

Apellido y Nombre: \_\_\_\_\_

## Examen Java Orientado a Objetos

### 1 – Cual opción es correcta para invocar el método toString() de la clase padre desde un método de la clase hija?

Indique cual de todas las respuesta es correcta.

1. super().toString();
2. super().toString;
3. extends.toString();
4. super.toString;
5. super.toString();
6. Ninguna respuesta es correcta.

\_\_\_\_\_

### 2 – Dado el siguiente código :

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println(this instanceof Test);  
    }  
}
```

Indique cual de todas las respuesta es correcta.

1. Imprime true.
2. Imprime false.
3. No se puede referenciar this desde un entorno estatico.
4. El operador instanceof no devuelve un booleano.
5. Ninguna respuesta es correcta.

\_\_\_\_\_

### 3 – Dado el siguiente código :

```
nr1++;
```

Indique cual de todas las respuesta es correcta.

1. El código contiene un operador binario.
2. El código contiene un operador ternario.
3. El código no contiene operadores.
4. El código contiene un operador unario
5. Ninguna respuesta es correcta.

\_\_\_\_\_

### 4 – Dado el siguiente código :

```
public class _10Examen {  
    public static void main(String... args) {  
    }  
}
```

Indique cual de todas las respuesta es correcta.

1. No compila.
2. Ejecuta con errores.
3. Ejecuta sin errores.
4. El método main no es correcto y no se puede ejecutar.

\_\_\_\_\_

### 5 – Dado el siguiente código :

```
public class &10Examen {  
    public static void main(String... args) {  
    }  
}
```

Indique cual de todas las respuesta es correcta.

1. No compila.
2. Ejecuta con errores.
3. Ejecuta sin errores.
4. El método main no es correcto y no se puede ejecutar.

\_\_\_\_\_

### 6 – Dado el siguiente código :

```
public class $10Examen {
    public static void main(String... args) {
    }
}
```

Indique cual de todas las respuesta es correcta.

- 
1. No compila por que la clase no puede tener un número como parte del nombre.
  2. Ejecuta con errores.
  3. Ninguna respuesta es correcta.
  4. El método main no es correcto y no se puede ejecutar.

**7 – Dado el siguiente código :**

```
public class Test {
    public static void main(String[] args) {
        final int[] vector=new int[5];
        vector[0]=2;
    }
}
```

Indique cual de todas las respuesta es correcta.

- 
1. No compila por que el vector es constante.
  2. Ejecuta con errores por que el vector es constante.
  3. Ejecuta sin errores.
  4. No se puede declarar constante un vector.

**8 – Dado el siguiente código :**

```
public class Test {
    private final String texto;
    public void Test(){
        texto="chau";
    }
    public static void main(String[] args) {
    }
}
```

Indique cual de todas las respuesta es correcta.

- 
1. La clase no tiene constructor declarado.
  2. Ejecuta sin errores.
  3. Todas las respuestas anteriores son correctas.
  4. Ninguna respuesta es correcta.

**9 – Dado el siguiente código :**

```
public class Test {
    protected int atributo1;
    protected int atributo2;
}
```

Indique cual de todas las respuesta es correcta.

- 
1. Todos los atributos no son visibles desde clases del mismo paquete.
  2. Todos los atributos son visibles desde clases del mismo paquete.
  3. Todos los atributos son visibles desde clases hijas, pero no desde clases del mismo paquete.
  4. Todas las respuesta son validas.

**10 – Dado el siguiente código :**

```
public class Test {
    public static void main(String[] args) {
        System.out.println(true||funcion1());
    }
    public static boolean funcion1(){ return true; }
}
```

Indique cual de todas las respuesta es correcta.

- 
1. No ejecuta por que funcion1() no puede ser static.
  2. No se ejecuta funcion1().
  3. Se ejecuta funcion1().

- Se ejecuta la funcion1() e imprime falso.

**11 – Dado el siguiente código :**

