



SousJava  
95 Questions

NOM : \_\_\_\_\_

CLASSER : \_\_\_\_\_

DATE : \_\_\_\_\_

1. Which annotation is used to create Pk-Fk relation b/w two tables?

☐ A @ForeignKey

☐ B @JoinedKey

☐ C @JoinColumn

2. Which annotation is used to link two tables through a relation table?

☐ A @RelationTable

☐ B @JoinTable

☐ C @LinkTable

3. Which attribute of @OneToMany is used to mark an entity as owned?

☐ A cascade

☐ B propertyRef

☐ C mappedBy

4. What is the JPA equivalent of hibernate.cfg.xml file?

☐ A configuration.xml

☐ B jpa.configuration.xml

☐ C persistence.xml

5. JPA implementation is provided by ---?

☐ A Ibatis

☐ B Hibernate

☐ C Toplink

6. What's true about the following @Entity association between House and Window?

```
@Entity
public class Window {
    @Id
    private int winNo;
    @ManyToOne
    private House aHouse;
}

@Entity
public class House {
    @Id
    private int houseNo;
    @OneToMany(mappedBy="aHouse")
    private List windows;
}
```

- ☐ A The association owner is the House class    ☐ B The association owner is the Window class
- ☐ C It's OneToMany bidirectional association    ☐ D It's OneToMany unidirectional association

7. Session.createQuery creates a new instance of Query for the given HQL query string

- ☐ A true    ☐ B false

8. While creating SessionFactory in hibernate, which design pattern should be adopted?

- ☐ A Prototype    ☐ B Singleton

9. \_\_\_\_\_ object is used to create SessionFactory object in hibernate ?

- ☐ A Transaction    ☐ B Configuration
- ☐ C Session    ☐ D Factory

10. in which extension in Hibernate is used to store database table configuration ?

- ☐ A .db    ☐ B .dbm
- ☐ C .sql    ☐ D .hbm

11. which of the following methods returns proxy object in Hibernate ?

- ☐ A load()    ☐ B loadAsync()
- ☐ C loadDatabase()    ☐ D getDatabase()

12. which method is easy for a java programmerto add a criterion ?

- ☐ A HCQL ☐ B HQL
- ☐ C SQL

13. Quel serait le résultat de ce programme

```
interface AddInterface2 {
    public int add(int a, int b);
}

public class LambdaQuiz {
    public static void main(String[] args) {
        AddInterface2 addInterface2 = (a, b) -> {
            System.out.println("Adding two numbers");
            a + b;
        };
        int sum = addInterface2.add(100, 200);
        System.out.println("Sum of two numbers : " + sum);
    }
}
```

- ☐ A Pas d'erreur, il imprimera 300 ☐ B Erreur de compilation
- ☐ C Runtime Exception

14. Quel serait le résultat de ce programme

```
interface AddInterface1 {
    public int add(int a, int b);
}

public class LambdaQuiz {
    public static void main(String[] args) {
        AddInterface1 addInterface1 = (a, b) -> -> return (a + b);
        int sum = addInterface1.add(100, 200);
        System.out.println("Sum of two numbers : " + sum);
    }
}
```

- ☐ A Runtime Exception ☐ B Erreur de compilation
- ☐ C Pas d'erreur, il imprimera 300

15. Quel serait le résultat de ce programme

```
interface AddInterface1 {
    public int add(int a, int b);
}

public class LambdaQuiz {
    public static void main(String[] args) {
        AddInterface1 addInterface1 = (a, b) -> {
            System.out.println("Adding two numbers");
            return (a + b);
        };
        int sum = addInterface1.add(100, 200);
        System.out.println("Sum of two numbers : " + sum);
    }
}
```

- ☐ A Runtime Exception ☐ B Erreur de compilation
- ☐ C Pas d'erreur, il imprimera 300

16. Lequel des éléments suivants n'est pas un concept POO en Java?

- ☐ A Héritage ☐ B Compilation
- ☐ C Encapsulation ☐ D Polymorphisme

17. Quand la surcharge de méthode est-elle déterminée ?

- ☐ A Au moment de l'exécution ☐ B Au moment de la compilation
- ☐ C Au moment de l'exécution ☐ D Au moment du codage

18. Quand la surcharge ne se produit pas ?

☐ A Quand il y a plusieurs méthodes avec le même nom, la même signature, le même nombre de paramètres mais un type différent

