

1.- ¿Qué es un “code smell”?

- a) Un componente de la biblioteca estándar de Java
- b) Un error en tiempo de ejecución que se produce en Java
- c) Un indicador de que puede haber un problema en el código que puede ser difícil de detectar o que podría ser una fuente potencial de errores o problemas de mantenimiento en el futuro
- d) Una práctica de programación recomendada en Java

2.- ¿Cuáles de las siguientes afirmaciones son verdaderas? (Elija todas las correctas)

- a) Las excepciones de tiempo de ejecución son lo mismo que las excepciones comprobadas
- b) Las excepciones en tiempo de ejecución son lo mismo que las excepciones no comprobadas
- c) Solo puede manejar subclases de excepción
- d) Solo puede declarar excepciones comprobadas (checked)
- e) Puede declarar solo excepciones no comprobadas (unchecked)

3.- Cuales son las salidas del siguiente código:

```
public class StringBuilders {  
    public static StringBuilder work(StringBuilder a,  
    StringBuilder b) {  
        b = new StringBuilder("a");  
        b.append("b");  
        return b;  
    }  
  
    public static void main(String[] args) {  
        StringBuilder s1 = new StringBuilder("s1");  
        StringBuilder s2 = new StringBuilder("s2");  
        StringBuilder s3 = work(s1, s2);  
        System.out.println("s1 = " + s1);  
        System.out.println("s2 = " + s2);  
        System.out.println("s3 = " + s3);  
    }  
}
```

- a) El código no compila
- b) s3 = a
- c) s1 = a
- d) s2 = s2
- e) s1 = s1
- f) s3 = null
- g) s2 = s2b

4.- ¿Cuáles de las siguientes opciones son verdaderas? (Elija todas las correctas)

- a) Java permite la sobrecarga de operadores
- b) Java es un lenguaje de programación funcional
- c) Java tiene punteros a ubicaciones específicas en la memoria
- d) Java es un lenguaje orientado a objetos
- e) El código Java compilado en Windows puede ejecutarse en Linux
- f) Java es un lenguaje procedimental

5.- Selecciona la respuesta correcta con respecto al resultado del bloque de código

```
public class Test1 extends Concreate{
    Test1(){
        System.out.println("t ");
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        new Test1();
    }
}
class Concreate extends Send {
    Concreate(){
        System.out.println("c ");
    }
    private Concreate(String s){

    }
}
abstract class Send{
    Send(){
        System.out.println("s ");
    }
}
```

- a) Error en tiempo de ejecución
- b) No compila
- c) t,s,c
- d) c,s,t

6.- ¿Cuál es la función del operador de doble dos puntos (::) en Java?

- a) El operador de doble dos puntos se utiliza para acceder a métodos no estáticos en Java 8
- b) El operador de doble dos puntos se utiliza para acceder a métodos estáticos en Java 8
- c) El operador de doble dos puntos no se utiliza en Java 8
- d) El operador de doble dos puntos se utiliza para crear una nueva instancia de una clase de Java 8

7.- ¿Qué significa el acrónimo CRUD en una API REST?

- a) Code, Register, Update, Debug
- b) Create, Read, Update, Delete
- c) Call, Recive, Use, Debug
- d) Customize, Request, Use, Debug

8.- ¿Qué es un bean de Spring?

- a) Un archivo de configuración XML que se utiliza para definir la estructura de una tabla de base de datos
- b) Una instancia de una clase que se administra por el contenedor de Spring
- c) Una herramienta de inyección de dependencias que se utiliza para inyectar dependencias en una clase
- d) Una clase que se utiliza para configurar la conexión a una base de datos

9.- Son patrones de diseño de software estructural

- a) Builder, Singleton, Prototype y Abstract Factory
- b) Adapter, Proxy, Prototype y Bridge
- c) Agile, Builder, Singleton y Prototype
- d) Adapter, Bridge, Proxy y Composite

10.- ¿Para qué nos sirve utilizar un profile dentro del archive pom.xml?

- a) Etiqueta por la cual podemos definir la versión de nuestras dependencias
- b) Es la etiqueta por la cual podemos definir las características que tendrá nuestro proyecto al ser compiladas
- c) Etiqueta por la cual definimos los parámetros de conexión a un repositorio
- d) No existe esta etiqueta en maven

11.- ¿Cuál es el comando utilizado para actualizar la rama local con los cambios de la rama en Git?

