

1. 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)

2. 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

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

- a) Abstractas
- b) Anónimas
- c) Estáticas
- d) Regulares

4. Se refiere a la declaración de clases dentro de otra.

- a) Clases abstractas

b) Clases heredadas

c) Clases Internas

d) Ninguna de las anteriores

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

a) Hora

b) Minuto

c) Segundo

d) Milisegundo

6. 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)

7. 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.

8. 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

9. 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.

10. 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

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

- a) Entre paréntesis
- b) Entre paréntesis angulares < & >
- c) Entre corchetes
- d) Entre llaves

12. 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

13. 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.

14. Una clase Java soporta:

a) Herencia simple

b) Herencia multiple

c) Herencia compuesta

d) Ninguna de las anteriores

15. 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`

16. 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`

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

a) `main`

- b) central
- c) head
- d) prime

18. 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

19. 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

20. 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

21. Un objeto de tipo String se caracteriza por ser:

a) Inmutable

b) Un tipo primitivo

c) Mutable

d) Polimórfico

22. 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)

23. 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

24. 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

25. 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

26. 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

27. 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

28. 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

29. 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

30. 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

31. 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)

32. 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

33. ¿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

34. ¿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

35. ¿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

36. ¿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

37. 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

38. ¿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

39. ¿Qué código colocado después del comentario //For loop podrá llenar el arreglo con los valores de la variable i ?

```
public class Lin1{  
  
    public static void main(String arg[]){  
  
        int i[] =new int[4];  
  
        for(int i=0;i < 5; i++ ) //for loop }
```

- a) i=0
- b) i[i] = 2
- c) i = i+1

d) `i[i] = i;`

40. ¿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

41. 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

42. Relaciona las columnas entre sí. The pairs are:

(C) Pruebas Unitarias --> A) Validar que los componentes desarrollados se ensamblen de forma adecuada con la aplicación.

(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.

(B) Pruebas de Sistema --> C) Validar que los componentes que forman parte del sistema funcionan correctamente y cumplen con los requisitos de manera independiente.

(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.

43. Relaciona las columnas entre sí. The pairs are:

(A) Pruebas Negativas o de Excepción --> A) Destinadas a mostrar que un componente o sistema no funciona.

(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.

(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.

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

44. 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

45. ¿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

46. 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