☐ B Quand il y a plusieurs méthodes avec le même nom mais avec une signature de méthode différente et un nombre ou un type de paramètres différent

☐ C Quand il y a plusieurs méthodes avec le même nom, le même nombre de paramètres et le type mais une signature différente

☐ D Quand il y a plusieurs méthodes avec le même nom, la même signature mais avec une signature différente

19. Quel concept de Java est utilisé en combinant des méthodes et des attributs dans une classe?

☐ A Héritage

☐ B Encapsulation

☐ C Polymorphisme

☐ D Abstraction

20. Comment s'appelle-t-on dans le cas où l'objet d'une classe mère est détruit donc l'objet d'une classe fille sera détruit également?

☐ A Agrégation

☐ B Association

☐ C Encapsulation

☐ D Composition

21. La surcharge d'une méthode peut remplacer l'héritage et le polymorphisme?

☐ A Faux

☐ B Vrai

22. Quels keywords sont utilisés pour spécifier la visibilité des propriétés et des méthodes ?

☐ A private

☐ B public

☐ C final

☐ D protected

☐ E abstract

23. Lequel des énoncés suivants à propos du Java est vrai ?

☐ A Aucun de ces réponses

☐ B Java ne supporte qu'une approche procédurale de la programmation.

☐ C Java prend en charge uniquement l'approche OOP.

☐ D L'approche procédurale et la POO sont prises en charge dans Java.

24. Lequel de ces mots-clés est utilisé pour créer une classe ?

- ☐ A struct ☐ B int
- ☐ C class

25. Lequel de ces opérateurs est utilisé pour allouer de la mémoire à un objet?

- ☐ A alloc ☐ B malloc
- ☐ C realloc ☐ D new

26. Lequel de ces opérateurs est utilisé pour allouer de la mémoire à un objet?

- ☐ A la méthode main () doit être publique ☐ B Le programme ne nécessitent pas de méthode main ()
- ☐ C Il ne peut y avoir qu'une seule méthode main () dans un programme ☐ D Chaque classe doit contenir une méthode main ()

27. Quelle est la sortie de ce programme ?

```
class Main
{
    public static void main(String args[])
    {
        int x = 9;
        if (x == 9)
        {
            int x = 4;
            System.out.println(x);
        }
    }
}
```

- ☐ A Erreur d'exécution ☐ B 4
- ☐ C Erreur de compilation ☐ D 3

28. Pouvons-nous créer une instance d'Enum en dehors d'Enum elle-même ?

- ☐ A Vrai ☐ B Faux

29. Quelle est la sortie de ce programme ?

```
class Hot
{
}

enum Enums extends Hot
{
    AAA, BBB, CCC, DDD;
}
```

- ☐ A Erreur de compilation ☐ B Erreur d'exécution
- ☐ C Exception EnumNotDefined ☐ D Il fonctionne bien

30. What can you use to create new instances in java ?

- ☐ A Another instance ☐ B Constructor
- ☐ C Private method ☐ D Field

31.

```
public static void main(String[] args) {
    for (int i = 0; i < 10; i++) {
        System.out.println("Hello");
    }
}
```

How many times will this code print "Hello"

A

5 times

B

9 times

C

10 times

D

Infinite number of times

32.

```
public static void main(String[] args) {
    String message = "Hello world!";
    String newMessage = message.substring(6, 12) + message.substring(12, 6);
    System.out.println(newMessage);
}
```

What is the output of this code ?

A

world !!world

B

A runtime exceptions is thrown

C

The code does not compile

D

world !world !

33.

How can you achieve runtime polymorphism in java ?

A

method overriding

B

method overrunning

C

method calling

D

method overloading

34.

```
public static void main(String[] args) {
    try {
        System.out.println("A");
        boomMethod();
        System.out.println("B");
    } catch (Exception e) {
        System.out.println("C");
    } finally {
        System.out.println("E");
    }
}

public static void boomMethod() {
    throw new Error();
}
```

Which letters will print when this code is run ?

A

A, B and E

B

C and E

C

A, C and E

D

A and E

35.

```
public static void main(String[] args) {
    String message = "Hello";
    for (int i = 0; i < message.length(); i++) {
        System.out.println(message.charAt(i+1));
    }
}
```

What is the output of this code ?