- a) git checkout
- b) git clone
- c) git push
- d) git pull

12.- Dadas las siguientes clases Vehicle y Car

```
package my.vehicles;

public class Vehicle {
    public String make;
    protected String model;
    private int year;
    int mileage;
}
```

```
package my.vehicles.cars;
import my.vehicles.*;

public class Car extends Vehicle {
    public Car() {
        //línea 7
    }
}
```

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

- a) `milage = 15285;`
- b) Ninguna de las anteriores
- c) `make = "Honda";`
- d) `year = 2009;`
- e) `model = "Pilot";`

13.- Enumere cuatro interfaces de la API de colecciones:

- a) `List, Map, Set, Queue`
- b) `ArrayList, Map, Set, Queue`
- c) `List, HashMap, HashSet, PriorityQueue`
- d) `List, Map, HashSet, PriorityQueue`

14.- De los siguientes, ¿Qué tipos de declaraciones se deben usar para contar la cantidad de monedas de cinco centavos en una matriz de cadenas de varias monedas?

- a) `Assertion`
- b) `Assignment`
- c) `Conditional`
- d) `Iteration`

15.- Selecciona la respuesta correcta con respecto al resultado del bloque de código

```
public class Tests {  
    public static void main(String args[]) {  
        Side primerIntento = new Head();  
        Side segundoIntento = new Tail();  
        Coin.overload(primerIntento);  
        Coin.overload(segundoIntento);  
        Coin.overload((Object)segundoIntento);  
        Coin.overload((Side)primerIntento);  
    }  
}  
  
interface Side { String getSide(); }  
  
class Head implements Side {  
    public String getSide() { return "Head "; }  
}  
  
class Tail implements Side {  
    public String getSide() { return "Tail "; }  
}  
  
class Coin {  
    public static void overload(Head side)  
    {System.out.println(side.getSide());}  
    public static void overload(Tail side)  
    {System.out.println(side.getSide());}  
    public static void overload(Object side) {System.out.println("Object  
");}  
}
```

- a) Head
Object
Tail
Side
- b) No compila
- c) Side
Object
Tail
Side

- d) Head
Head
Tail
Tail
- e) Side
Head
Tail
Side

16.- Cual es la salida al ejecutar el siguiente código

```
public class Lion {  
    public void roar(String roar1, StringBuilder roar2) {  
        roar1.concat("!!!");  
        roar2.append("!!!");  
    }  
  
    public static void main(String[] args) {  
        String roar1 = "roar";  
        StringBuilder roar2 = new StringBuilder("roar");  
        new Lion().roar(roar1, roar2);  
        System.out.println(roar1 + " " + roar2);  
    }  
}
```

- a) roar roar!!!
- b) roar!!! roar
- c) Se lanza una excepción
- d) roar!!! roar!!!
- e) roar roar
- f) El código no compila

17.- ¿Cuál de los siguientes es cierto acerca de una subclase concreta?

- a) Una subclase concreta no se puede marcar como final
- b) Una subclase concreta debe implementar todos los métodos definidos en una interfaz heredada
- c) Una subclase concreta debe implementar todos los métodos abstractos heredados
- d) Una subclase concreta puede declararse como abstracta
- e) Los métodos abstractos no pueden ser anulados por una subclase concreta

18.- ¿Cuál es la salida del siguiente código?

```
1  public abstract class Catchable {
2      protected abstract void catchAnObject(Object x);
3  }
4
5  public static void main(String [] args) {
6      java.util.Date now = new java.util.Date();
7      Catchable target = new MyStringCatcher();
8      target.catchAnObject(now);
9  }
10
11 class MyStringCatcher extends Catchable {
12     public void catchAnObject(Object x) {
13         System.out.println("Caught object");
14     }
15
16     public void catchAnObject(String s) {
17         System.out.println("Caught string");
18     }
19 }
```

- a) Error compilación línea 12
- b) Error compilación línea 16
- c) Caught string
- d) Error compilación línea 2
- e) Caught object

19.- Seleccione la respuesta que considere correcta, dado el siguiente bloque de código

```
import java.util.Arrays;
import java.util.List;

public class Example {

    public static void main(String[] args) {
        List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

        int result = numbers.stream()
                            .filter(n -> n % 2 == 0)
                            .reduce(0, (a,b) -> a + b);

        System.out.println(result);
    }
}
```

- a) 3
- b) 9
- c) 14
- d) 6

20.- ¿Qué declaración representa una declaración válida que permitirá la inclusión de la clase del paquete java.util?

