1. ¿Qué es un “code smell”?
2. Un componente de la biblioteca estándar de Java
3. Un error en tiempo de compilación que se produce en Java
4. Un indicador de que puede haber un problema en el código que puede ser dificil de detectar o que podría ser una fuente potencial de errores o problemas de mantenimiento en el futuro.
5. Una práctica de programación recomendada en Java.
6. ¿Cuál de las siguientes afirmaciones son verdaderas? (Elije todas las correctas)
7. Las excepciones de tiempo de ejecución son lo mismo que las excepciones comprobadas.
8. Las excepciones en tiempo de ejecuciones son lo mismo que las excepciones no comprobadas.
9. Solo pueden manejar subclases de exception.
10. Solo puede declarar excepciones comprobadas (checked).
11. Puede declarar solo excepciones no comporbadas.
12. ¿Cuáles son las salidas del siguiente código? (Elije todas las correctas)

Texto

Descripción generada automáticamente

1. El código no compila.
2. s3 = a
3. s1 = a
4. s2 = s2
5. s1 = s1
6. s3 = null
7. s2 = s2b
8. ¿Cuáles de las siguientes opciones son verdaderas? (Elije todas las correctas)
9. Java permite la sobrecarga de operadores.
10. Java es un lenguaje de programación funcional.
11. Java tiene punteros a ubicaciones específicas en la memoria.
12. Java es un lenguaje orientado a objetos.
13. El código java compilado en Windows puede ejecutarse en Linux.
14. Java es un lenguaje procedimental.
15. Selecciona la respuesta correcta con respecto al resultado del bloque de código.

Texto

Descripción generada automáticamente

1. Error en tiempo de ejecución.
2. No compila.
3. t,s,c
4. c,s,t
5. ¿Cuál es la función del operador de doble dos puntos (::) en Java 8?
6. El operador doble dos puntos se utiliza para crear una nueva instancia de una clase en Java 8.
7. EL operador de doble dos puntos no se utiliza en java 8.
8. El operador de doble dos puntos se utiliza para acceder a métodos estáticos en Java 8.
9. El operador de doble dos puntos se utiliza para acceder a métodos no estáticos en Java 8.
10. ¿Qué significa el acrónimo CRUD en una API REST?
11. Code, Register, Update, Debug
12. Create, Read, Update, Delete
13. Call, Receive, Use, Debug
14. Customize, Request, Use, Debug
15. ¿Qué es un bean en Spring?
16. Un archivo de configuración XML que se utiliza para definir la estructura de una tabla de base de datos.
17. Una instancia de una clase que se administra por el contendedor de Spring.
18. Una herramienta de inyección de dependencias que se utiliza para inyectar dependencias en una clase.
19. Una clase que se utiliza para configurar la conexión a una base de datos.
20. Son patrones de diseño de software estructural.
21. Builder, Singleton, Prototype y abstract factory.
22. Adapter, Proxy, Prototype y Bridge.
23. Agile, Builder, Singleton y Prototype.
24. Adapter, Bridge, Proxy y Composite.
25. ¿Para qué nos sirve utilizar un profile dentro del archivo pom.xml?
26. Etiqueta por la cual podemos definir la versiones de nuestras dependencias.
27. Es la etiqueta por la cual podemos definir las características que tendrá nuestro proyecto al ser compiladas.
28. Etiqueta por la cual definimos los parámetros de conexión a un repositorio.
29. No existe esta etiqueta en Maven.
30. ¿Cuál es el comando utilizado para actualizar la rama local con los cambios de la rama remota en Git?
31. git checkout
32. git clone
33. git push
34. git pull
35. Dadas las siguientes clases Vehicle y Car

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

¿Cuál de las siguientes instrucciones pueden colocarse en la línea 7 para que la clase Car compile correctamente? (Seleccione las que apliquen)

1. mileage = 15285;
2. Ninguna de las anteriores.
3. make = “Honda”;
4. year = 2009;
5. model = “Pilot”;
6. Enumere cuatro interfaces de la API colecciones
7. List, Map, Set, Queue.
8. ArrayList, Map, Set, Queue.
9. List, HashMap, HashSet, PriorityQueue.
10. List, Map, HashSet, PriorityQueue.
11. De los siguientes ¿Qué tipos de declaraciones se deben usar para contar la cantidad de monedas de cinco centavos en cadenas de varias monedas? (no se ve bien el final de la pregunta)
12. Assertion
13. Iteration
14. Assignment
15. Conditional
16. Selecciona la respuesta correcta con respecto al resultado del bloque de código.

