

Question 1: Which of the following statements is true about arrays in Java?

An array has a fixed size.

- b. An array allows multiple dimensions.
- c. An array is mutable.
- d. An array is immutable.

Question 2: ¿Cuál es el comando en Bash para cambiar el directorio actual a uno especificado?

a. cd

- b. sh
- c. move
- d. goto
- e. chdir

Question 3: Which of the following statements is true about arrays in Java?

a. An array has a fixed size.

- b. An array allows multiple dimensions.
- c. An array is mutable.
- d. An array is immutable.

Question 4: ¿Cuál de los siguientes componentes es parte de una solicitud HTTP?

a. Headers, scripts, funciones.

b. URL, headers, cuerpo de la solicitud.

- c. Encabezados de la respuesta, cuerpo de la respuesta, estado de la respuesta.
- d. URL, cookies, archivos adjuntos.

Question 5: What is the output of the above Java code?

```
public class LoopQuestion {  
    public static void main(String[] args) {  
        int count = 0;  
  
        for (int i = 0; i < 10; i++) {  
            if (i % 2 == 0) {  
                count++;  
            }  
        }  
    }  
}
```

```
        System.out.println("Count: " + count);
    }
}
```

content_copy Use code [with caution](#).Java

- a. Count: 4
- b. Count: 6
- c. Count: 10
- d. Compilation fails
- e. Count: 5**

Question 6: Which method is used to sort elements of a List in natural order in Java?

- a. Collections.sort()**
- b. Collections.order()
- c. Arrays.sort()
- d. List.sort()

Question 7: ¿Qué significa que un cambio en el software sea retrocompatible?

- a. El cambio introduce nuevas funcionalidades que no afectan el comportamiento existente del software.
- b. El cambio corrige errores menores y realiza mejoras de rendimiento.
- c. El cambio puede romper la funcionalidad existente, requiriendo ajustes en el código que depende del software.
- d. El cambio garantiza que el software siga funcionando sin necesidad de modificaciones en el código que depende de él.**

Question 8: Which statement about the following code is correct?

```
class Base {
    public Base() {
        System.out.println("Base constructor");
    }

    public Base(String message) {
        System.out.println("Base constructor with message: " + message);
    }
}

class Derived extends Base {
```

```

public Derived() {
    super("Hello");
    System.out.println("Derived constructor");
}
}

class Test {
    public static void main(String[] args) {
        Derived derived = new Derived();
    }
}

```

- a. It will throw a compilation error.
- b. It will print "Base constructor" followed by "Derived constructor"
- c. It will print "Base constructor with message: Hello" followed by "Derived constructor"
- d. It will print "Base constructor" followed by "Base constructor with message: Hello" followed by "Derived constructor"

Question 9: ¿Cuál es la principal función de Ifrog Artifactory en un entorno de desarrollo de software?

- a. Gestionar y almacenar artefactos de software, como dependencias y bibliotecas, en un repositorio centralizado.
- b. Proporcionar un entorno de desarrollo integrado (IDE) para aplicaciones Java.
- c. Actuar como un servidor web para alojar sitios HTML y CSS.
- d. Ofrecer un sistema de control de versiones para proyectos de software.

Question 10: ¿Cuál es la nomenclatura y la ruta de creación de un archivo que sigue el formato (UUID[CODE].VERSION[COUNTRY].xml en un proyecto?

- a. VERSION.xml y se crea en la ruta src/main/resources/
- b. Ninguna opción es correcta
- c. COUNTRY[CODE(UUID)].xml y se crea en la ruta src/test/resources/
- d. [CODE]TUUA[COUNTRY].xml y se crea en la ruta src/main/java/
- e. [UUIDCODE[COUNTRY].xml y se crea en la ruta src/main/resources/

Question 11: ¿Cuál de los siguientes métodos de Mockito se utiliza para verificar que un método de un mock ha sido llamado un número específico de veces?

- a. when() & thenReturn()
- b. doReturn()
- c. doThrow()

d. Ninguna opción es correcta.

e. verify()

Question 12: Given the following classes, what will be the output of the program?

```
abstract class Animal {  
    public abstract void makeSound();  
}  
class Dog extends Animal {  
    @Override  
    public void makeSound(){  
        System.out.println("Bark");  
    }  
}  
  
public class Test{  
    public static void main(String[] args){  
        Animal myDog = new Dog();  
        myDog.makeSound();  
    }  
}
```

a. No output

b. Bark

c. Runtime error

d. Compilation error

Question 14

¿Cuál de las siguientes características es fundamental en una base de datos relacional?

a. Almacenamiento de datos en un sistema de archivos distribuido.

b. Ninguna opción es correcta.

c. Utilización de nodos y relaciones para representar datos.

d. Almacenamiento de datos en formato JSON.

e. Organización de datos en tablas con filas y columnas.

