

Foundation Test

Name Taken on - 08 may, '25 04:25 PM
Correct Answers 69
Time Taken 02:08:29
Start Time 08 may 25 16:25

Status Passed 88%
Total Questions 78
Total Time 02:46:24
Finish/Pause Time 09 may 25 22:37

Simuladores

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
- h) s3 = ab

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
- e) s,c,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) mileage = 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

f) Object, Object, Object, Object

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
- f) Error de compilación. El método main no esta definido dentro de una clase.

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.- ¿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 pregunta parte de una función
- d) Hacer que las pruebas dependen 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 cuatro 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.- ¿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

Examen ASO A1

1.- ¿Cuál es el comando utilizado para deshacer el último commit en git?

- a) git reset
- b) git revert
- c) git amend
- d) git checkout

4.- ¿Qué es un archivo JAR en java?

- a) Un archivo que contiene un archivo de configuración Maven
- b) Un archivo que contiene un archivo de configuración Git.
- c) Un archivo que contiene una clase Java compilada
- d) Un archivo que contiene un archivo de configuración de Spring

5.- ¿Qué es la sobrecarga de métodos en Java?

- a) Cuando un método tiene múltiples definiciones con el mismo nombre y tipo de parámetros.
- b) Cuando un método tiene múltiples definiciones con diferentes nombres y cantidades de parámetros.
- c) Cuando un método tiene múltiples definiciones con diferentes tipos de cantidades de parámetros.
- d) Cuando un método tiene múltiples definiciones con diferentes nombres y tipos de parámetros.

6.- ¿Cuál es la diferencia entre un ArrayList y un LinkedList en Java?

- a) ArrayList es más rápido que un LinkedList para agregar y eliminar elementos.
- b) ArrayList es más eficiente en el uso de memoria que LinkedList.
- c) LinkedList es una clase abstracta mientras que ArrayList es una clase concreta.
- d) LinkedList es más rápido que ArrayList para acceder a elementos aleatorios.

7.- ¿Cuándo se debe usar un bloque finally en una declaración try regular (no un try-with-resources)?

- a) Cuando no hay bloques catch en una declaración try.
- b) Nunca.
- c) Cuando hay dos o más bloques catch en una sentencia try.
- d) Cuando hay exactamente un bloque catch en una sentencia try.
- e) Cuando el código del programa no termina por sí solo.

8.- ¿Cuál es el propósito principal de los test unitarios?

- a) Comprobar la eficiencia del hardware.
- b) Medir la velocidad de la aplicación.
- c) Ahorrar tiempo en el desarrollo.
- d) Asegurar la calidad del software.

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

```
public class Test3 {  
    public static void main(String[] args) {  
        String cad1 = "hola";  
        String cad2 = new String( original: "hola");  
        String cad3 = "hola";  
  
        if (cad1 == cad2)  
            System.out.println("ca1 es igual a cad2");  
        else System.out.println("cad1 diferente a cad2");  
  
        if (cad1 == cad3)  
            System.out.println("cad1 es igual a cad3");  
        else  
            System.out.println("cad1 diferente a cad3");  
    }  
}
```

- a) cad1 diferente a cad2
cad1 es igual a cad3
- b) ca1 es igual a cad2
ca1 es igual a cad3
- c) No compila
- d) cad1 diferente a cad2
cad1 diferente a cad3

10.- ¿Cuál es la salida al ejecutar el siguiente código?

```
1: class Mammal {
2:     public Mammal(int age) {
3:         System.out.print("Mammal");
4:     }
5: }
6: public class Platypus extends Mammal {
7:     public Platypus() {
8:         System.out.print("Platypus");
9:     }
10:     public static void main(String[] args) {
11:         new Mammal(5);
12:     }
13: }
```

- a) Mammal.
- b) MammalPlatypus.
- c) El código no se compila en la línea 11.
- d) El código no compila en la línea 7, ya que no se hace una llamada correcta al constructor de Mammal en la clase Platypus.

11.- ¿Cómo se manejan las excepciones en java?