```
1- public class Test {  
2-     private String texto1;  
3-     public static void main(String[] args) {  
4-         String texto2=null;  
5-         System.out.println(texto1);  
6-         System.out.println(texto2);  
7-     }  
8- }
```

Indique cual de todas las respuesta es correcta.

- Imprime null null.
- Error solo en linea 5.
- Error solo en linea 6.
- Error en linea 5 y 6.
- Ninguna respuesta es correcta.

---

**12 – Cual de las siguientes afirmaciones es correcta :**

---

```
Map<int,String>map=new HashMap( );
```

- Crea un mapa no sincronizado.
- Crea un mapa sincronizado.
- Crea un mapa utilizando internamente una lista Enlazada
- Error de tipo de datos en la declaración <Generic>.
- Ninguna respuesta es correcta.

**13 – La interface Serializable:**

---

- Permite ejecutar procesos en un nuevo Thread.
- No aporta ninguna funcionalidad.
- Permite indicar que un atributo sera serializado.
- Permite indicar que un método sera serializado.
- Ninguna respuesta es correcta.

**14 – Dado el siguiente código :**

```
1- public class Test {  
2-     private static int nro;  
3-     public static void main(String[] args) {  
4-         nro--;  
5-         nro+=3;  
6-         System.out.println(nro);  
7-     }  
8- }
```

Indique cual de todas las respuesta es correcta.

- Imprime null.
- Error solo en linea 4.
- Error solo en linea 5.
- Error en linea 4, 5 y 6.
- Ninguna respuesta es correcta.

---

**15 – Cual de las siguientes afirmaciones es correcta :**

---

- HashSet es una implementación de Set que permite duplicados.
- LinkedHashSet es una implementación sincronizada de Set (Sus métodos son synchronized).
- TreeSet es la implementación de Set más veloz.
- Una implementación de Set no contiene valores duplicados en sus elementos
- Ninguna respuesta es correcta.

**16 – Cual de las siguientes afirmaciones es correcta :**

---

- La interface Comparable, contiene el método int compare(T o) usado para comparar Objetos

2. La clase StringBuffer es SafeThead.
3. La clase StringBuilder es SafeThread.
4. La clase LinkedList es SafeThead
5. Ninguna respuesta es correcta.

**17 – Cual de las siguientes afirmaciones es correcta :**

---

1. La estructura Try With resources usándola adecuadamente, permite que las implementaciones de Iterable se cierren automáticamente.
2. La estructura Try With resources usándola adecuadamente, permite que las implementaciones de Runnable se cierren automáticamente.
3. La estructura Try With resources usándola adecuadamente, permite que las implementaciones de Comparable se cierren automáticamente.
4. La estructura Try With resources usándola, adecuadamente permite que las implementaciones de Serializable se cierren automáticamente.
5. Ninguna respuesta es correcta.

**18 – Cual de las siguientes afirmaciones es correcta :**

---

1. .get(int i) es un método declarado en la interface Set
2. .get(int i) es un método declarado en la interface List
3. .get(int i) es un método declarado en la interface Collections
4. Todas las respuestas anteriores son correctas.
5. Ninguna respuesta es correcta.

**19 – En la clase PreparedStatement :**

---

1. El método .execute() devuelve un valor entero.
2. El método .executeUpdate() devuelve un booleando.
3. El método .executeQuery() no tiene devolución de valor.
4. Todas las respuestas anteriores son correctas.
5. Ninguna respuesta es correcta.

**20 – Dado el siguiente código:**

---

```
public void metodo(String texto){  
    System.out.println(texto.charAt(20));  
}
```

1. El método puede lanzar una NullPointerException.
2. El método puede lanzar una StringIndexOutOfBoundsException.
3. El método imprime si es posible el carácter de índice 20 del parámetro texto.
4. Todas las respuestas anteriores son correctas.
5. Ninguna respuesta es correcta.