

Prácticas

de

Java

Práctica 6: Excepciones

Uploaded by

Ingteleco

<http://ingteleco.webcindario.com>

ingtelecowed@hotmail.com

La dirección URL puede sufrir modificaciones en el futuro. Si no funciona contacta por email

PRÁCTICA 6ª : EXCEPCIONES

EJERCICIO 1º: DIVISIÓN POR CERO

Escribe una clase llamada *DividePorCero* que pruebe a dividir un entero entre otro recogiendo la excepción de división por cero y mostrando el mensaje "Error: división por cero" si se produce.

EJERCICIO 2º: CLASE FRACCIÓN

Diseña e implementa una clase *Fraccion* que permita crear fracciones (numerador y denominador enteros), con métodos para sumar, restar, multiplicar y dividirlos.

Crea una excepción *FraccionException* (declárala como una excepción explícita) que se lance siempre que en una operación de fracción se produzca un error, a saber:

- Creación de una fracción con denominador cero.
- Si el resultado de una división es una fracción con denominador cero (se dará si alguna de las fracciones a dividir tiene como numerador un cero).

Hacer que las operaciones críticas lancen esta excepción si se da el caso, con un mensaje indicativo del tipo de error (mensaje incluido en la excepción, no salida a pantalla!).

Construir un programa que pruebe el funcionamiento de la clase Fracción y sus operaciones.

EJERCICIO 3º: SUMA DE NÚMEROS

Construir un programa que reciba una serie de números (enteros) introducidos como parámetros desde la línea de comandos, los sume y visualice el resultado de la suma. En caso de que alguno de los números introducidos no sea un entero, el programa lo ignorará, mostrará un mensaje de error y continuará con la suma del resto de los números.

Ayuda: Para resolver este ejercicio puede ser interesante que hagas uso de la clase Integer.

EJERCICIO 4º: PREGUNTAS

Responde a las siguientes preguntas, suponiendo que la excepción con la que se está trabajando se trata de una excepción **Explícita**:

1. Suponer que en un bloque try, después de invocar al método que lanza la excepción, se pone otra sentencia, por ejemplo un mensaje a pantalla "Sentencia después de capturar una excepción". Si como resultado de ejecutar el método se lanza la excepción y se captura: ¿Sale el mensaje por pantalla?
2. Y, si no se captura y se ignora la excepción, ¿Sale el mensaje por pantalla?
3. Desde una método X se llama a un método Y que lanza una excepción ¿Si se ignora y no se pone en la declaración del método X, cuál es el mensaje de error del compilador?
4. ¿Sucede lo mismo si quien llama al método que lanza la excepción es el método 'main'?
5. Hacer la siguiente secuencia de llamadas:

Método f3: Llama al método f2 y saca un mensaje por pantalla "Soy el método f3"

Método f2: Llama al método f1 con el valor 2 como parámetro, vuelve a llamar al método f1 con el valor 1 como parámetro y saca un mensaje por pantalla "Soy el método f2"

Método f1: Lanza una excepción (Parametro1Exception) si el valor del parámetro que recibe es 1, en caso contrario saca un mensaje "Soy el método f1". La excepción hay que definirla.

a.- ¿Qué pasa si nadie captura la excepción?

b.- Modificar el programa para que f3 capture la excepción

c.- Modificar el programa para que f2 capture la excepción

d.- Indicar como conclusión:

¿Se ejecuta el código que está después del bloque catch cuando se ha producido la excepción?

¿Se ejecuta el código que viene después de la invocación al método que produce la excepción?

¿Qué pasa si tanto el método f2 como el método f3 capturan la excepción?

6. Con el ejemplo anterior, comprobar mediante ejemplos que el bloque finally se ejecuta:
 - a.- Si todo va bien y no se lanza la excepción
 - b.- Si se lanza la excepción y se captura
 - c.- Si se lanza la excepción pero no se captura
7. ¿Se puede poner un bloque finally sin un bloque try?
8. ¿Se puede poner un bloque finally sin un bloque catch?