Archivos Examen 1

Question 1: 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 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 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);
       }
}
```

- 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]TUUAA[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 26: ¿Cuál es la función 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: ¿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. 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) + " "); //false
            System.out.println(str3.equals(str4) + " "); //true
            System.out.println(str3 == str5 + " "); //false
            System.out.println(str4 == str5); //false
        }
}
```

- 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

Answer 33: b. false true false false

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 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. Librerias
- **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
}
```

content_copy Use code with caution.Java

- **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

Archivos Examen 2

Pregunta 9:

Es el nombre del estado en el que se encuentra un hilo, antes de invocar el método Thread.start()

a) Nuevo (new)

b) Ejecutable (runnable)

c) Muerto (dead)

d) En ejecución (running)

Pregunta 10:

Son los tipos de clases internas en Java.

a) Abstracta, estática, heredada, anónima

b) Anónima, pública estándar estática

c) Pública, nativa, heredada, anónima

d) Regular, estática, local a un método, anónima

Pregunta 11:

Son consideradas clases internas, las cuales no tienen ninguna relación especial con su clase externa.

a) Abstractas

b) Anónimas

c) Estáticas

d) Regulares

Pregunta 12:

Se refiere a la declaración de clases dentro de otra.

a) Clases abstractas

b) Clases heredadas

c) Clases Internas

d) Ninguna de las anteriores

Pregunta 13:

Es la unidad de tiempo empleada en el argumento al método estático Thread.sleep().

a) Hora

b) Minuto

c) Segundo

d) Milisegundo

Pregunta 14:

Es un tipo de colección, en la cual, existe un mapeo entre una llave única (id) a un valor específico. Ambos, valor y llave son objetos.

a) Mapa (Map)

b) Árbol (Tree)

c) Lista (List)

d) Conjunto (Set)

Pregunta 15:

Una clase interna no-estática.

a) Tiene acceso a todos los miembros de su clase externa.

b) No tiene acceso a los miembros de su clase externa.

c) Tiene acceso solo a los miembros públicos de su clase externa.

d) Tiene acceso a todos los miembros de su clase externa, con excepción de los privados.

Pregunta 16:

El método String.split regresa.

a) Un arreglo de cadenas

b) Una lista de cadenas

c) Un mapa de cadenas

d) Ninguno de los anteriores

Pregunta 17:

Es la única manera de acceder a una clase interna regular.

a) A través de la línea de comandos, empleando el comando java, seguido del nombre de la clase interna.

b) Directamente en tiempo de ejecución, mediante una referencia a un objeto de su tipo.

c) Indirectamente en tiempo de ejecución, mediante una referencia a un objeto del tipo de clase que la contiene.

d) Ninguna de las anteriores.

Pregunta 18:

Es la sintaxis empleada en Java, para declarar una colección que solo acepta objetos de un tipo en particular.

a) Genérico

b) Especificador

c) Modificador

d) Limitante

Pregunta 19:

El tipo parametrizado, en la sintaxis de un genérico se escribe:

a) Entre paréntesis

b) Entre paréntesis angulares < & >

c) Entre corchetes

d) Entre llaves

Pregunta 20:

Es el modificador del lenguaje Java, con el cual se marca una variable de instancia, la cual no se desea incluir en la serialización de una clase.

a) static

b) volatile

c) remote

d) transient

Pregunta 21:

Al emplear el operador de incremento (++) en una variable declarada con el modificador final.

a) La variable conserva su valor original.

b) La variable es preincrementada.

c) La variable es postincrementada.

d) Se produce un error en tiempo de compilación.

Pregunta 22:

Una clase Java soporta:

a) Herencia simple

b) Herencia multiple

c) Herencia compuesta

d) Ninguna de las anteriores

Pregunta 23:

Es el nombre del método que es invocado de manera implícita en un objeto, cuando se le pasa una referencia al mismo al método System.out.println:

a) print

b) toChar

c) toString

d) hashCode

Pregunta 24:

La invocación de métodos de manera polimórfica aplica solo para:

a) Métodos estáticos

b) Métodos de instancia

c) Variables de instancia

d) Métodos marcados con el modificador native

Pregunta 25:

Es el nombre del hilo, en el que corre el método main, de un programa Java.

a) main

b) central

c) head

d) prime

Pregunta 26:

En la práctica, son los modificadores de acceso que se emplean en las variables de instancia de una clase, para controlar los datos asignables a las mismas.

a) final, private

b) static, transient

c) public, private

d) private, protected

Pregunta 27:

Una excepción en Java es un error:

a) En tiempo de ejecución

b) En tiempo de compilación

c) En la sintaxis del código

d) Ninguna de las anteriores

Pregunta 28:

Son clases en Java, que a diferencia de los arreglos, pueden expandirse o contraerse dinámicamente, conforme se les agregan o restan elementos.

a) Expansores

b) Reductores

c) Cadenas

d) Colecciones

Pregunta 29:

Un objeto de tipo String se caracteriza por ser:

a) Inmutable

b) Un tipo primitivo

c) Mutable

d) Polimórfico

Pregunta 30:

Se refiere al mecanismo que permite al programador Java, probar suposiciones durante la fase de desarrollo, sin tener que declarar el código a probar dentro de un bloque try.

a) Bloque

b) Comentario

c) Contención (contention)

d) Asención (assertion)

Pregunta 31:

Es el estado en el que se encuentra un hilo, cuando se encuentra “esperando” la disponibilidad de un recurso.

a) Durmiendo (sleep)

b) Bloqueado (blocked)

c) Muerto (dead)

d) Ninguno de los anteriores

Pregunta 32:

Los operadores lógicos de corto circuito (short-circuit) evalúan:

a) Siempre ambos lados de la expresión lógica

b) Condicionalmente el lado derecho de la expresión lógica

c) Únicamente el lado izquierdo de la expresión lógica

d) Únicamente el lado derecho de la expresión lógica

Pregunta 33:

Es el componente de la máquina virtual de Java, que coordina la ejecución de varios hilos.

a) Controlador (controller)

b) Pila (stack)

c) Programador de hilos (thread scheduler)

d) Memoria

Pregunta 34:

Se refiere a los diseños orientados a objetos, los cuales ya han sido probados y garantizan la reducción de potenciales fallas en el código.

a) Patrones de diseño

b) Patrones de arquitectura

c) Patrones de comportamiento

d) Ninguno de los anteriores

Pregunta 35:

Al comparar caracteres, Java emplea el valor:

a) Unicode de los caracteres

b) ANSI de los caracteres

c) UTF-8 de los caracteres

d) Ninguno de los anteriores

Pregunta 36:

Al terminar de escribir datos a un flujo de salida (output stream), se emplea este método para garantizar que todos los datos en el flujo, sean escritos al archivo asociado

a) Unload

b) Discharge

c) Empty

d) Flush

Pregunta 37:

Es el mecanismo nativo de Java a través del cual, el estado de un objeto puede ser guardado y posteriormente recuperado

a) Contención

b) Serialización

c) Almacenamiento

d) Registro

Pregunta 38:

En el API I/O de Java, se encuentra definida como una clase, la cual es una representación abstracta de la ruta de un archivo o directorio

a) Directory

b) Pathname

c) File

d) FilePath

Pregunta 39:

Es un tipo de colección, la cual, no acepta elementos duplicados, para lo cual, emplea el método equals

a) Mapa (Map)

b) Árbol (Tree)

c) Lista (List)

d) Conjunto (Set)

Pregunta 40:

Selecciona una respuesta