A

Hello

B

ello

C

A runtime exception is thrown

D

The code does not compile

36.

```
public class Hello {
    private String flavor ;
    private int numScoops ;
}
```

Write a constructor for this class :

A

Hello(int flavor, this. this.

String flavor = numScoops = }  
numScoops) { flavor ; numScoops ;

B

Hello() {flavor ="guava" ;numScoops = 2 ;}

C

Hello(int flavor,  
String numScoops) {  
flavor = numScoops =  
flavor ; numScoops ;

D

public Hello(int this. this.  
flavor, String flavor = numScoops =  
numScoops) { flavor ; numScoops ; }

37. Which keyword lets you call the constructor of a parent class ?

- ☐ A new ☐ B this  
☐ C super ☐ D parent

38. The runtime system starts your program by calling function first ?

- ☐ A iterative ☐ B main  
☐ C print ☐ D hello

39. What does this code print ?

```
public static void main(String[] args) {
    int a = 123451234512345;
    System.out.println("Hello".compareTo("Bonjour"));
}
```

- ☐ A false ☐ B a negative number  
☐ C a positive number ☐ D 0

40. what is the output of this code ?

```
public static void main(String[] args) {
    int a = 123451234512345;
    System.out.println(a);
}
```

- ☐ A a negative integer value ☐ B 12345100000  
☐ C will not compile ☐ D 123451234512345

41. What is the output of this code ?

```
public static void main(String[] args) {
    try {
        System.out.println("Hello World");
    } catch (Exception e) {
        System.out.println("C");
    } catch (ArithmeticException e) {
        System.out.println("D");
    } finally {
        System.out.println("E");
    }
}
```

- ☐ A Hello World ☐ B It will not compile  
☐ C Hello World ! ☐ D It will throw a runtime exception

42. Lequel des énoncés suivants est correct à propos de JDBC ?

- ☐ A Vous l'utilisez pour accéder à des bases de données relationnelles sans intégrer une dépendance à un type de base de données spécifique dans votre code. ☐ B L'API JDBC est une API pour accéder à différentes bases de données relationnelles.

☐ C JDBC signifie Java DataBase Connectivity.

43. Lequel des énoncés suivants est correct à propos de JDBC ?

- ☐ A De nouveaux pilotes peuvent être connectés à l'API JDBC sans changer le code client. ☐ B Lequel des énoncés suivants est correct à propos de JDBC ?

44. Lequel des éléments suivants est généralement utilisé pour modifier les bases de données ?

- ☐ A execute()
- ☐ B executeUpdate()
- ☐ C executeQuery()

45. Lequel des éléments suivants gère une liste de pilotes de base de données dans JDBC ?

- ☐ A Connection
- ☐ B DriverManager
- ☐ C JDBC driver
- ☐ D Statement

46. Lequel des énoncés suivants est correct à propos de la classe Statement de JDBC?

- ☐ A Statement encapsule une instruction SQL qui est transmise à la base de données pour être analysée et compilée.
- ☐ B Statement encapsule une instruction SQL qui est transmise à la base de données pour être planifiée et exécutée.

47. Lequel des énoncés suivants est plus efficace que Statement en raison de la précompilation de SQL ?

- ☐ A Statement
- ☐ B PreparedStatement
- ☐ C CallableStatement

48. Lequel des éléments suivants présente l'avantage d'utiliser PreparedStatement en Java?

- ☐ A Encourage l'injection SQL
- ☐ B Empêche l'injection SQL
- ☐ C Ralentissement des performances
- ☐ D Utilise plus de mémoire

49. Lequel des méthodes suivants est utilisé pour annuler une transaction JDBC ?

- ☐ A commit()
- ☐ B roll()
- ☐ C rollback()
- ☐ D removeTransaction()

50. What is the output of this code ?

```
public static void main(String[] args) {  
    new Boxer1(new Integer(4));  
}  
  
public class Boxer1 {  
    Integer i;  
    int x;  
    public Boxer1(int y) {  
        x = i*y;  
        System.out.println(x);  
    }  
}
```

- ☐ A Compilation fails because of an error in line " x = i+y;"
- ☐ B Compilation fails because of an error in line " new Boxer1(new Integer(4));"
- ☐ C A NumberFormatException occurs at runtime
- ☐ D A NullPointerException occurs at runtime
- ☐ E The value 4 is printed at the command line



51.

```

class Main {
    Object message() {
        return "Hello";
    }

    public static void main(String[] args) {
        System.out.print(new Main().message());
        System.out.print(new Main().message());
    }
}

class Main2 extends Main {
    String message() {
        return "World !";
    }
}

```

What is the result of this code ?