- a) Con la instrucción try-catch.
- b) Con la instrucción if-else.
- c) Con la instrucción for.
- d) Con la instrucción while.

12.- ¿La anotación @Ignore es usada para omitir un test por lo que no se ejecuta?

- a) Verdadero
- b) Falso

13.- ¿Cuál es el resultado de compilar y ejecutar el siguiente código?

```
public class Tester {
    static {
        int x = 3;
    }
    2 usages
    static int x;
    public static void main(String[] args) {
        x--; // line 7
        System.out.println(x);
    }
}
```

- a) Error de compilación en la línea 7, x no se inicializa.
- b) -1
- c) -2
- d) 0

14.- ¿Qué es un operador de short circuit?

- a) Es un patrón de arquitectura de microservicios que nos permite evitar el consumo de servicios que están...
- b) Sirve para realizar más eficientes las operaciones condicionales evitando realizar operaciones si están...
- c) Sirve para lanzar una excepción personalizada.
- d) Operador que nos sirve para crear una nueva clase anónima.

15.- ¿Qué es el patrón de diseño DAO y cómo se implementa en Java?

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

19.- ¿Qué es Maven y para qué se utiliza en el desarrollo de aplicaciones?

- a) Maven es un lenguaje de programación. Se utiliza en el desarrollo de aplicaciones Java para escribir...
- b) Maven es un servidor de base de datos. Se utiliza en el desarrollo de aplicaciones java para alojar...
- c) Maven es un sistema de control de versiones. Se utiliza en el desarrollo de aplicaciones java para...
- d) Maven es una herramienta de gestión de dependencias. Se utiliza en el desarrollo de aplicaciones... (no se ve) en el proyecto.

21.- ¿Qué es Git y cuáles son algunos de sus comandos básicos?

- a) Git es un lenguaje de programación. Algunos comandos básicos de Git incluyen "print" e "if-else".
- b) Git es una herramienta para el análisis de código. Algunos comandos básicos de Git incluyen "analyze"...
- c) Git es un sistema de control de versiones. Algunos comandos básicos de Git incluyen "commit" y "push".
- d) Git es una herramienta para realizar pruebas de software. Algunos comandos básicos de Git incluyen...

22.- Dados los siguientes segmentos de código, ¿Qué respuesta no es una implementación de java válida?

- a) `int variableA = 10;`
`float variableB = 10.5f;`
`int variable = variableA + variableB;`
- b) `byte variableA = 10;`
`double variableB = 10.5f;`
`double variableC = variableA + variable;`
- c) `byte variableA = 10;`
`float variableB = 10.5f;`
`float variable = variableA + variable;`

23.- ¿Qué escenario es el mejor uso de una excepción?

- a) La computadora se incendió.
- b) No sabe cómo codificar un método.
- c) No se encuentra un elemento al buscar en una lista.
- d) **Se pasa un parámetro inesperado a un método.**
- e) Quiere recorrer una lista.

26.- ¿Cuáles de las siguientes afirmaciones sobre el polimorfismo son verdaderas? (Elija todas las correctas).

- a) **Un método que toma un parámetro con tipo `java.lang.Object` tomará cualquier referencia.**
- b) **Una referencia a un objeto se puede convertir a una subclase del objeto sin una conversión explícita...**
- c) Todas las excepciones de conversión se pueden detectar en tiempo de compilación.
- d) Al definir un método de instancia pública en la superclase, garantiza que el método específico se...
- e) **Si un método toma una superclase de tres objetos, cualquiera de esas clases puede pasarse...**

29.- ¿Qué son las pruebas de integración?

- a) Pruebas que comprueben el rendimiento de la aplicación.
- b) Pruebas que comprueben el funcionamiento de la interfaz de usuario.
- c) **Pruebas que comprueban el funcionamiento de varias unidades de código juntas.**
- d) Pruebas que comprueban el funcionamiento de una sola unidad de código.

30.- ¿Qué comando se utiliza para enviar los cambios confirmados en un repositorio local al repositorio remoto?

- a) **git push**
- b) it pull
- c) git commit
- d) git add

32.- ¿Cuál es el comando utilizado para crear una nueva rama en Git?