Texto

Descripción generada automáticamente

1. Head  
   Object  
   Tail  
   Side
2. No compila
3. Side  
   Object  
   Tail  
   Side
4. Head  
   Head  
   Tail  
   Tail
5. Side  
   Head  
   Tail  
   Side
6. ¿Cuál es la salida al ejecutar el siguiente código?

Texto, Carta

Descripción generada automáticamente

1. roar roar!!!
2. roar!!! Roar
3. Se lanza una excepción
4. roar!!! roar!!!
5. roar roar
6. El código no compila
7. ¿Cuál de los siguientes es cierto acerca de una subclase completa?
8. Una subclase concreta no se puede marcar como final.
9. Una subclase concreta debe implementar todos los métodos definidos en una interfaz heredada.
10. Una subclase concreta debe implementar todos los métodos abstractos heredados.
11. Una subclase concreta puede declararse como abstracta.
12. Los métodos abstractos no pueden ser anulados por una subclase concreta.
13. ¿Cuál es la salida del siguiente código?

Una captura de pantalla de una computadora

Descripción generada automáticamente con confianza media

1. Error de compilación línea 12
2. Error compilación línea 16
3. Caught string
4. Error compilación línea 2
5. Caught Object
6. Seleccione la respuesta que considere correcta, dado el siguiente bloque de código.

Texto

Descripción generada automáticamente

1. 3
2. 9
3. 14
4. 6
5. ¿Qué declaración representa una declaración válida que permitirá la inclusión de clases del paquete java.util?
6. #include java.util.\*;
7. #include java.util;
8. import java.util.\*;
9. import java.util;
10. ¿Qué es la cobertura de código?
11. La cantidad de veces que se ejecuta una línea de código.
12. La cantidad de errores detectados por una prueba.
13. La cantidad de código que se ejecuta durante una prueba.
14. La cantidad de tiempo que tarda una prueba en ejecutarse.
15. ¿Qué es Git?
16. Una herramienta de automatización de compilación que se utiliza para compilar un proyecto.
17. Una herramienta de generación de informes que se utiliza para generar informes sobre el rendimiento de una aplicación.
18. Una herramienta de gestión de dependencias que se utiliza para descargar bibliotecas y paquetes en un proyecto de Java.
19. Una herramienta de control de versiones que se utiliza para almacenar y administrar el código fuente de un proyecto.
20. ¿Cuál es el formato correcto para hacer un commit en Git?
21. Descripción breve del cambio y nombre del autor.
22. Tipo de cambio, descripción breve, cuerpo opcional y notas de pie de página.
23. Solo se necesita una breve descripción del cambio.
24. Nombre de la rama, descripción detallada del cambio y fecha.
25. ¿Qué es el patrón de diseño Singleton y cómo se implementa en Java 8?
26. El patrón de diseño Singleton es un patrón que se utiliza para garantizar que una clase tenga una única instancia en todo el sistema. Se implementa utilizando una variable estática y un constructor privado.
27. El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de infraestructura en una aplicación. Se implementa utilizando excepciones y bloques try-catch.
28. El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de presentación en una aplicación. Se implementa con interfaces y clases concretas.
29. El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de negocios en una aplicación. Se implementa utilizando clases abstractas y métodos estáticos.
30. Seleccione la respuesta que considere correcta dado el siguiente bloque de código