A

Hello !Hello !

B

Hello !World !

C

It will not compile

52.

L'interprétation des programmes Java est effectuée par

A

API

B

JVM

C

JDK

D

AWT

53.

Trouver la phrase qui n'est pas une caractérisation correcte de polymorphisme :

A

le polymorphisme est un mécanisme consistant à rassembler les données et les méthodes au sein d'une structure en cachant l'implémentation de l'objet.

B

le polymorphisme signifie que la même opération peut se comporter différemment sur différentes classes de la hiérarchie

C

le polymorphisme consiste à autoriser le même code à être utilisé avec différents types, ce qui permet des implémentations plus abstraites et générales.

D

le polymorphisme offre la possibilité à plusieurs objets de natures différentes d'exposer une interface identique au système, et ainsi répondre à un même message d'une manière qui leur est propre

54.

Qu'est-ce qui est faux pour les interfaces ?

A

Une interface déclare des méthodes sans les implémenter

B

Une interface peut être implémentée

C

Une interface peut être le type d'une référence

D

Une interface peut être instanciée

55.

```

class A {
    public static int f(int x) {
        return(x*x);
    }
    public int g(int x) {
        return (x);
    }
}

class B extends A {
    public static int f(int x) {
        return(x*x);
    }
    public int g(int x) {
        return (x*x);
    }
}

```

Pour les classes A et B définies comme suit :  
qu'affichera le code suivant ?

D d=new D();

A a =d;

System.out.println(a.f(2)\*a.g(3));

A

21

B

18

C

66

D

77

56. Java est un langage :

☐ A compilé et interprété

☐ B Ni compilé ni interprété

☐ C interprété

☐ D Compilé

57. En Java, puis-je hériter de plusieurs classes ?

☐ A NON

☐ B OUI

58. Quand la surcharge de méthode est-elle déterminée ?

☐ A Au moment de la compilation

☐ B Au moment du codage

☐ C Au moment de l'exécution

☐ D Au moment de l'exécution

59. Quand la surcharge ne se produit pas ?

☐ A Quand il y a plusieurs méthodes avec le même nom, le même nombre de paramètres et le type mais une signature différente

☐ B Quand il y a plusieurs méthodes avec le même nom mais avec une signature de méthode différente et un nombre ou un type de paramètres différent

☐ C Quand il y a plusieurs méthodes avec le même nom, la même signature, le même nombre de paramètres mais un type différent

☐ D Quand il y a plusieurs méthodes avec le même nom, la même signature mais avec une signature différente

60. Quel concept de Java est utilisé en combinant des méthodes et des attributs dans une classe ?

☐ A Abstraction

☐ B Polymorphisme

☐ C Héritage

☐ D Encapsulation

61. Comment s'appelle-t-on dans le cas où l'objet d'une classe mère est détruit donc l'objet d'une classe fille sera détruit également ?

☐ A Agrégation

☐ B Association

☐ C Encapsulation

☐ D Composition

62. Comment s'appelle-t-on l'objet à son propre cycle de vie et l'objet d'une classe fille ne dépend pas à un autre objet d'une classe mère ?

☐ A Encapsulation

☐ B Association

☐ C Agrégation

☐ D Composition

63. La surcharge d'une méthode peut remplacer l'héritage et le polymorphisme?

☐ A

Faux

☐ B

Vrai

64. Quels keywords sont utilisés pour spécifier la visibilité des propriétés et des méthodes ?

☐ A

final

☐ B

public

☐ C

abstract

☐ D

protected

☐ E

private

65. Lequel des énoncés suivants à propos du Java est vrai ?

☐ AJava ne supporte qu'une approche  
procédurale de la programmation.☐ B

Aucun de ces réponses

☐ CJava prend en charge uniquement  
l'approche OOP.☐ DL'approche procédurale et la POO sont  
prises en charge dans Java.

66. Lequel de ces mots-clés est utilisé pour créer une classe ?

☐ A

struct

☐ B

class

☐ C

int

67. Les expressions lambda dans java 8 sont basées sur \_\_\_\_

☐ A

Programmation procédurale

☐ B

Programmation fonctionnelle

☐ C

Tout les réponses sont vrais

☐ D

Programmation des données

68. Combien de méthodes existe-t-il dans une interface fonctionnelle en Java 8?

☐ A

1

☐ B

0

☐ C

2

☐ D

3

69. Dans les interfaces Java 8, les méthodes peuvent être \_\_\_\_

☐ A

abstract

☐ B

Aucune de ces réponses n'est vraie.