Question 15

Which line of code will compile successfully without any additional import statements?

```
public class PackageTest {  
    public static void main(String[] args) {  
        // Line A  
        String str = "Hello, World!";
```

```
// Line B
ArrayList<String> list = new ArrayList<>();
// Line C
File file = new File("example.txt");
// Line D
URL url = new URL("http://example.com");
}
}
```

- a. Line C
- b. Line D
- c. Line A**
- d. Line B

Question 16

Which of the following statements is true about the following code?

```
abstract class Shape {
    public abstract void draw();
    public void printShape(){
        System.out.println("This is a shape");
    }
}

class Circle extends Shape{
    @Override
    public void draw() {
        System.out.println("Drawing a circle");
    }
}

public class Test {
    public static void main (String[] args){
        Circle circle = new Circle();
        circle.draw();
        circle.printShape();
    }
}
```

- a. Compilation error because the Shape class is abstract
- b. The program will print "This is a shape" followed by "Drawing a circle"
- c. The program will print "Drawing a circle" followed by "This is a shape"**
- d. Compilation error because the Circle class does not implement the draw method.

Question 17

Una transacción APX es la unidad aplicativo que se ejecutará en APX Batch.

a. Verdadero

b. Falso

Question 18

¿Cuál es el propósito principal del archivo pom.xml en un proyecto Maven?

a. Gestionar la interfaz gráfica de usuario del proyecto.

b. Definir las dependencias, plugins y configuraciones del proyecto.

c. Contener la documentación del código fuente del proyecto.

d. Almacenar configuraciones de la base de datos del proyecto.

Question 19

Which section in the pom.xml file specifies the external libraries and dependencies required by the project?

a. <build>

b. <dependencies>

c. <repositories>

d. <plugins>

Question 20

What will be the output of the following code snippet?

```
public class ScopeTest {  
    private int value = 10;  
    public void printValue() {  
        int value = 20;  
        System.out.println(this.value);  
    }  
    public static void main(String[] args){  
        ScopeTest test = new ScopeTest();  
        test.printValue();  
    }  
}
```

a. 20

b. Compilation Error

c. 10

d. Runtime error.

Question 21

Which of the following is NOT part of the Agile Software development lifecycle?

a. Testing

- b. Coding
- c. Planning
- d. Documenting

Question 22

What is the primary purpose of a Data Transfer Object (DTO) in Software Design?

- a. To provide a user interface to the data model
- b. To encapsulate the data flow within the application.
- c. To handle addition operations directly
- d. To handle data flow between different layers of the application

Question 21

What is the primary purpose of a Data Transfer Object (DTO) in software design?

- a. To provide a user interface for data input.
- b. To encapsulate business logic within the application.
- c. To handle database operations directly.
- d. To transfer data between different layers or tiers of an application.

Question 22

Which declaration correctly initializes a boolean variable in Java?

- a. boolean f = "true";
- b. boolean c = (7 > 7);
- c. boolean a = (3 < 6);
- d. boolean d = (4 != 4);
- e. boolean b = (5 == 2);
- f. boolean e = (10 > 5 && 2 < 3);

Question 23

Which of the following statements about the 'throw' keyword is true?

- a. It is used to define the cleanup code that must be executed.
- b. It is used to declare that a method can throw an exception.
- c. It is used to catch exceptions thrown by other methods.
- d. It is used to manually throw an exception.

Question 24

Which of the following code snippets will throw a ClassCastException at runtime?

- a. class A {}
class B extends A {}
public class Test {

```

public static void main(String[] args) {
    B obj = new B();
    A a = (A) obj;
}
}

b. class A {}
class B extends A {}
public class Test {
    public static void main(String[] args) {
        A obj = new B();
        A a = (A) obj;
    }
}

c. class A {}
class B extends A {}
public class Test {
    public static void main(String[] args) {
        A obj = new A();
        B b = (B) obj;
    }
}

d. class A {}
class B extends A {}
public class Test {
    public static void main(String[] args) {
        B obj = new B();
        B b = (B) obj;
    }
}

```

Question 25

¿Cuál de las siguientes afirmaciones es correcta sobre la aserción assertEquals en JUnit?

- a. Se utiliza para verificar que una colección contiene un elemento específico.
- b. Se utiliza para verificar que dos valores son iguales.
- c. Se utiliza para verificar que una condición es verdadera.
- d. Se utiliza para verificar que dos objetos referencian la misma instancia.

Question 26: ¿Cual es la funcion principal del JDK (Java Development Kit)?

