

Quizlet

1. Which of the following statements is true about arrays in Java?
 - a. An array has a fixed size.
 - b. An array is mutable.
 - c. An array allows multiple dimensions.
 - d. An array is immutable

2. ¿Cuál de los siguientes componentes es parte de una solicitud HTTP?
 - a. URL, cookies, archivos adjuntos.
 - b. Headers, scripts, funciones.
 - c. Encabezados de la respuesta, cuerpo de la respuesta, estado de la respuesta.
 - d. URL, headers, cuerpo de la solicitud.

3.

```
public class LogicQuestion {  
    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);  
    }  
}
```

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

Debido al if, count solo incrementa cuando i sea un número par. Se ejecuta el for hasta que i sea igual a 10, por lo tanto, solo incrementa 5 veces.

4. Which method is used to sort elements of a List in natural order in Java?
 - a. List.sort()
 - b. Arrays.sort()
 - c. Collections.order()
 - d. Collections.sort()

List.sort() incrementa los elementos de un List pero necesita un comparator, Arrays.sort() se usa en Arrays y no en Collections. Collections.order() no existe.

5. ¿Qué significa que un cambio en el software sea retro-compatible?
- a. El cambio introduce nuevas funcionalidades que no afectan el comportamiento existente del software.
 - b. El cambio puede romper la funcionalidad existente, requiriendo ajustes en el código que depende de él.
 - c. El cambio garantiza que el software será funcional sin necesidad de modificaciones en el código que depende de él.
 - d. El cambio corrige errores menores y realiza mejoras de rendimiento.

```
6. 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");  
    }  
}  
public class Test {  
    public static void main(String[] args) {  
        Derived derived = new Derived();  
    }  
}
```

- a. Compile error.
- b. Base constructor
Derived constructor
- c. Derived constructor
Base constructor with message: Hello
- d. Base constructor with message: Hello
Derived constructor

Cuando se crea el objeto, se usa su constructor definido, este constructor llama al constructor de la clase padre pasándole un String antes de imprimir el mensaje.

7. ¿Cuál es la principal función de Jfrog Artifactory en un entorno de desarrollo de software?
- a. Proporcionar un entorno de desarrollo integrado (IDE) para aplicaciones Java.
 - b. Actuar como un servidor web para alojar sitios HTML y CSS.
 - c. Ofrecer un sistema de control de versiones para proyectos de software.
 - d. Gestionar y almacenar artefactos de software, como dependencias y bibliotecas, de manera centralizada.
8. ¿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. `mock.verifyCall(n)`
 - b. `mock.checkInvocations(count)`
 - c. `assertCalled(mock, numberOfTimes)`
 - d. `verify()` / `verify(mock, times(n))`

Los métodos de las opciones a, b, c no existen en Mockito.

9.

```
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 compila.
 - b. No se imprime nada.
 - c. Exception al ejecutarse.
 - d. Bark!

Ya que la clase Dog sobrescribe correctamente el método abstracto de la clase Animal, se imprime el mensaje indicado sin problemas.

10. ¿Cuál es el propósito principal de la anotación @Test en JUnit?
- a. Ejecutar un método antes de todos los métodos de prueba en una clase.
 - b. Ejecutar un método después de todos los métodos de prueba en una clase.
 - c. Marcar un método como un método de prueba.
 - d. Ignorar un método de prueba.

11. ¿Cuál de las siguientes características es fundamental en una base de datos relacional?
- a. Utilización de nodos y relaciones para representar datos.
 - b. Almacenamiento de datos en un sistema de archivos distribuido.
 - c. Almacenamiento de datos en formato JSON.
 - d. Organización de datos en tablas con filas y columnas.

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

```
public class Program{  
    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 D
- b. Line A
- c. Line B
- d. Line C

Un String es parte del paquete java.lang (es un paquete que se importa automáticamente).

13. 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. Drawing a circle
- b. Drawing a circle**  
**This is a shape**
- c. No compile
- d. This is a shape  
Drawing a circle

Después de crear el objeto, se llama al método draw() que imprime "Drawing a circle", después se llama al método de la clase abstracta printShape() que imprime "This is a shape".

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

- a. Almacenar configuraciones de la base de datos del proyecto.
- b. Gestionar la interfaz gráfica de usuario del proyecto.
- c. Contener la documentación del código fuente del proyecto.
- d. Definir las dependencias, plugins y configuraciones del proyecto.**

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

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

<plugins> se usa para definir complementos o herramientas que ejecutan tareas específicas durante el ciclo de vida del build de Maven. <build> se usa para configurar el proceso de construcción del proyecto, dentro de esta etiqueta se definen los plugins. <repositories> se usa para definir repositorios adicionales desde donde Maven puede descargar dependencias si no están disponibles en el repositorio central de Maven (Maven Central).

