

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 Link{
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;**

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 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