- a) #include java.util.*;
- b) #include java.util;
- c) Import java.util.*;
- d) Import java.util;

21.- ¿Qué es la cobertura de código?

- a) La cantidad de veces que se ejecuta una línea de código
- b) La cantidad de errores detectados por una prueba
- c) La cantidad de tiempo que tarda una prueba en ejecutarse
- d) La cantidad de código que se ejecuta durante una prueba

22.- ¿Qué es Git?

- a) Una herramienta de control de versiones que se utiliza para almacenar y administrar el código fuente de un proyecto
- b) Una herramienta de gestión de dependencias que se utiliza para descargar bibliotecas y paquetes en un proyecto de Java
- c) Una herramienta de generación de informes que se utiliza para generar informes sobre el rendimiento de una aplicación

d) Una herramienta de automatización de compilación que se utiliza para compilar y construir un proyecto

23.- Which one do you like. ¿Cuál es el formato correcto para hacer un commit en Git?

- a) Descripción breve de cambio y nombre del autor
- b) Tipo de cambio, descripción breve, cuerpo opcional y notas de pie de pagina
- c) Solo se necesita una breve descripción del cambio
- d) Nombre de la rama, descripción detallada del cambio y fecha

24.- ¿Qué es el patrón de diseño Singleton y como se implementa en Java 8?

- a) 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 puede implementar en Java 8 utilizando una variable estática y un constructor privado
- b) El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de infraestructura en una aplicación. Se puede implementar en Java 8 utilizando excepciones y bloques try-catch
- c) El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de presentación en una aplicación. Se puede implementar en Java 8 utilizando interfaces y clases concretas
- d) El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de negocios en una aplicación. Se puede implementar en Java 8 utilizando clases abstractas y métodos estáticos

25.- Seleccione la respuesta que considere correcta, dado el siguiente bloque de código

```
import java.util.Arrays;
import java.util.List;

public class Example {

    public static void main(String[] args) {
        List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

        double result = numbers.stream()
            .mapToInt(n -> n)
            .average()
            .orElse(0);

        System.out.println(result);
    }
}
```

- a) 1
- b) 5
- c) 3.0
- d) 2.5

26.- ¿Qué es un Microservicio?

- a) Ninguna de las anteriores
- b) Es un componente que se pueden desplegar de forma independiente, de función múltiple es decir englobando endpoint que no necesariamente están relacionados.
- c) Es el conjunto de endpoints contenidos en múltiples desarrollos que se despliegan en conjunto y que están estrechamente relacionados
- d) Es un componente que se pueden desplegar de forma independiente y que suelen ser de función única es decir englobando endpoint que están estrechamente relacionados

27.- Cual no es un objetivo de maven (maven goal)

- a) debug
- b) Clean
- c) Package
- d) Install

28.- En los verbos REST, ¿cuál es la diferencia en el uso de PATCH y PUT?

- a) Son exactamente iguales, no hay diferencia en su uso
- b) PATCH requiere se le envíe la entidad completa mientras que PUT solo los atributos a modificar
- c) PUT requiere se le envíe la entidad completa mientras que PATCH solo los atributos a modificar
- d) PATCH es un verbo deprecado sustituido por PUT

29.- ¿Qué es una expresión lambda en Java 8? ¿Cómo se utiliza?

- a) Una expresión lambda es un método que se llama automáticamente cuando se crea un objeto
- b) Una expresión lambda es un método que se llama de forma explícita desde el código
- c) Una expresión lambda es una forma de escribir una clase anónima en Java 8
- d) Una expresión lambda es una forma de escribir una función anónima en Java 8

30.- ¿Cuál es la diferencia entre una clase abstracta y una interfaz en Java 8?