16. 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. Compilation error
- b. 20
- c. Runtime error
- d. 10

En el método printValue() se especifica a la variable de instancia del objeto (con this) y no a la variable definida en el método.

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

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

18. Which declaration correctly initializes a boolean variable in Java?

- a. boolean f = "true";
- b. boolean f = () => f;
- c. boolean f = (1 + 0);
- d. boolean d = (a < b);
- e. boolean b = 0 < 1;
- f. boolean a = (10 > 5 && 2 < 3);

En la opción a se define un String por las "". En la opción b se intenta definir una expresión lambda. En la opción c se intenta definir el resultado de una operación numérica. La opción d podría ser correcta pero no se sabe si las variables son de un tipo válido a comparar.

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

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

d. It is used to manually throw an exception.

20. Which of the following code snippets will throw a ClassCastException

- 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 B();
 B b = (B) obj;
 }
}
```
- d. 

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

Una variable de referencia del tipo de una subclase, no puede apuntar a un objeto de una superclase.

21. ¿Cuál de las siguientes afirmaciones es correcta sobre la aserción `assert()` en `jUnit`?

- a. Se utiliza para verificar que dos objetos referencian la misma instancia.
- b. Se utiliza para verificar que una colección contiene un elemento específico.

- c. Se utiliza para verificar que una condición es verdadera.
- d. Se utiliza para verificar que dos valores son iguales.

22. ¿Cuál es la función principal del JDK (Java Development Kit)?

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

23. Which of the following statements accurately describe the differences between Comparator and Comparable interfaces in Java?

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

24. What is the purpose of the "throws" keyword in a method declaration in Java?

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

25. ¿Cuáles de los siguientes comandos de Git se utilizan para gestionar ramas en un repositorio? (Seleccione todas las que correspondan).

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

git commit se usa para registrar cambios en un repositorio. git init se usa para inicializar un repositorio.

26. ¿Cuál de los siguientes patrones de diseño es adecuado para crear una estructura de objetos en forma de árbol para representar jerarquías parte-todo, permitiendo a los clientes tratar objetos individuales y compuestos de manera uniforme?

- a. Patrón compuesto (Composite Pattern).
- b. Patrón adaptador (Adapter Pattern).
- c. Patrón estrategia (Strategy Pattern).



d. Patrón fachada (Facade Pattern).

27. Which file is used to configure user specific settings in Maven?

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

28. 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]
- b. [banana, cherry]
- c. [apple, banana, cherry]
- d. [apple, banana, cherry, apple]

El método contiene un if que define que si un elemento no existe, este se añadirá al ArrayList.

29. 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. true true false false
- c. false true false false
- d. false false false false

Primero se imprime false, ya que se compara si "Hello" y "World" con el equals. Después se imprime true, ya que str3 y str4 tienen el mismo contenido (aunque no apuntan al mismo objeto, porque el String sobrescribe el método equals). Después se imprime false ya que se compara con ==, este operador compara las referencias y no el contenido, lo mismo para la última comparación.

30. 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 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");
 }
}

```
- b. 

```

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

```

```

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

```

Los métodos abstractos de una interfaz son por defecto public, por lo tanto, una clase al sobrescribir estos métodos tiene que definirlos como public, ya que no se puede reducir la visibilidad.

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

Instance block

Constructor

Static method

c. Static block

Instance block

Constructor

Instance block

Constructor

d. Compilation error

En una clase siempre se ejecutan al inicio (y una sola vez) los bloques estáticos. Al definir un objeto de la clase se ejecutan primero sus bloques de instancia y por ultimo los constructores.

32. En el contexto de bases de datos relacionales, si una transacción cumple con la propiedad de Aislamiento (Isolation) del principio ACID, esto significa que:

- a. Las transacciones se ejecutan como si fueran la única operación en el sistema, sin interferencia de otras transacciones concurrentes.
- b. Los cambios realizados por una transacción son permanentes y sobreviven a fallos del sistema.
- c. Una transacción debe mantener la consistencia de la base de datos antes y después de su ejecución.
- d. Las operaciones de una transacción se ejecutan todas o ninguna, no hay estados intermedios.

33. 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.

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

- a. Falso
- b. Verdadero

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

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

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

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

37. 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. [1,3]
- b. 1 2

El método remove() elimina el índice 1 y no el valor. Para eliminar el valor se utilizaría valueOf(): list.remove(valueOf(1));

38. Which of the following methods can be used to remove all elements from an ArrayList?

- a. deleteAll()
- b. clear()

- c. removeAll()
- d. eraseAll()

deleteAll() y eraseAll() son métodos que no existen. removeAll() existe, pero elimina los elementos que están presentes en otra Collection.

39. ¿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.

40. ¿Cuál de las siguientes afirmaciones describe mejor un Step en el contexto de Spring Batch?

