

<b>Comenzado el</b>	lunes, 30 de septiembre de 2019, 10:01
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	miércoles, 2 de octubre de 2019, 12:02
<b>Tiempo empleado</b>	2 días 2 horas
<b>Calificación</b>	<b>12,00</b> de 12,00 ( <b>100%</b> )

Pregunta **1**

Correcta

Puntúa 1,00 sobre 1,00

En un mismo método puede haber más de un **bloque try/catch**.

Seleccione una:

- ☒ Verdadero
- ☐ Falso

Correcta.  
La respuesta correcta es 'Verdadero'

Pregunta **2**

Correcta

Puntúa 1,00 sobre 1,00

La cláusula **throws** nos permite arrojar instancias solamente de las excepciones creadas por el programador.

Seleccione una:

- ☐ Verdadero
- ☒ Falso

Correcta.  
La respuesta correcta es 'Falso'

Pregunta **3**

Correcta

Puntúa 1,00 sobre 1,00

Un bloque **finally** se ejecuta solamente cuando se da una excepción.

Seleccione una:

- ☐ Verdadero
- ☒ Falso

Correcta.  
La respuesta correcta es 'Falso'

Pregunta **4**

Correcta

Puntúa 1,00  
sobre 1,00

Las excepciones creadas por el programador derivan de la clase **Throwable**.

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

Correcta.

La respuesta correcta es 'Verdadero'

Pregunta **5**

Correcta

Puntúa 1,00  
sobre 1,00

La excepción **ArrayIndexOutOfBoundsException** se genera al intentar acceder a un elemento de un array más allá de los límites definidos inicialmente para ese array.

.

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

Correcta.

La respuesta correcta es 'Verdadero'

Pregunta **6**

Correcta

Puntúa 1,00  
sobre 1,00

**Manejo de excepciones** es un mecanismo poderoso que se encarga de los errores en tiempo de ejecución para mantener el flujo normal de la aplicación.

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

Correcta.

La respuesta correcta es 'Verdadero'


Pregunta **7**

Correcta

Puntúa 1,00  
sobre 1,00

Completa los espacios en blanco para manejar cualquier excepción posible.

`int x= 12, y= 0;`

 {  
`try`  
`int z= x/y;`

`System.out.println(z);`

`}`

 (Exception e) {

`catch`  
`System.out.println("Error");`

`}`

Respuesta correcta

La respuesta correcta es:

Completa los espacios en blanco para manejar cualquier excepción posible.

`int x= 12, y= 0;`

`[try]` {

`int z= x/y;`

`System.out.println(z);`

`}`

`[catch]` (Exception e) {

`System.out.println("Error");`

`}`


Pregunta **8**

Correcta

Puntúa 1,00  
sobre 1,00

¿Cuántos bloques **catch** puede contener un bloque **try/catch**?

Seleccione una:

- ☒ a. Tantos como necesites  Correcta.
- ☐ b. Solo uno
- ☐ c. Uno.
- ☐ d. Sólo dos.

Respuesta correcta

La respuesta correcta es: Tantos como necesites


Pregunta **9**

Correcta

Puntúa 1,00  
sobre 1,00

Un bloque **try** no puede contener múltiples bloques **catch** que manejen diferentes excepciones separadamente.

Seleccione una:

- ☐ Verdadero
- ☒ Falso 

Correcta

La respuesta correcta es 'Falso'

Pregunta **10**

Correcta

Puntúa 1,00  
sobre 1,00

La palabra clave **throw** te permite generar manualmente excepciones desde tus métodos.

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

Correcta.

La respuesta correcta es 'Verdadero'

Pregunta **11**

Correcta

Puntúa 1,00  
sobre 1,00

Arrastra y suelta las opciones a continuación para crear un método que lance una **IOException** si el parámetro es negativo.

```
public void hacer(int x) throws ✓ IOException ✓ {  
    if (x < 0)  
        throw ✓ new ✓ IOException();  
}
```

try

catch

Respuesta correcta

La respuesta correcta es:

Arrastra y suelta las opciones a continuación para crear un método que lance una **IOException** si el parámetro es negativo.

```
public void hacer(int x) [throws] [IOException] {  
    if (x < 0)  
        [throw] [new] IOException();  
}
```

Pregunta **12**

Correcta

Puntúa 1,00 sobre 1,00

Completa los espacios en blanco para definir un método “miFuncion”, la cual lance una excepción si su parámetro es menor a 0.

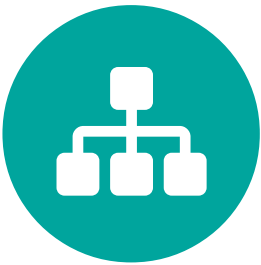
```
public void miFuncion(int arg) throws  {  
     ☒ (arg < 0) {  
         ☒ new IllegalArgumentException();  
    }  
}
```

Respuesta correcta

La respuesta correcta es:

Completa los espacios en blanco para definir un método “miFuncion”, la cual lance una excepción si su parámetro es menor a 0.

```
public void miFuncion(int arg) [throws] IllegalArgumentException {  
    [if] (arg < 0) {  
        [throw] new IllegalArgumentException();  
    }  
}
```



[Técnicas de Programación](#)



[Programación Orientada a Objetos](#)



[Base de Datos](#)



[Desarrollo de Software](#)



[Relaciones Laborales](#)