- a) git branch
- b) git merge
- c) git commit
- d) git push

33.- ¿Cuál es el resultado de compilar la siguiente clase?

```
public class Book {  
    3 usages  
    private int ISBN;  
    private String title, author;  
    private int pageCount;  
  
    public int hashCode(){  
        return ISBN;  
    }  
  
    public boolean equals(Object obj){  
        if(!(obj instanceof Book)){  
            return false;  
        }  
        Book other = (Book) obj;  
        return this.ISBN == other.ISBN;  
    }  
}
```

- a) Línea 15 no compila porque other.ISBN es un atributo con modificador de acceso private.
- b) Línea 14 no compila porque no está declarada o manejada ClassCastException
- c) La clase compila satisfactoriamente.

34.- ¿Cuál es la primera línea en fallar al compilar?

```
class Tool {  
    private void repair(){} //r1  
    1 override  
    void use(){}  
}  
  
class Hammer extends Tool{  
    private int repair(){return 0;} //r3  
    private void use(){} //r4  
    public void bang(){} //r5  
}
```

- a) r5
- b) r4**
- c) r3
- d) Ninguna de las anteriores.

36.- ¿Cuál de las siguientes excepciones lanza la JVM? (Elija todas las correctas)

- a) ArrayIndexOutOfBoundsException**
- b) NumberFormatException
- c) ExceptionInInitializerError**
- d) Java.io.IOException
- e) NullPointerException**

38.- ¿Cuál es el comando utilizado para fusionar una rama en Git?

- a) git Branch
- b) git merge**
- c) git push
- d) git pull

39.- ¿Qué es REST y cuál es su relación con las API web?

- a) REST es un protocolo de comunicación. Su relación con las API web es que se utiliza para definir los elementos...
- b) REST es un estilo de arquitectura para aplicaciones web. Su relación con las API web es que se utiliza...**
- c) REST es un servicio en la nube. Su relación con las API web es que se utiliza para alojar las aplicaciones...
- d) REST es un lenguaje de programación. Su relación con las API web es que se utiliza para crear... no se ve.

Examen ASO A2

2.- Dado el siguiente código:

```
public class Main {  
    public static void main(String[] args) {  
        int[] numeros = {1,2,3,4,5};  
        int suma = 0;  
        for (int i = 1; i <= numeros.length; i++){  
            suma += numeros[i];  
        }  
  
        System.out.println("La suma de los números es: " + suma);  
    }  
}
```

¿Este código compila sin errores?

- a) Si, compila sin errores. Pero arroja una excepción.
- b) No, hay un error en el ciclo for.
- c) No, hay un error en la inicialización de la variable “suma”.
- d) No, hay un error en la declaración del arreglo.

3.- ¿Qué método se utiliza para obtener el mensaje de una excepción en Java?

- a) getClass()
- b) printStackTrace()
- c) toString()
- d) getMessage()

5.- ¿Cuál es el resultado de ejecutar el siguiente código?

```
String s = "hello";  
s.toUpperCase();  
System.out.println(s);
```

- a) NullPointerException
- b) hello
- c) HELLO
- d) Hello

8.- ¿Cuál es el paquete de importación necesario para usar la clase ArrayList?

- a) import.java.net.*;
- b) import.java.awt.*;
- c) import.java.io.*;
- d) import.java.util.*;

9.- ¿Cuál es el formato de los datos que se envían y reciben en una API REST?

- a) YAML
- b) XML
- c) JSON
- d) Todos los anteriores

11.- ¿Qué palabra clave se utiliza para definir una excepción personalizada en Java?

- a) try
- b) throw
- c) finally
- d) catch

13.- ¿Cuál de los siguientes comandos elimina el directorio target antes de iniciar el proceso de construcción?