☐ C

default

70. Quel est le type de retour de l'expression lambda ?

- |                            |        |                            |          |
|----------------------------|--------|----------------------------|----------|
| <input type="checkbox"/> A | Object | <input type="checkbox"/> B | void     |
| <input type="checkbox"/> C | String | <input type="checkbox"/> D | Function |

71. Quelle est la nouvelle méthode introduite dans java 8 pour parcourir une collection ?

- |                            |                                |                            |                      |
|----------------------------|--------------------------------|----------------------------|----------------------|
| <input type="checkbox"/> A | for(String i : StringList)     | <input type="checkbox"/> B | StringList.forEach() |
| <input type="checkbox"/> C | foreach(String i : StringList) | <input type="checkbox"/> D | List.for()           |

72. Quels sont les deux types de Streams proposés par Java 8 ?

- |                            |                          |                            |                         |
|----------------------------|--------------------------|----------------------------|-------------------------|
| <input type="checkbox"/> A | Parallèle et aléatoire   | <input type="checkbox"/> B | Séquentiel et aléatoire |
| <input type="checkbox"/> C | Aléatoire et synchronisé | <input type="checkbox"/> D | Séquentiel et parallèle |

73. 

```
int i = 8;  
boolean b = false;  
float f = 5.3f;  
double d = 5.3;
```

 Étant donné les déclarations suivantes, lesquelles des instructions données dans les options ci-dessous compileraient ?

- |                            |          |                            |                  |
|----------------------------|----------|----------------------------|------------------|
| <input type="checkbox"/> A | i = 9;   | <input type="checkbox"/> B | d = i;           |
| <input type="checkbox"/> C | f = d;   | <input type="checkbox"/> D | b = (boolean) i; |
| <input type="checkbox"/> E | f = 3.8; |                            |                  |

74. Laquelle de ces méthodes supprime tous les éléments d'une collection ?

- |                            |           |                            |          |
|----------------------------|-----------|----------------------------|----------|
| <input type="checkbox"/> A | refresh() | <input type="checkbox"/> B | delete() |
| <input type="checkbox"/> C | clear()   | <input type="checkbox"/> D | reset()  |

75. Pourquoi les beans sont utilisés dans l'architecture J2EE au lieu d'écrire tout le code dans les JSP ?

- |                            |  |                            |   |
|----------------------------|--|----------------------------|---|
| <input type="checkbox"/> A | Permet la séparation des rôles entre les développeurs Web et les développeurs d'applications | <input type="checkbox"/> B | Permet l'intégration avec les outils de gestion de contenu (Content Management tools) |
|----------------------------|--|----------------------------|---|

76. Qu'est-ce qui est correct sur les Scriptlets JSP ?

- |                            |  |                            |  |
|----------------------------|--|----------------------------|--|
| <input type="checkbox"/> A | Les instructions dans un Scriptlet doivent suivre la syntaxe Java        | <input type="checkbox"/> B | Le point-virgule est nécessaire à la fin de chaque déclaration dans un Scriptlet |
| <input type="checkbox"/> C | Une boucle peut commencer dans un Scriptlet et se terminer dans un autre |                            |  |

77. Quelle méthode est appelée en premier à chaque appel d'une servlet ?

- |                            |           |                            |         |
|----------------------------|-----------|----------------------------|---------|
| <input type="checkbox"/> A | Servive() | <input type="checkbox"/> B | Start() |
| <input type="checkbox"/> C | init()    | <input type="checkbox"/> D | Run()   |

78. Dans JSP, comment pouvez-vous savoir quelle méthode HTTP (GET ou POST) est utilisée par la requête du client ?

- |                            |                                   |                            |                                   |
|----------------------------|-----------------------------------|----------------------------|-----------------------------------|
| <input type="checkbox"/> A | Impossible de savoir              | <input type="checkbox"/> B | En utilisant request.getMethod () |
| <input type="checkbox"/> C | En utilisant request.setMethod () |                            |                                   |

