses() {  
12        return losses;  
13    }  
14    public String toString() {  
15        return "<"+wins + ", "+losses + " > "  
16    }  
17    // Ajouter le code  
18 }
```

Une seule réponse possible.

- ☐ public int compareTo(Object o) { /*mode code here*/ }
- ☒ public int compareTo(Score other) { /*more code here*/ }
- ☐ public int compare(Score s1,Score s2){ /*more code here*/ }
- ☐ public int compare(Object o1,Object o2){ /*more code here*/ }

14. Choisir la réponse correcte *

```
1 public static void parse(String str) {  
2     try {  
3         float f = Float.parseFloat(str);  
4     } catch (NumberFormatException nfe) {  
5         f = 0;  
6     } finally {  
7         System.out.println(f);  
8     }  
9 }  
10 public static void main(String[] args) {  
11     parse("invalid");  
12 }
```

Une seule réponse possible.

- ☒ 0.0
- ☐ Erreur de compilation
- ☐ ParseException est levée par la méthode parse
- ☐ NumberFormatException est levée par la méthode parse

15. Que va être la sortie de ce programme ?

```
1 public static void main(String args[]) {  
2     HashMap < String, String > hashMap = new HashMap ();  
3     hashMap.put("voiture", "Audi");  
4     hashMap.put("voiture", "BMW");  
5     System.out.println(hashMap);  
6 }
```

Une seule réponse possible.

- ☐ Erreur d'exécution
- ☐ Erreur de compilation
- ☐ {voiture=Audi},{voiture=BMW}
- ☒ {voiture=BMW}

16. Qu'est ce que java.util.Collections ? *

Une seule réponse possible.

- ☐ L'interface à partir de laquelle s'étendent les listes, les files d'attente et les ensembles
- ☐ Le cadre des collections
- ☒ Une classe utilitaire à utiliser avec les collections
- ☐ Nous ne mentionnons jamais cela

17. L'exécution du code suivant :

```
1 try {  
2     int a = 5, b = 0, c, d;  
3     int[] tab = {  
4         1,  
5         2,  
6         3,  
7         4,  
8         5  
9     };  
10    c = a / b;  
11    d = tab[a];  
12 } catch (ArithmeticException e) {  
13     System.out.println("division par zéro ");  
14 }  
15 System.out.println("succès ");  
16 catch (ArrayIndexOutOfBoundsException e) {  
17     System.out.println("erreur d'accès ");  
18 }
```

Une seule réponse possible.

- ☐ affiche « division par zéro »
- ☐ affiche « succès »
- ☐ affiche « erreur d'accès »
- ☐ affiche « division par zéro » ensuite « succès » ensuite « erreur d'accès »
- ☒ présente une erreur de compilation

18. Parmi les expressions suivantes, lesquelles ne sont pas des expressions lambda valides? *

Une seule réponse possible.

- ☐ () -> {}
- ☐ () -> "Raoul"
- ☐ () -> {return "Mario";}
- ☒ (Integer i) -> return "Alan" + i;

19. Quelle est la réponse exacte ? *

```
class Test
{
    public static void main (String[] args)
    {
        try
        {
            int a = 0;
            System.out.println ("a = " + a);
            int b = 20 / a;
            System.out.println ("b = " + b);
        }

        catch (ArithmeticException e)
        {
            System.out.println ("Divide by zero error");
        }

        finally
        {
            System.out.println ("inside the finally block");
        }
    }
}
```

Une seule réponse possible.

- ☐ Divide by zero error
- ☐ inside the finally block
- ☒ a=0 ensuite Divide by zero error ensuite inside the finally block
- ☐ a=0

20. Soit prod une liste de produits (List<Produit>) Quelle instruction permet de trier les éléments *

Une seule réponse possible.

- ☒ `prod.stream().sorted((a, b) -> a.getPrix() - b.getPrix()).forEach((e) -> System.out.println(e));`
- ☐ `prod.stream().sorted((a, b) -> a.getPrix() - b.getPrix()) . collect(Collectors.toSet());`
- ☐ `prod.stream().collect(Collectors.toCollection()-> new TreeSet<Employee>((a,b)->a.getPrix - b.getPrix()))`
- ☐ `prod.stream().sorted().collect(Collectors.toCollection()-> new TreeSet<Employee>((a,b)->a.getPrix - b.getPrix()))`

21. Quelle est la bonne réponse ? *

Une seule réponse possible.

- ☐ La déclaration d'interface suivante compile sans erreur : `public interface I {public void m() {}}`
- ☒ La déclaration d'interface suivante compile sans erreur : `public interface I {public void m();}`
- ☐ Les déclarations d'interface et de classes suivantes compilent sans erreur : `public interface I {public void m();} public class C implements I {}`
- ☐ Aucune réponse n'est correcte

Ce contenu n'est ni rédigé, ni cautionné par Google.

Google Forms