- a) mvn site
- b) mvn build

- c) mvn answer
- d) mvn clean

17.- ¿Qué muestra el siguiente código fuente por pantalla?

```
int x = 1;
switch (x){
    case 1:
        System.out.println("Uno");
    case 2:
        System.out.println("Dos");
    case 3:
        System.out.println("Tres");
    default:
        System.out.println("Otro número");
}
```

- a) Dos
- b) Uno Dos Tres Otro número
- c) Uno
- d) Otro número

20.- De los siguientes paquetes, ¿cuáles contienen clases para construir una interfaz gráfica? (Elije todas las que correspondan)

- a) java.net
- b) java.io
- c) javax.swing
- d) java.util
- e) java.awt

23.- ¿Cuál de las siguientes líneas deben ir en el espacio en blanco para que el código compile?

```
public class News < ____> { }
```

- a) Solo N
- b) Solo ?
- c) Ninguna de las anteriores
- d) News,y Object
- e) ? y N
- f) N, News y Object

25.- ¿Qué es un stream en Java 8 y para qué se utiliza?

- a) Un objeto que representa una conexión de entrada o salida de datos.
- b) Un objeto que representa una secuencia de elementos y se utiliza para procesar colecciones de forma declarativa.
- c) Un objeto que se utiliza para leer y escribir archivos de texto.
- d) Un objeto que se utiliza para crear y manipular bases de datos.

27.- Son patrones de diseño de microservicios

- a) Circuit Breaker, Adaptative Lifo, MQ Strategy
- b) System, Process y Client
- c) Retry, Circuit Breaker, Adaptative Lifo y Bulkhead
- d) Ninguna de las anteriores

30.- ¿Qué afirmaciones son verdaderas tanto para las clases abstractas como para las interfaces? (Elije todas las correctas)

- a) Ambos pueden contener métodos estáticos.
- b) Ambos se pueden ampliar con la clave extend.
- c) Ambos pueden contener métodos predeterminados.
- d) Ambos heredan de java.lang.Object.
- e) Ninguno de los dos puede ser instanciado directamente.
- f) Ambos pueden contener variables finales estáticas públicas.
- g) Supone que todos los métodos dentro de ellos son abstractos.

32.- ¿Si deseas obtener una copia de un repositorio Git existente en un servidor qué comando se utiliza?

- a) git commit
- b) git log
- c) git clone
- d) git add

34.- ¿Qué es un repositorio remoto en Git?

- a) Una herramienta que se utiliza para compartir y fusionar cambios entre diferentes ramas de un repositorio.
- b) Una copia local de un repositorio que se utiliza para hacer cambios en el código fuente.
- c) Un servidor Git que almacena una copia central del repositorio.
- d) Un archivo que contiene una instantánea del código fuente en un momento determinado.

36.- Dada la siguiente clase

```
public class Helper {  
    public static < U extends Exception > void  
        printEception(U u){  
        System.out.println(u.getMessage());  
    }  
  
    public static void main(String[] args) {  
        //línea 9  
    }  
}
```

¿Cuál de las siguientes instrucciones puede colocarse en la línea 9 para que la clase Helper compile?

- a) Helper.printException(new Exception("B"));
- b) Helper.printException(new FileNotFoundException("A"));

- c) Helper.<Throwable>printException(new Exception("C"));
- d) Helper.<NullPointerException>printException(new NullPointerException("D"));
- e) Helper.printException(new Throwable("E"));

40.- ¿A qué hace referencia el principio de Liskov?

- a) Nos indica que una clase no debe tener solo una funcionalidad sino varias para reducir el uso de objetos.
- b) Este principio nos indica que dentro del programa una clase puede ser sustituida por cualquier clase que se extienda de ella sin alterar el comportamiento del programa.
- c) Nos indica que cualquier clase se puede extender para agregar funcionalidad, pero no se puede modificar.
- d) Este principio nos indica que dentro del programa una clase puede ser sustituida por su clase padre sin alterar el comportamiento del programa.