- a) Una clase abstracta puede contener variables de instancia mientras que una interfaz no puede
- b) Una interfaz puede contener implementaciones de métodos, mientras que una clase abstracta no puede
- c) Una interfaz solo puede heredar de una clase, mientras que una clase abstracta puede heredar de múltiples interfaces
- d) Una clase abstracta puede contener implementaciones de métodos, mientras que una interfaz no puede

31.- ¿Cuál es la diferencia entre las anotaciones: @RestController, @Component, @Service y @Repository?

- a) @Controller es una anotación que nos ayuda a construir un api rest mientras que @Service, @Component y @Repository solo marcan las clases que se deben de inicializar
- b) @Controller, @Component son anotaciones que crea bean y exponen la serialización de las clases mientras que @Service y @Repository requieren de una inicialización manual
- c) 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
- d) @Service y @Repository son anotaciones que crean bean y exponen la serialización de las clases mientras que @Controller, @Component requieren de una inicialización manual

32.- ¿Cuál es una buena práctica al escribir pruebas unitarias?

- a) Ejecutar pruebas con poca frecuencia
- b) Asegurarse de que las pruebas sean claras y concisas
- c) Probar solo una pequeña parte de una función
- d) Hacer que las pruebas dependan de otras pruebas

33.- ¿Cuál es la ventaja principal de usar API Rest sobre otros tipos de servicios web?

- a) Mayor seguridad
- b) Mayor facilidad de implementación
- c) Mayor velocidad de transferencia de datos
- d) Mayor compatibilidad con diferentes plataformas

34.- Selecciona la respuesta correcta con respecto al resultado del bloque de código

```
public class Test4 {  
  
    public static void main(String[] args) {  
        List list = Arrays.asList(25,7,25,67);  
        System.out.println(list);  
        System.out.println(new HashSet(list));  
        System.out.println(new TreeSet(list));  
        System.out.println(new HashSet(list));  
        System.out.println(new ConcurrentSkipListSet(list));  
    }  
}
```

- a) No compila
- b) [25,7,25,67]
[67,7,25]
[7,25,67]
[67,7,25]
[7,25,67]
- c) [25,7,67]
[67,7,25]
[7,25,67]
[67,7,25]
[7,25,67]
- d) [67,7,25]
[67,7,25]
[67,7,25]
[67,7,25]
[67,7,25]
- e) [25,7,25,67]
[7,25,67]
[67,7,25]
[7,25,67]
[67,7,25]

35.- Cual es la salida al ejecutar el siguiente código:

```
public class Fish {  
    public static void main(String[] args) {  
        int numFish = 4;  
        String fishType = "tuna";  
        String anotherFish = numFish + 1;  
        System.out.println(anotherFish + " " + fishType);  
        System.out.println(numFish + " " + 1);  
    }  
}
```

- a) 5
- b) 5 tuna
- c) 41
- d) 5tuna
- e) 51tuna
- f) 4 1
- g) El código no compila

36.- ¿Cuáles son los 4 pilares de la programación orientada a objetos?

- a) Polimorfismo, Coerción, Herencia y Encapsulamiento
- b) Encapsulamiento, Coerción, Polimorfismo y Abstracción
- c) Polimorfismo, Herencia, Encapsulamiento y Sincronia
- d) Polimorfismo, Abstracción, Herencia y Encapsulamiento

37.- Which one do you like? ¿Cuál es el comando utilizado para ver el historial de cambios en Git?

- a) git diff
- b) git status
- c) git log
- d) git commit

38.- ¿Qué utilidad de línea de comandos basada en MS Windows te permite ejecutar el interprete de Java sin abrir la ventana de la consola?

- a) jconsole
- b) javaw
- c) interpw
- d) java -wo

39.- ¿Qué es un endpoint en una API REST?

- a) Un endpoint es un objeto que se utiliza para almacenar datos en una API REST
- b) Un endpoint es un método que se utiliza para procesar datos en una API REST
- c) Un endpoint es un controlador que se utiliza para administrar una API REST
- d) Un endpoint es la URL que se utiliza para acceder a una API REST

40.- ¿Cuál de lo siguiente es cierto? (Elija todas las correctas)

- a) Java toma el nombre del archivo .bytecode como parámetro
- b) javac compila un archivo java en un archivo .class
- c) Java toma el nombre de la clase como parámetro
- d) javac compila un archivo .class en un archivo .java
- e) javac compila un archivo .java en un archivo .bytecode
- f) Java toma el nombre del archivo .class como parámetro