79. Quelle est la différence entre l'utilisation de « forward » et « sendRedirect() » ?

- |                            |   |                            |  |
|----------------------------|---|----------------------------|--|
| <input type="checkbox"/> A | Les deux méthodes fonctionnent de manière identique.                              | <input type="checkbox"/> B | forward s'exécute côté client pendant que sendRedirect() s'exécute côté serveur. |
| <input type="checkbox"/> C | forward s'exécute côté serveur pendant que sendRedirect () s'exécute côté client. |                            |  |

80. Comment une servlet peut appeler une page d'erreur JSP?

- |                            |   |                            |   |
|----------------------------|---|----------------------------|---|
| <input type="checkbox"/> A | La servlet doit rediriger la réponse vers la page d'erreur  | <input type="checkbox"/> B | La servlet doit transmettre la requête à l'URL de la page d'erreur spécifique. L'exception est transmise en tant qu'attribut nommé « javax.servlet.jsp.jspException » |
| <input type="checkbox"/> C | Lorsque la servlet lève l'exception, elle sera automatiquement interceptée par la page JSP appelant | <input type="checkbox"/> D | Cette fonctionnalité n'est pas supportée  |

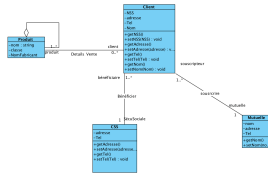
81. Toute demande du client pour accéder aux ressources dans le répertoire WEB-INF/ doit être retournée avec un \_\_\_\_\_?

- |                            |                 |                            |                          |
|----------------------------|-----------------|----------------------------|--------------------------|
| <input type="checkbox"/> A | BAD_REQUEST     | <input type="checkbox"/> B | SC_NOT_FOUND             |
| <input type="checkbox"/> C | ESC_BAD_REQUEST | <input type="checkbox"/> D | SC_INTERNAL_SERVER_ERROR |

82. **Par quoi une association entre deux classes peut-elle être caractérisée ?**

- |                            |            |                            |                   |
|----------------------------|------------|----------------------------|-------------------|
| <input type="checkbox"/> A | Des rôles  | <input type="checkbox"/> B | Des multiplicités |
| <input type="checkbox"/> C | Un losange | <input type="checkbox"/> D | Des noms          |

83.



La relation en boucle sur produit est :

A

une relation réflexive

B

une relation de généralisation

C

une relation de composition

D

une relation d'aggrégation

84.



Que se passe-t-il en cas de suppression de la voiture?

A

les roues peuvent être réutilisées par une autre voiture

B

les roues sont aussi supprimées

C

pas de réponse

85.



Quelle est la signification des contraintes?

A

la voiture a au moins 2 roues

B

le nombre de roue d'une voiture est inconnu

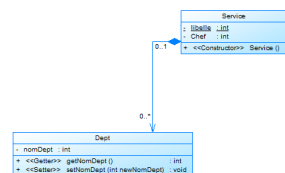
C

le nombre de roues d'une voiture évolue dans le temps

D

le nombre de roues d'une voiture est invariant

86.



Comment l'association suivante est transformée lors de la génération du code Java

A

un tableau (hasSet) de Dept dans Service et un objet de type Service dans Dept

B

une collection d'objection (hasSet) de Dept dans Service

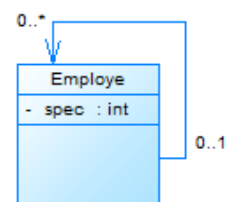
C

Sans réponse

D

un objet de type Service dans Dept

87.



Comment l'association est-elle transformée lors de la génération du code Java

A

Rien à cause de la navigabilité réduite

B

Sans réponse

C

Une collection d'objet de type Employe dans la classe Employe

D

un objet de type Employe dans la classe Employe

88. Laquelle des méthodes suivantes est utilisée pour stocker un objet dans un objet request ?

- |                            |                                       |                            |                                       |
|----------------------------|---------------------------------------|----------------------------|---------------------------------------|
| <input type="checkbox"/> A | setAttribute(String name, Object obj) | <input type="checkbox"/> B | setAttribute(String name, String obj) |
| <input type="checkbox"/> C | putAttribute(String name, Object obj) | <input type="checkbox"/> D | addObject(String name, Object obj)    |
| <input type="checkbox"/> E | addAttribute(String name, String obj) |                            |                                       |