- a. Servir como un servidor web para aplicaciones Java.
- b. Proporcionar un entorno de ejecución para aplicaciones Java.
- c. Ofrecer herramientas necesarias para compilar, depurar y ejecutar aplicaciones Java.
- d. Ninguna opción es correcta.
- e. Permitir la edición de archivos HTML y CSS.

Question 27: Which of the following statements accurately describe the differences between Comparator and Comparable interfaces in Java?

- a. Comparable must be implemented by the class whose objects are being compared, whereas Comparator can be implemented by any class
- b. Comparable defines the compareTo method, whereas Comparator defines the compare method.
- c. Comparator allows for multiple ways of comparing objects, while Comparable allows only one way of comparing objects.
- d. All of the above.
- e. Comparable is used to compare the natural ordering of objects, whereas Comparator is used for custom ordering.

Question 28: What is the purpose of the “throws” keyword in a method declaration in Java?

- a. To catch exceptions thrown by other methods.
- b. To indicate the exceptions that the method can throw to the caller.
- c. To create a new exception instance.
- d. To throw an exception within the method.

Question 29: ¿Cuales de los siguientes comandos de Git se utilizan para gestionar ramas en un repositorio? (Seleccione todas las que correspondan).

- a. git commit
- b. git merge
- c. git checkout
- d. git branch
- e. git init

Question 30: ¿Cual de los siguientes patrones de diseño es adecuado para crear una estructura de objetos en forma de arbol para representar jerarquias parte-todo, permitiendo a los clientes tratar objetos individuales y compuestos de manera uniforme?

- a. Ninguna opción es correcta.

- b. Patrón Adaptador (Adapter Pattern).
- c. Patrón Fachada (Facade Pattern).
- d. Patrón Compuesto (Composite Pattern).
- e. Patrón Estrategia (Strategy Pattern).

Question 31: Which file is used to configure user specific settings in Maven?

- a. settings.xml
- b. build.xml
- c. pom.xml
- d. user.xml

Question 32: What will be the output of the following code snippets? `import java.util.ArrayList; import java.util.List; public class GenericTest{ public static <T> void addIfAbsent(List<T> list, T element){ if(!list.contains(element)){ list.add(element); } } public static void main(String[] args) { List<String> items = new ArrayList<>(); items.add("apple"); items.add("banana"); addIfAbsent(items, "cherry"); addIfAbsent(items, "apple"); System.out.println(items); } }`

- a. [apple, banana, cherry]
- b. [banana, cherry]
- c. [apple, banana]
- d. [apple, banana, cherry, apple]

Question 33: What will be the output of the following code snippet?

```
public class StringConcatenationTest {
public static void main(String[] args) {
    String str1 = "Hello";
    String str2 = "World";
    String str3 = str1 + " " + str2;
    String str4 = str1.concat(" ").concat(str2);
    String str5 = new StringBuilder().append(str1).append(" ").append(str2).toString();

    System.out.println(str1.equals(str2) + " ");
    System.out.println(str3.equals(str4) + " ");
    System.out.println(str3 == str5 + " ");
    System.out.println(str4 == str5);
}
}
```

- a. Compilation fails

b. false true false false

c. false false false false

d. true true false false

e. false false true true

f. true true true true

Question 34: Which of the following code snippets will result in a compilation error when implementing the vehicle interface?

```
interface Vehicle{  
    void start();  
    void stop();  
}
```

```
a. public class Bike implements Vehicle {  
    public void start(){  
        System.out.println("Bike starts");  
    }  
    void stop(){  
        System.out.println("Bike stops");  
    }  
}
```

```
b. public class Truck implements Vehicle {  
    public void start(){  
        System.out.println("Truck starts");  
    }  
    public void stop(){  
        System.out.println("Truck stops");  
    }  
    public void load(){  
        System.out.println("Truck loads");  
    }  
}
```

```
c. public class Scooter implements Vehicle {  
    public void start(){  
        System.out.println("Scooter starts");  
    }  
    public void stop(){  
        System.out.println("Scooter stops");  
    }  
}
```

```
d. public class Car implements Vehicle {
    public void start(){
        System.out.println("Car starts");
    }
    public void stop(){
        System.out.println("Car stops");
    }
}
```

Question 35: What will be the output of the following code snippet?

```
public class StaticNonStaticBlockTest {
    static {
        System.out.println("Static block");
    }

    {
        System.out.println("Instance block");
    }

    public StaticNonStaticBlockTest(){
        System.out.println("Constructor");
    }

    public static void staticMethod(){
        System.out.println("Static method");
    }

    public static void main(String[] args) {
        StaticNonStaticBlockTest test = new StaticNonStaticBlockTest();

        new StaticNonStaticBlockTest();
    }
}
```

- a. Static block
- Instance block
- Constructor
- Static method
- b. Static block
- Static method

Instance block
Constructor
c. Compilation error
d. Static method
Static block
Instance block
Constructor

Question 36: Una transacción APL representa un proceso funcional ACID donde se ejecuta un conjunto de operaciones lógicas.

