AUTOEVALUACIÓN UT9

1. ¿Qué método **imprime por la salida de error una traza** con las llamadas a los métodos desde los que se ha producido la excepción?
   1. getMessage()
   2. getClass()
   3. toString()
   4. printStackTrace()
2. ¿Cuál de los siguientes bloques para control de excepciones **NO es opcional**?
   1. finally
   2. try
   3. catch
   4. Todas las opciones son incorrectas.
3. Para **capturar una excepción**:
   1. Se debe utilizar un bloque try-catch-finally
   2. Se puede modificar la firma del método añadiendo throws y el tipo de excepción a capturar
   3. Se puede crear la excepción y lanzarla con la palabra reservada launch.
   4. Se puede crear la excepción y lanzarla con la palabra reservada throw.
4. La **excepción** que se generará al **convertir una cadena** “tres” en un número entero de **tipo primitivo** será:
   1. java.lang.PrimitiveIntegerTypeException
   2. java.lang.NumberFormatException
   3. java.lang.IllegalArgumentException
   4. java.lang.NullPointerException
5. Una **excepción personalizada**:
   1. Puede ser declarada como subclase de Exception
   2. Puede ser declarada como subclase de RunTimeException
   3. Las opciones a y b son correctas
   4. Todas las opciones anteriores son incorrectas.
6. Una **excepción checked**:
   1. Generalmente se produce por errores ajenos al código del programa.
   2. Se corresponde, por ejemplo, con la excepción *RuntimeException*.
   3. Es obligatorio que sea controlada.
   4. Todas las opciones anteriores son correctas.
7. Una **excepción unchecked**:
   1. Se corresponde, por ejemplo, con la excepción *ArrayIndexOutOfBoundsException.*
   2. Generalmente se produce errores de programación.
   3. No debe ser forzosamente capturada.
   4. Todas las opciones anteriores son correctas.