89. L'objet « ServletContext » est accessible depuis lequel des objets suivants ?

- |                            |                |                            |                 |
|----------------------------|----------------|----------------------------|-----------------|
| <input type="checkbox"/> A | HttpSession    | <input type="checkbox"/> B | ServletResponse |
| <input type="checkbox"/> C | ServletConfig  | <input type="checkbox"/> D | HttpServlet     |
| <input type="checkbox"/> E | GenericServlet |                            |                 |

90. Lequel des listeners suivants est déclenché lorsqu'une session est initialisée ?

- |                            |                            |                            |                            |
|----------------------------|----------------------------|----------------------------|----------------------------|
| <input type="checkbox"/> A | HttpSessionChangedListener | <input type="checkbox"/> B | HttpSessionBindingListener |
| <input type="checkbox"/> C | SessionBindingListener     | <input type="checkbox"/> D | HttpSessionListener        |

91. Lequel des éléments suivants n'est pas inclus dans une URL ?

- |                            |                        |                            |                |
|----------------------------|------------------------|----------------------------|----------------|
| <input type="checkbox"/> A | Protocole              | <input type="checkbox"/> B | Nom du serveur |
| <input type="checkbox"/> C | La requête             | <input type="checkbox"/> D | Nom du port    |
| <input type="checkbox"/> E | L'adresse IP du client |                            |                |

92. Est-ce que l'instruction suivante est VRAI ou FAUX ?

Les quatre méthodes pour la gestion des sessions dans le contexte d'une application Web sont:

- Cookie,
- HttpSession object,
- URL rewriting
- Hidden value

- |                            |      |                            |      |
|----------------------------|------|----------------------------|------|
| <input type="checkbox"/> A | VRAI | <input type="checkbox"/> B | FAUX |
|----------------------------|------|----------------------------|------|

93. Laquelle des méthodes suivantes vous permettra d'obtenir une ou plusieurs valeurs d'un objet request ?

- |                            |                            |                            |                                 |
|----------------------------|----------------------------|----------------------------|---------------------------------|
| <input type="checkbox"/> A | getParameter(String name)  | <input type="checkbox"/> B | getParameterValues(String name) |
| <input type="checkbox"/> C | getAllParameters()         | <input type="checkbox"/> D | getAllAttributes()              |
| <input type="checkbox"/> E | getParameters(String name) |                            |                                 |

94. Which of the following command removes the target directory with all the build data before starting the build process?

- |                            |             |                            |             |
|----------------------------|-------------|----------------------------|-------------|
| <input type="checkbox"/> A | mvn clean   | <input type="checkbox"/> B | mvn build   |
| <input type="checkbox"/> C | mvn compile | <input type="checkbox"/> D | mvn install |

95. What is the use of mvn package command in Maven?

- |                            |   |                            |  |
|----------------------------|---|----------------------------|--|
| <input type="checkbox"/> A | Prints out the version of Maven you are running.                                  | <input type="checkbox"/> B | Clears the target directory into which Maven normally builds your project. |
| <input type="checkbox"/> C | Builds the project and packages the resulting JAR file into the target directory. |                            |  |



**Corrigé**

|      |      |      |      |
|------|------|------|------|
| 1.c  | 2.b  | 3.c  | 4.c  |
| 5.   | 6.c  | 7.b  | 8.b  |
| 9.b  | 10.d | 11.a | 12.b |
| 13.b | 14.b | 15.b | 16.b |
| 17.b | 18.c | 19.b | 20.d |
| 21.b | 22.  | 23.d | 24.c |
| 25.d | 26.d | 27.c | 28.b |
| 29.a | 30.b | 31.c | 32.b |
| 33.a | 34.d | 35.c | 36.d |
| 37.c | 38.b | 39.c | 40.c |
| 41.b | 42.  | 43.  | 44.b |
| 45.b | 46.  | 47.b | 48.b |
| 49.c | 50.d | 51.b | 52.b |
| 53.a | 54.d | 55.d | 56.  |
| 57.a | 58.a | 59.a | 60.d |
| 61.d | 62.c | 63.b | 64.  |
| 65.d | 66.b | 67.b | 68.a |
| 69.  | 70.d | 71.b | 72.d |
| 73.  | 74.c | 75.a | 76.  |
| 77.c | 78.b | 79.c | 80.b |
| 81.b | 82.  | 83.  | 84.b |
| 85.  | 86.b | 87.c | 88.a |
| 89.  | 90.d | 91.e | 92.a |
| 93.  | 94.a | 95.c |      |