```
class test {

public static void main (String [] blah )

{

System.out.printf("%s", new test());

}

public String toString()

{

return "testing something";

}

}
```

a) Da un runtime exception

b) Imprime testing1234 o algo como eso

c) Compila con error

d) Imprime testing something

Pregunta 41:

¿Qué va a ser impreso si se intenta compilar y ejecutar el siguiente código?

```
int i=0;

switch (i) {

default:

System.out.println(“default”);

case 0:

System.out.println(“cero”);

break;

}
```

a) default

b) cero

c) da error de compilación

d) nada

Pregunta 42:

¿Cuál es el valor de funcionRetornoControlador?

```
var funcionRetornoControlador;

traerArchivo0;

function traerArchivo0{

leerArchivoServidor(‘PruebaParamsAJAXUltraAvanzado.jsp’, recibeArchiv0);

}

function recibirArchivoTexto(texto){

document.getElementById('divContenido').innerHTML = texto;

}

function leerArchivoServidor(archivo, funcionRetorno){
```

funcionRetornoControlador = funcionRetorno;

funcionRetorno(leerArchivoServidor(archivo));

}

a) funcionRetorno

b) leerArchivoServidor

c) traerArchivo

d) recibirArchivo

Pregunta 43:

¿Qué sucederá cuando compiles y ejecutes el siguiente código?

public class MyClass{

static int i;

public static void main(String arg[]){

System.out.println(i);

}

}

a) Al null

b) 1

c) 0

d) Error: Variable i may not have been initialized

Pregunta 44:

¿Cuál será el resultado cuando se intenta compilar y ejecutar el siguiente código?

public class Conv{

Conv c=new Conv();

String s=new String("ello");

```
c.amethod(s);

public void amethod(String s1){

char c='H';

s1=c+s1;

System.out.println(s1);

}

}
```

a) La compilación y generación de la cadena "hello"

b) La compilación y generación de la cadena 'H ello'

c) La compilación y generación de la cadena 'helloH'

d) Compila y genera error en tiempo de ejecución

Pregunta 45:

Selecciona una respuesta

```
int [] iarr= new int[3];

String [] sarr={"a","b","c"};

for(String s: sarr)

System.out.println(s);
```

a) Imprime las variables

b) Genera un error de excepción de tipo NULL

c) Genera un error de sintaxis.

d) Genera una excepción

Pregunta 46:

¿Qué línea imprime FALSE?