Texto

Descripción generada automáticamente

1. 3.0
2. 1
3. 5
4. 2.5
5. ¿Qué es un Microservicio?
6. Ninguna de las anteriores
7. Es un componentes que se pueden desplegar de forma independiente en función múltiple, es decir, englobando endpoint que no necesariamente están relacionados.
8. Es el conjunto de endpoints contenidos en múltiples desarrollos que se despliegan en conjunto y que están estrechamente relacionados.
9. Es un componentes que se pueden desplegar de forma independiente y que suelen ser de función única, es decir, englobando endpoint y están estrechamente relacionandos.
10. ¿Cuál no es un objetivo de Maven?
11. Debug
12. Clean
13. Package
14. Install
15. En los verbos REST ¿Cuál es la diferencia en el uso de PATCH y PUT?
16. Son exactamente iguales, no hay diferencia de uso.
17. PATCH requiere se le envíe la entidad completa mientras que PUT solo los atributos a modificar.
18. PUT requiere se le envíe la entidad completa mientras que PATCH solo los atributos a modificar.
19. PATCH es un verbo deprecado sustituido por PUT.
20. ¿Qué es una expresión lambda en Java 8?
21. Una expresión lambda es una forma de escribir una clase anónima en Java 8.
22. Una expresión lambda es una forma de escribir una función anónima en Java 8.
23. Una expresión lambda es un método que se llama automáticamente cuando se crea un objeto.
24. Una expresión lambda es un método que se llama de forma explícita desde el código.
25. ¿Cuál es la diferencia entre una clase abstracta y una interfaz en Java 8?
26. Una interfaz solo puede heredar de una clase, mientras una clase abstracta puede heredar de múltiples interfaces
27. Una clase abstracta puede contener implementaciones de métodos, mientras que una interfaz no puede.
28. Una clase abstracta puede contener variables de instancia, mientras que una interfaz no puede.
29. Una interfaz puede contener implementaciones de métodos, mientras que una clase abstracta no puede.
30. ¿Cuál es la diferencia entre las anotaciones: @RestController, @Component, @Service y @Repository?
31. @Controller es una anotación que nos ayuda a construir una API REST mientras que @Service, @Component y @Repository solo marcan las clases que se deben de inicializar.
32. @Controles, @Component son anotaciones que crean bean y exponen la serialización de las clases mientras que @Service y @Repository requieren de una inicialización manual.
33. No existe diferencia funcional entre ellas sino semántica, las 4 son anotaciones de Spring que crean un bean y lo agregan al contexto de Spring.
34. @Service y @Repository son anotaciones que crear bean y exponen la serialización de las clases mientras que @Controller. @Component requiere de una inicialización manual.
35. ¿Cuál es una buena práctica al escribir pruebas unitarias?
36. Ejecutar pruebas con poca frecuencia.
37. Asegurarse de que las pruebas sean claras y concisas.
38. Probar solo una pequeña parte de una función.
39. Hacer que las pruebas dependan de otras pruebas.
40. ¿Cuál es la ventaja de usar APIs REST sobre otros tipos de servicios web?
41. Mayor seguridad.
42. Mayor facilidad de implementación.
43. Mayor velocidad de transferencia de datos.
44. Mayor compatibilidad con diferentes plataformas.
45. Selecciona la respuesta correcta con respecto al resultado del bloque de código.

Texto

Descripción generada automáticamente

1. No compila
2. [25, 7, 25, 67]

[67, 7, 25]

[7, 25, 67]

[67, 7, 25]

[7, 25, 67]

1. [25, 7, 67]  
   [67, 7, 25]  
   [7, 25, 67]  
   [67,7, 25]  
   [7, 25, 67]
2. [67, 7, 25]  
   [67, 7, 25]  
   [67, 7, 25]  
   [67, 7, 25]  
   [67, 7, 25]
3. [25, 7, 25, 67]  
   [7, 25, 67]  
   [67, 7, 25]  
   [7, 25, 67]  
   [67, 7, 25]
4. ¿Cuál es la salida al ejecutar el siguiente código?

Texto

Descripción generada automáticamente

1. 51tuna
2. 5tuna
3. 5
4. 41
5. 5 tuna
6. 4 1
7. El código no compila
8. ¿Cuáles son los 4 pilares de la programación orientada a objetos?
9. Polimorfismo, Coerción, Herencia y Encapsulamiento.
10. Encapsulamiento, Coerción, Polimorfismo y Abstracción.
11. Polimorfismo, Herencia, Encapsulamiento y Sincronía.
12. Polimorfismo, Abstracción, Herencia y Encapsulamiento.
13. ¿Cuál es el comando utilizado para ver el historial de cambios en git?
14. git diff
15. git status
16. git log
17. git commit
18. ¿Qué utilidad de línea de comandos basada en MS Windows le permitirá ejecutar el intérprete de Java sin abrir la ventana de la consola?
19. jconsole
20. javaw
21. interpw
22. java -wo
23. ¿Qué es un endpoint en unaAPI REST?
24. Un endpoint es un objeto que se utiliza para almacenar datos en una API REST.
25. Un endpoint es un método que se utiliza para procesar datos en una API REST.
26. Un endpoint es un controlador que se utiliza para administrar una API REST.
27. Un endpoint es la URL que se utiliza para acceder a una API REST.
28. ¿Cuál de lo siguiente es cierto?
29. java toma el nombre del archivo .bytecode como parámetro.
30. javac compila un archivo .java en un archivo .class
31. Java toma el nombre de la clase como parámetro.
32. javac compila un archivo .class en un archivo .java
33. javac compila un archivo .java en un archivo. bytecode
34. Java toma el nombre del archivo .class como parámetro.