Seleccione una:

- Verdadero
- Falso

Question 37: Which of the following statements accurately describe the relationships that can exist between classes in Java?

- a. Both inheritance and composition can be used together to model complex relationships.
- b. Inheritance represents an "is-a" relationship where one class derives from another class.
- c. Composition represents a "has-a" relationship where one class contains an instance of another class.
- d. Composition should be preferred over inheritance to promote code reuse and flexibility.
- e. Usage (or association) represents a "uses-a" relationship where one class uses methods or instances of another class.
- f. Inheritance should be preferred over composition to promote code reuse and flexibility

Question 38: Todo el acceso a los datos debe estar encapsulado en una biblioteca para facilitar la reutilización y control de acceso.

Seleccione una:

- Verdadero
- Falso

Question 39: ¿Cuál es el marco de trabajo en el que se basa APIX Batch?

- a. Spring Cloud
- b. Spring Microservices
- c. Spring Job
- d. Spring Batch
- e. Spring Web

Question 40: En el contexto de Maven, ¿Cuál es la función principal del archivo settings.xml?

- a. Ninguna opción es correcta.
- b. Generar informes de construcción y documentación.
- c. Especificar las dependencias del proyecto.
- d. Configurar la información del repositorio local y remoto, así como las credenciales y perfiles de usuario.
- e. Definir la estructura de directorios del proyecto.

Question 41: Where do you configure the plugins used for various build tasks in Maven's pom.xml?

- a. <repositories>
- b. <plugins>
- c. <dependencies>
- d. <build>

Question 42: What will be the result of the following code execution?

```
import java.util.ArrayList;
import java.util.List;

public class ArrayListTest {
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        list.remove(1);
        System.out.println(list);
    }
}
```

- a. [1, 2, 3]
- b. [1, 3]
- c. [1, 2]
- d. [2, 3]

Question 43: Which of the following methods can be used to remove all elements from an ArrayList?

- a. removeAll()
- b. clear()
- c. eraseAll()
- d. deleteAll()

Question 44: ¿Cuál de las siguientes opciones son capacidades ofrecidas en APIX?

- a. Todas las opciones son correctas.
- b. Sin integración con servicios de seguridad.
- c. Cold Deployment.
- d. Procesamiento transaccional.
- e. Procesamiento batch.

Question 45: ¿Cuáles son los tipos de componentes en APIX?

- a. Librerías
- b. Todas las opciones son correctas
- c. DTOS
- d. Transacciones
- e. Jobs

Question 46: ¿Cuál es la principal desventaja del antipatrón "contenedor mágico" en el desarrollo de software?

- a. Utiliza un número excesivo de patrones de diseño, complicando la estructura del código.
- b. Introduce dependencias circulares que son difíciles de resolver.
- c. Oculta demasiada lógica de negocio en un contenedor genérico, lo que hace que el código sea difícil de entender y depurar.
- d. Depende en gran medida de servicios externos, lo que reduce la portabilidad del software.

Question 47: ¿Cuál de los siguientes componentes de APIX representa una entidad comercial en forma de un bean?

- a. transacciones
- b. Librerías
- c. Jobs
- d. DTOs
- e. Todas las opciones son correctas.

Question 48: ¿Cuál de las siguientes afirmaciones describe mejor un Step en el contexto de Spring Batch?

- a. Un objeto de dominio que encapsula una fase independiente y secuencial de un trabajo por lotes.
- b. Una interfaz que define los métodos para realizar operaciones CRUD en los datos del trabajo por lotes.
- c. Una clase que gestiona la configuración de la base de datos utilizada por un trabajo por lotes.
- d. Un componente que se encarga exclusivamente de la validación de datos en un trabajo por lotes.

Question 49: Which method override is valid given the following classes?

```
class Parent {  
    void display() {  
        System.out.println("Parent");  
    }  
}  
  
class Child extends Parent {  
    // Override here  
}
```

- a. `public void display() { System.out.println("Child"); }`
- b. `private void display() { System.out.println("Child"); }`
- c. `void display() { System.out.println("Child"); }`
- d. `static void display() { System.out.println("Child"); }`

Question 50: ¿Cuál es la rama principal de Gitflow en la que se integran las nuevas funcionalidades antes de lanzarlas a producción?

- a. feature
- b. develop
- c. release
- d. hotfix
- e. master