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

41. Which method override is valid given the following classes?

```
class Parent {
 void display() {
 System.out.println("Parent");
 }
}

class Child extends Parent {
 // Override here
}
```

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

La opción a no es correcta, ya que un método estático no puede sobrescribir a uno instancia de la clase padre, y no se pueden vincular a una instancia de la clase. La opción c

es invalida ya que no se puede reducir la visibilidad de un método al sobrescribir, la opción d podría ser valida, pero en el caso de que las clases no formaran parte del mismo paquete daría error.

42. ¿Cuál es la rama principal de GitHub en la que se integran las nuevas funcionalidades antes de lanzarlas a producción?

- a. analyze
- b. develop

43. ¿Cuál es el comando en Bash para cambiar el directorio actual a uno especificado?

- a. sh
- b. cd
- c. move
- d. chdir

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

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

### **Simulador JavaAPX**

8. ¿Cuál de las siguientes herramientas proporciona análisis de calidad del código, detectado errores, vulnerabilidades y malas prácticas en el código fuente?

- a. Artifactory
- b. Jenkins
- c. Bitbucket
- d. SonarQube
- e. Jira

13. ¿Cuál de los siguientes comandos de Maven se utiliza para compilar un proyecto y crear un .jar pero sin instalarlo en el repositorio local?

- a. mvn package
- b. mvn clean
- c. mvn install
- d. mvn compile

mvn clean elimina los archivos generados anteriormente, como los directorios target/ y los archivos .class. mvn install si compila y crea el .jar pero también instala el archivo restante en el repositorio local. mvn compile solo compila el código fuente.

16. ¿Cuál de las siguientes ramas en Gitflow se utiliza para aplicar correcciones rápidas directamente en producción?
- a. feature
  - b. Release
  - c. develop
  - d. hotfix
  - e. master
22. ¿Cual de los siguientes comandos de Git se utilizan para crear una nueva rama?
- a. Git commit -m "mensaje"
  - b. Git branch <nombre-de-rama>
  - c. Git switch <nombre-de-rama>
  - d. Git merge <nombre-de-rama>
  - e. Git checkout -b <nombre-de-rama>
24. ¿Cuál es el propósito principal de OSGi (Open Service Gateway initiative) en el desarrollo de software modular?
- a. Servir como un sistema de control de versiones para proyectos de software
  - b. Proporcionar un entorno de desarrollo integrado (IDE) para aplicaciones Java
  - c. Gestionar dependencias y permitir el despliegue dinámico de módulos en aplicaciones Java
  - d. Actuar como un servidor de aplicaciones para ejecutar aplicaciones empresariales Java
30. ¿Cuál es la principal característica del antipatrón BLOB (Binary Large Object) en el diseño de software?
- a. Uso excesivo de primitivas en lugar de objetos, lo que dificulta la extensión y el mantenimiento del código.
  - b. Almacenamiento de grandes cantidades de datos binarios en una base de datos relacional.
  - c. Implementación de interfaces que no tienen una funcionalidad claramente definida, lo que lleva a un código confuso y propenso a errores.
  - d. Creación de una clase o módulo que acumula demasiadas responsabilidades y se vuelve difícil de mantener y modificar.
33. ¿Cuál es la nomenclatura y la ruta de creación de un archivo que sigue el formato [UUAAT][CODE]-[VERSION]-[COUNTRY].xml en un proyecto?
- a. [CODE][UUAAT]-[VERSION]-[COUNTRY].xml y se crea en la ruta src/main/java/.
  - b. [COUNTRY]-[VERSION]-[CODE][UUAAT].xml y se crea en la ruta src/test/resources/.



- c. Ninguna opción es correcta.
- d. [UUAA][CODE]-[VERSION]-[COUNTRY].xml y se crea en la ruta src/main/resources/.
- e. [VERSION]-[UUAA][CODE]-[COUNTRY].xml y se crea en la ruta src/main/resources/.

**36.** ¿Cuál de las siguientes opciones es verdadera de Spring Batch?

- a. Es una abstracción que representa la salida de un Step
- b. Marco de trabajo para lotes diseñado para permitir el desarrollo de aplicaciones por lotes
- c. Marco de trabajo basado en jobs y steps
- d. Es una abstracción que representa la recuperación de la entrada de un Step
- e. Es una abstracción que representa el procesamiento comercial de un ítem

**48.** ¿Cuál es el propósito de la directiva Import-Package en el archivo pom?

- a. Definir los paquetes OSGi que deben ser importados para que funcione correctamente.
- b. Especificar las dependencias del proyecto y sus versiones.
- c. Establecer las propiedades del proyecto, como el nombre y la versión.
- d. Configurar los plugins que se utilizarán durante el ciclo de vida de construcción del proyecto.