Leer detenidamente las siguientes instrucciones, antes de abordar la realización de la prueba, con el objetivo de tener claro los aspectos a valorar y/o lo demandado en la misma.

1. Hay preguntas con más de una posible respuesta.
2. Marca la solución que creas correcta con la herramienta de resaltado.
3. Algunas preguntas requieren escribir código (utiliza comentarios para aclarar los desarrollos) o escribir definiciones.
4. Por supuesto PREGUNTA cualquier duda o aclaración que necesites.
5. TIEMPO ESTIMADO: 65 min

A rellenar por el alumno:

Nombre:

Apellidos:

Fecha:

1. **¿En qué consiste el proceso de programación en Java?**

a) Escritura, compilación y verificación de la funcionalidad del código fuente de un programa

b) Compilación del código fuente de un programa

c) Compilación y verificación del código Bytecode de un programa

1. **Un programa Java compilado es portable porque:**

a) El código Bytecode es ejecutable por los principales sistemas operativos del mercado

b) El entorno de ejecución de Java incluye una máquina virtual que interpreta el código Bytecode

c) El entorno de ejecución interpreta el código Java, independientemente de la máquina virtual

1. **El entorno de ejecución de Java (Java Runtime Environment)**

a) Es un conjunto de librerías para desarrollo de aplicaciones Java

b) Es una pieza intermedia entre el código Bytecode y los distintos sistemas operativos existentes en el mercado. Incluye la máquina virtual de Java

c) Es la máquina virtual de Java

1. **La legibilidad de un programa Java es importante porque:**

a) Facilita el mantenimiento del software y permite corregir errores o modificar la funcionalidad con menor coste

b) Evita errores del compilador

c) Permite corregir errores, aunque no facilita el proceso de mantenimiento de una aplicación

1. **Java es un lenguaje que distingue letras mayúsculas y minúsculas.**

a) Falso

b) Verdadero

1. **Por convención, en Java los nombres de variables y métodos deben empezar por mayúscula. Si el nombre está compuesto por varias palabras, cada palabra debe empezar a su vez por mayúscula.**

a) Falso

b) Verdadero

1. **Complete las siguientes oraciones sobre el entorno de Java:**

El comando\_\_java\_\_\_\_\_\_ del JDK ejecuta una aplicación de Java.

El comando \_\_javac\_\_\_\_\_\_ del JDK compila un programa de Java.

Un archivo de programa de Java debe terminar con la extensión de archivo\_\_\_.java\_\_\_

Cuando se compila un programa en Java, el archivo producido por el compilador termina con la extensión .class

1. **Un tipo de dato indica los valores que pueden almacenar una variable y el rango de valores que admite.**

a) Falso

b) Verdadero

1. **Indique el valor de x: boolean x = 3 > 2 && 6 < 10 || true;**

a) true

b) false

1. **Indique la salida por consola del siguiente programa.**

public class Calculo2 {

public static void main (String[] args) {

boolean w, x = true, y = true, z = false;

w = x && y || x && z || y && z;

System.out.print("El resultado es ");

System.out.print(w);

}

}

Respuesta: El resultados es true

1. **Para utilizar componentes que están en otro paquete diferente o fuera del paquete java.lang se debe añadir una declaración de importación con la sintaxis:**

a) include nombre-del-paquete

b) import nombre-del-paquete

c) package nombre-del-paquete

1. **La estructura if-else es una estructura de selección doble porque selecciona entre dos bloques de sentencias mutuamente excluyentes. Si se cumple la condición, se ejecuta el bloque de sentencias asociado al if. Si la condición no se cumple, entonces se ejecuta el bloque de sentencias asociado al else.**

a) Falso

b) Verdadero

1. **La estructura switch es una estructura de selección múltiple que permite seleccionar un bloque de sentencias entre varios casos. Es equivalente a una estructura de selección de if-else anidados y siempre que se puede utilizar un if-else anidado se puede aplicar un switch.**

a) Falso

b) Verdadero

1. **¿Es posible interrumpir la ejecución de una estructura de selección switch?**

a) Sí, se utiliza la sentencia continue que provoca la finalización del switch. El flujo del programa continúa en la sentencia inmediatamente posterior al switch

b) Sí, cuando se cumple una de las condiciones se utiliza la primera sentencia break que encuentra, provocando la finalización del switch. El flujo del programa continúa en la sentencia inmediatamente posterior al switch

c) No, no es posible interrumpir la ejecución de un switch

1. **Seleccione la opción que describe correctamente las diferencias entre las estructuras de repetición while, do-while y for.**

a) La estructura de repetición while repite el bloque de sentencias mientras la condición es verdadera. La estructura do-while ejecuta el bloque de sentencias cero o más veces, comprueba la condición y repite el bloque de sentencias mientras la condición es verdadera. La estructura for repite el bloque de sentencias mientras la condición es verdadera.