```
Integer eye = new Integer(42);

Double d = new Double(42.0);

int i = 42;

double dd = 42.0;

System.out.println(eye==eye); //1

System.out.println(eye.equals(d)); //2

System.out.println(eye == 42); //3

System.out.println(eye.intValue() == dd); //4

System.out.println(i == dd); //5
```

a) 3

b) 1

c) 4

d) 2

Pregunta 47:

¿Qué código colocado después del comentario //For loop podrá llenarse elementos el arreglo al recorrer la variable i ?

```
public class Linl{

public static void main(String arg[]){

int i[] =new int[4];

for(int i=0;i < 5; i++ )

{

//for loop podrá llenarse elementos el arreglo al recorrer la variable i

}

}
```

a) i=0

b) i[i] = 2

c) i = i+1

d) i[i] = i;

e) error de variables i duplicadas

Pregunta 48:

¿Qué línea imprime FALSE?

Integer eye = new Integer(42);

Double d = new Double(42.0);

int i = 42;

double dd = 42.0;

System.out.println(eye==eye); //1

System.out.println(eye.equals(d)); //2

System.out.println(eye == 42); //3

System.out.println(eye.intValue() == dd); //4

System.out.println(i == dd); //5

a) 3

b) 1

c) 4

d) 2

Pregunta 48:

¿Qué línea es la salida?

class test {

test0 {

try {

```
throw new RuntimeException();
}
finally {
System.out.println("Damn it");
}
public static void main(String arg[]){
try {
new test();
}
catch(Throwable t) {
System.out.println("Caught");
}
}
```

a) Compila con error: incorrect syntax

b) Ninguno de los anteriores

c) Error Runtime

d) Compila con error: incompatible types

Pregunta 49:

Selecciona una respuesta

```
int [] iarr= new int[]{1,2,3,4};
String [] sarr=Arrays.toString(iarr);
for( String s: sarr){
System.out.println(s);
}
```

a) Imprime 1 2 3 4

b) Error Runtime

c) Ninguno de los anteriores

d) Compila con error

Question 6:

"Instrucción: Relaciona las columnas entre sí en tu hoja de respuestas."

The pairs are:

- 00. (C) Pruebas Unitarias --> A) Validar que los componentes desarrollados se ensamblen de forma adecuada con la aplicación.
- 00. (A) Pruebas de Ensamblaje --> B) Este tipo de pruebas se llevan a cabo en un ambiente o entorno previo; incluye pruebas Funcionales, de Integración, de Regresión y de Excepción.
- 00. (B) Pruebas de Sistema --> C) Validar que los componentes que forman parte del sistema funcionan correctamente y cumplen con los requisitos de manera independiente.
- 00. (D) Pruebas Funcionales --> D) Validar los requerimientos de negocio (lo que se supone que el sistema debe hacer); pretenden validar que el sistema construido hace lo que razonablemente se espera de él.

Question 7:

"Instrucción: Relaciona las columnas entre sí en tu hoja de respuestas."

The pairs are:

- 00. (A) Pruebas Negativas o de Excepción --> A) Destinadas a mostrar que un componente o sistema no funciona.
- 00. (C) Pruebas Técnicas (Estructurales) --> B) Pruebas finales ejecutadas por Socio de Negocio y/o Usuario para asegurar que el sistema satisfaga las necesidades de la organización y usuario final, contando con la aceptación formal de que el sistema construido es el solicitado.
- 00. (B) Pruebas de Aceptación --> C) Incluyen un conjunto de categoría de prueba como: stress, volumen, seguridad, estándares; a fin de verificar que todas las partes del

sistema funcionan en sincronía y que la tecnología y arquitectura están siendo usadas adecuadamente.

- 00. (D) Pruebas Unitarias --> D) Validar que los componentes que forman parte del sistema funcionan correctamente y cumplen con los requisitos de manera independiente.

47. Qué código colocado después del comentario //For loop podrá llenarle elementos al arreglo al con los valores de la variable i?

```
 public class Lini{

public static void main(String args[]){

 Lini Lin = new Lini();

 Lamethod();

}

public void amethod() {

  int []a = new int[4];

  int initial=0;

  //Start For loop


 System.out.println(a[i]);


 }
```

A) for (int i=0; i <a.length(); i++)

B) for (int i=0; i<a.length()-1; i++)

C) for (int i=1; i<4; i++)

D) for (int i=0; i<a.length(i++);

E) Sin respuesta

48. ¿Qué línea es la salida?

```
    class test {
  test() {
    try{
      throw new RuntimeException();


     } finally {
     System.out.println("Damn F");
     }
  }
public static void main(String args[]) {
    try {
      new test();
    } catch (Throwable t) {
     System.out.println("Caught");
    }
  }
}
```

A) Compila con error. incorrect syntax

B) Ninguno de los anteriores

C) Error Runtime

D) Compila con error. Incompatible types

E) Sin respuesta

49. Selecciona una respuesta:

```
    int[] arr = {1, 2, 3, 4};
```

```
int [] arr2 = new int[4];

arr2=arr;

System.out.println(arr2[4]);
```

A) Imprime 4

B) Runtime Exception. ArrayOutofBounds

C) Compila con warnings

D) Compila con error

E) Sin respuesta