

Actividad 6.1. Excepciones

1. ¿Cuál es el nombre en Java de la clase que define las excepciones, y de la que debe heredar cualquier clase que queramos usar para representar una excepción?

La clase “abuela” sería la clase Object (java.lang.Object) y la clase padre de todas las Excepciones (y errores) es la clase Throwable. Si nos centramos meramente en la excepciones como tal, obviando los errores de la maquina de java, la clase de las excepciones es “Exception”(java.lang.Exception).

2. ¿Cuál es el nombre en Java de la clase que representa las excepciones que se producen al invocar un método de un objeto cuyo valor es “null”?

La clase más específica sería “NullPointerException”, que salta cuando se esperaba el valor de un objeto. Esta clase hereda de “RuntimeException”, que es una clase que define las excepciones en tiempo de ejecución.

3. ¿Cuál es el nombre en Java de la clase que representa las excepciones que se producen al obtener un comportamiento anómalo en la entrada / salida de información?

Estas excepciones entrarían en la categoría de “checked” puesto que son errores que no tienen que ver con la codificación del programa, si no que debemos de ser conscientes que pueden ocurrir y debemos de indicar un nuevo flujo.

En este caso, la clase encargada sería IOException, que se usa para manejar errores relacionados con la entrada y salida de los datos en Java.

4. Observa el siguiente fragmento de código: `String [] array_string = new String [25]; System.out.println (array_string [3].length());` ¿Qué excepción se produciría en el mismo?

Al no tener contenido (puesto que lo acabamos de crear) nos saldrá una excepción “NullPointerException” porque todas las posiciones carecen de información (null).

5. Observa el siguiente fragmento de código: `String aux = "hola"; int aux2 = Integer.parseInt (aux);` ¿Qué sucedería al ejecutar el mismo?

El código de ejemplo convierte los valores de una cadena en valores enteros, siempre y cuando tengan el formato adecuado. Al no tenerlo, nos salta la siguiente excepción: “`NumberFormatException`”. Esta excepción hereda de “`IllegalArgumentException`”, que tiene sentido porque es la excepción encargada de indicar si el argumento pasado es apropiado o no. Para que no salte la excepción del código proporcionado, deberemos introducir valores apropiados (en este caso valores numéricos).

6. Escribe un método auxiliar de nombre “`caracterEn`” en Java que realice la siguiente acción: Recibe como parámetros una cadena (`String`) y un entero (`int`); Si el entero está entre 0 y la longitud de la cadena (puedes hacer uso del método “`length(): int`” de la clase “`String`”) devuelve el carácter en la posición correspondiente (puedes hacer uso del método “`charAt(int)`” de la clase “`String`”). En caso contrario, construye y lanza una excepción de tipo “`Exception`”.

```
public class Actividad61Main {
    public static void main(String[] args) {
        /*
         String[] array_string=new String[25];
         System.out.println(array_string[3].length());
         */

        /*
         String aux = "hola";
         int aux2 = Integer.parseInt (aux);
         */
        System.out.println("Inicio del programa");
        try{
            String ex1="Holaquetalmellamajosemanuelytu?";
            int ex2=30;
            System.out.println("Caracter: "+caracterEn(ex1,ex2));
        }catch (StringIndexOutOfBoundsException ser){
            ser.printStackTrace();
            ser.getMessage();
        }finally {
            System.out.println("FIN DE LA EJECUCIÓN DEL PROGRAMA");
        }
    }
    public static char caracterEn(String a,int b){
        if (b<0&&b>a.length()){
            throw new StringIndexOutOfBoundsException("Fuera de la cadena");
        }else
            return a.charAt(b);
    }
}
```