b) La estructura de repetición while repite el bloque de sentencias mientras la condición es verdadera. La estructura do-while ejecuta el bloque de sentencias al menos una vez. Después comprueba la condición y repite el bloque de sentencias mientras la condición es verdadera. La estructura for repite el bloque de sentencias mientras la condición es verdadera

c) La estructura de repetición while repite el bloque de sentencias al menos una vez. La estructura do-while ejecuta el bloque de  sentencias cero o más veces, comprueba la condición y repite el bloque de sentencias mientras la condición es verdadera. La estructura for repite el bloque de sentencias mientras la condición es verdadera

1. **La sentencia continue únicamente puede aparecer en una estructura de repetición o bucle. Cuando se ejecuta un continue, se deja de ejecutar el resto de sentencias del bucle para volver al inicio de éste.**

a) Falso

b) Verdadero

1. **Indique la salida por la consola del siguiente programa**

public class Resultados {

public static void main(String[] args) {

int nota = 7;

if (nota >= 5) {

System.out.print("El resultado es aprobado ");

switch (nota) {

case 5:

case 6:break;

case 7:

case 8:System.out.println("con Notable");

break;

case 9:System.out.println("con Sobresaliente");

case 10:System.out.println("con Matrícula");

}

}

else{

System.out.println("El resultado es suspenso");

}

}

}

**Respuesta:** El resultado es aprobado con Notable

1. **Del siguiente programa indique la salida por la consola. ¿Cuántas iteraciones del for se realizan?**

public class Numeros {

public static void main(String[] args) {

for (int i=1; i<=1000; i++) {

if (i % 2 == 0)

continue;

if (i % 3 == 0)

continue;

System.out.println("Números: " + i);

if (i >= 15)

break;

}

}

}

**Respuesta:** 17

1. **El tipo base de un array es el tipo que se declara para todos sus elementos. El tipo base puede ser un tipo primitivo de Java, o un objeto como un String.**

a) Falso

b) Verdadero

1. **Para hacer referencia a un elemento de un array es necesario indicar la posición que ocupa en la estructura de almacenamiento. El primer elemento de un array se almacena en la posición 1 y el último elemento en la posición N, donde N es el tamaño del array.**

a) Falso

b) Verdadero

1. **La propiedad de un array que permite saber el número de elementos que tiene es:**

a) size

b) elements

c) length

1. **La declaración int[][][] numeros = new int[2][3][4] corresponde a:**

a) Un array de tipo base int de tres dimensiones, de tamaño

2x3x4, con identificador números donde todos los valores son null

b) Un array de tipo base int, de tres dimensiones, con identificador numeros, que aún no tiene valores.

c) Un array de tipo base int de tres dimensiones, de tamaño

2x3x4, con identificador números donde todos los valores son ceros

1. **Si tengo un array llamado colores que es bidimensional que contiene 3 filas y en donde la primera fila tiene 5 columnas, la segunda fila tiene 2 columnas y la tercera fila tiene 4 columnas qué resultado te devolvería la siguiente sentencia:**

***System.out.print(colores.length + colores[1].length);***

1. 8
2. 35
3. 53
4. 5

1. **Utilice un for (el que quieras) para recorrer el siguiente array y mostrar sus elementos en la consola.**

String[] meses = {"Enero","Febrero","Marzo","Abril",

"Mayo","Junio","Julio","Agosto",

"Septiembre","Octubre","Noviembre",

"Diciembre"};

Código: for (String mes:meses){

System.out.println(mes);

}

1. **Si tenemos la variable cadena de tipo String con el valor**

**“Java es el mejor lenguaje que existe y me gusta mucho”.**

**La sentencia cadena.substring(0,24).contains(“mejor”); devolverá:**

1. true
2. false
3. FatalException (“Eso es imposible”)
4. **¿Cuál de los siguientes métodos se utiliza para encontrar la posición de la primera ocurrencia de un carácter en una cadena en Java?**
5. indexOf()
6. charAt()
7. substring()
8. **¿Qué método se utiliza para eliminar todos los espacios en blanco que haya antes y después de un texto en Java?**
9. replaceAll()
10. replace()
11. trim()