Examen ASO B

1.- ¿Cuál es el valor de "x" y "y" al final el programa?

```
int x = 0;
do {
    System.out.println(x);
    x++;
} while (x < 10);
int y = 0;
while (y < 10) {
    System.out.println(y);
    y++;
}
```

- a) x=9 y=10
- b) x=10 y=9
- c) x=10 y=10
- d) x=9 y=9

2.- ¿Dado el siguiente enum y clase cuál es la opción que puede ir en el espacio en blanco para que el código compile?

```
enum Season { SPRING, SUMMER, WINTER }
public class Weather {
    public int getAverageTemperate(Season s) {
        switch (s) {
            default:
                _____ return 30;
        }
    }
}
```

- a) Ninguno de los anteriores
- b) case SUMMER ->
- c) case Season.Winter:

- d) case FALL:
- e) case Winter, Spring:
- f) case SUMMER | WINTER:

19.- ¿Qué declaración representa una declaración válida que permitirá la inclusión de clases del paquete Java.util?

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

20.- ¿Resultado de compilar y ejecutar el siguiente programa?

```
public static void main(String[] args) {  
    boolean stmt1 = "champ" == "champ";  
    boolean stmt2 = new String("champ") == "champ";  
    boolean stmt3 = new String("champ") == new String("champ"); System.out.println(stmt1 && stmt2 || stmt3);  
}
```

- a) **False**
- b) no se produce salida
- c) true
- d) error de compilación

25.- ¿Cómo se manejan las excepciones en Java?

- a) Las excepciones se manejan con bloques switch case en Java. La excepción try -with-resources es una forma de lanzar una excepción en un método.
- b) Las excepciones se manejan con bloques while en Java. La excepción try -with-resources es una forma de manejar excepciones de compilación
- c) Las excepciones se manejan con bloques if else en Java. La excepción try -with-resources es una forma de manejar las excepciones en tiempo de ejecución.
- d) **Las excepciones se manejan con bloques try catch finally en Java. La excepción try -with-resources es una forma de cerrar automáticamente los recursos abiertos en un bloque try.**

29.- ¿Qué clase del paquete java.io permite leer y escribir archivos en ubicaciones específicas dentro de un archivo?

- a) File
- b) FilenameFilter
- c) FileDescriptor
- d) **RandomAccessFile**

30.- Dadas las siguientes definiciones de clases my.school.Classroom y my.city.School ¿Qué números de línea en el método main generan un error de compilación? (Elija todas las opciones correctas)

```
1: package my.school;
2: public class Classroom {
3:     private int roomNumber;
4:     protected String teacherName;
5:     static int globalKey = 54321;
6:     public int floor = 3;
7:     Classroom(int r, String t) {
8:         roomNumber = r;
9:         teacherName = t; } }

1: package my.city;
2: import my.school.*;
3: public class School {
4:     public static void main(String[] args) {
5:         System.out.println(Classroom.globalKey);
6:         Classroom room = new Classroom(101, "Mrs. Anderson");
7:         System.out.println(room.roomNumber);
8:         System.out.println(room.floor);
9:         System.out.println(room.teacherName); } }
```

- a) Ninguna, el código compila bien
- b) línea 6
- c) línea 9
- d) línea 7
- e) línea 8
- f) línea 5

34.- ¿Qué hace el siguiente programa?

```
String sPalabra = "palabra";
int inc = 0;
int des = sPalabra.length() - 1;
boolean bError = false;
while ((inc < des) && (!bError)) {
    if (sPalabra.charAt(inc) == sPalabra.charAt(des)) {
        inc++;
        des--;
    } else {
        bError = true;
    }
}
```

- a) Calcula si una palabra es un palíndromo
- b) Cuenta las letras que hay en una palabra
- c) Calcula si hay letras repetidas en una palabra
- d) El programa no compila

38.- ¿Qué es una expresión lambda en Java?

- a) Una instancia de una clase que implementa una interfaz funcional
- b) Una instancia de una clase abstracta que se utiliza para implementar métodos anónimos

- c) Una forma concisa de representar una función anónima que se puede pasar como argumento
- d) Un método que no tiene cuerpo

39.- ¿Qué hace el siguiente código fuente?

```
int x = 0;
boolean flag = false;
while ((x < 10) || !flag) {
    System.out.println(x);
    x++;
}
```

- a) Muestra los números del 1 al 10
- b) Muestra un 10
- c) Se queda en un bucle infinito
- d) Muestra los números del 0 al 9