1. **¿Cuál de los siguientes modificadores de acceso usarías si queremos que un método sólo sea accedido por únicamente la propia clase donde está definido?**
2. public
3. protected
4. private
5. **¿Cuál de las siguientes afirmaciones es cierta con respecto a los métodos en Java?**
6. Los métodos pueden tener múltiples valores de retorno.
7. Los métodos pueden tener el mismo nombre si tienen diferente número y/o tipo de parámetros.
8. Los métodos no pueden aceptar argumentos.
9. **¿Qué palabra clave se utiliza en la declaración de un método para indicar que no devuelve ningún valor en Java?**
10. void
11. int
12. return
13. **¿Qué elementos definen a un objeto?**
    1. Su cardinalidad y su tipo
    2. Sus atributos y sus métodos
    3. La forma en que establece comunicación e intercambia mensajes
    4. La interfaz que implementa
14. ¿Qué código de los siguientes tiene que ver con la herencia?
    1. public class Componente inherit Producto
    2. public class Componente implements Producto
    3. public class Componente extends to Producto
    4. public class Componente extends Producto
15. Si desde un método estático quiero llamar a otro método que no es estático y está en otra clase se necesita:
    1. Llamar al método únicamente con su nombre
    2. Llamar al método a través del nombre de la clase
    3. Llamar al método a través de un objeto que de la clase donde está
    4. No es posible llamar a un método que está en otra clase.
16. ¿Qué significa instanciar una clase?
    1. Duplicar una clase
    2. Eliminar una clase
    3. Crear un objeto a partir de la clase
    4. Conectar dos clases entre sí
17. ¿Cuántos objetos de tipo Domicilio se crean en el siguiente fragmento de código suponiendo que tenemos definida la clase Domicilio con un único constructor al que se le pasa un único parámetro de tipo String?

*Domicilio domicilio1 = new Domicilio("Calle1");*

*Domicilio domicilio2 = domicilio1;*

*Domicilio domicilio3 = new Domicilio("Calle2");*

*domicilio1 = domicilio3;*

* 1. 2
  2. 4
  3. 3
  4. 1

1. ¿Qué es el cast en Java?
   1. Una herramienta para crear objetos en Java
   2. Una forma de eliminar errores de compilación
   3. Un operador utilizado para convertir una variable de un tipo a otro
   4. Un método utilizado para eliminar errores en tiempo de ejecución
2. ¿Cuáles de los siguientes métodos sobrecargan al metodo1 declarado aquí?

*public void metodo1 (){*

*System.out.println(“Hola”)*

*}*

* 1. public void metodo1 (){

System.out.println(“Hola2”)

}

* 1. public void metodo1 (String texto){

System.out.println(texto)

}

* 1. public void metodo2 (){

System.out.println(“Hola”)

}

* 1. public void metodo1 (String texto, int repeticiones){

System.out.println(texto + repeticiones)

}

1. Supongamos que b tiene el valor 5 y c el valor 8. ¿Cuál es el valor de a, b y c después de la ejecución de esta línea de código? a = b++ + ++c;
   1. a = 15, b = 6 y c = 9
   2. a = 14, b = 6 y c = 8
   3. a = 15, b = 6 y c = 8
   4. a = 14, b = 6 y c = 9
2. Que mostraría la ejecución del siguiente código:

*String nombre = “ La programación en java es sencilla ”;*

*System.out.println(nombre.trim().substring(nombre.indexOf(“ ”)));*

* 1. sencilla
  2. La programación en java es sencilla
  3. La
  4. es sencilla

1. Para la siguiente declaración de variable, ¿qué afirmación es correcta?

int final = 1;

* 1. No se permite inicializar las variables cuando se declaran.
  2. Es correcta.
  3. No se puede usar una palabra reservada como nombre de una variable.
  4. La expresión correcta sería final int = 1;

1. Un método abstracto en Java: (Varias respuestas)
   1. Obliga a que la clase sea abstracta
   2. Obliga a que la clase sea final
   3. Es un método en el que no se implementa código
   4. No existen, solo existen clases abstractas
2. ¿Cómo puedo obtener un número aleatorio entre 0 y 1?
   1. Math.exp(…)
   2. Operador \*\*
   3. Math.random()
   4. Operador ^
3. ¿Cuál es la sintaxis correcta para implementar una interfaz llamada MyInterface en una clase en Java?
   1. class MyClass implements MyInterface extends ClasePadre{}
   2. interface MyClass implements MyInterface {}
   3. class MyClass extends ClasePadre implements MyInterface {}
   4. interface MyClass extends MyInterface {}
4. Los 4 pilares de la POO son:

a. Herencia, abstracción, encapsulación y polimorfismo.

b. Herencia, abstracción, implementación y polimorfismo.

c. Herencia, polimorfismo, abstracción y simplicidad.

1. En la herencia Java, señale la FALSA:

a. Una clase final concreta puede heredar de una abstracta.

b. Una clase final concreta puede heredar de otra concreta.

c. Una clase final concreta puede ser heredada a otra concreta.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | 24 |  |
| 2 |  | 25 |  |
| 3 |  | 26 |  |
| 4 |  | 27 |  |
| 5 |  | 28 |  |
| 6 |  | 29 |  |
| 7 |  | 30 |  |
| 8 |  | 31 |  |
| 9 |  | 32 |  |
| 10 |  | 33 |  |
| 11 |  | 34 |  |
| 12 |  | 35 |  |
| 13 |  | 36 |  |
| 14 |  | 37 |  |
| 15 |  | 38 |  |
| 16 |  | 39 |  |
| 17 |  | 40 |  |
| 18 |  | 41 |  |
| 19 |  | 42 |  |
| 20 |  | 43 |  |
| 21 |  | 44 |  |
| 22 |  | 45 |  |
| 23 |  |  |  |