

UNIVERSIDAD DE EL SALVADOR FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE DEPARTAMENTO DE INGENIERIA. Análisis Numérico ciclo I/2011

Docente: Ing. Carlos Arturo Ruano Instructor: Br. Oscar G. Rodríguez

MANEJO DE EXCEPCIONES

Ya tenemos creadas la clase Fraccion y Matriz pero no hemos tomado en cuenta errores q pueden surgir, por ejemplo al querer crear una fracción con denominador igual a cero o querer sumar dos matrices de dimensiones distintas, es aquí donde crearemos excepciones para nuestras clases.

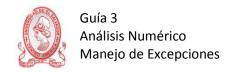
Las excepciones en Java están destinadas, al igual que en el resto de los lenguajes que las soportan, son para la detección y corrección de errores. Si hay un error, la aplicación no debería morirse. Se debería lanzar (*throw*) una excepción que nosotros deberíamos capturar (*catch*) y resolver la situación de error.

Lo haremos utilizando throw y para la implementación el try y el catch.

Lo primero que debemos hacer es crear una clase (dentro del paquete donde está la clase donde puede generarse la excepción) con el nombre que le daremos a nuestra excepción dicha clase heredara de la clase Exeption, en el constructor recibirá una cadena con el error y esa misma cadena la mandara a su clase padre (Exption) a través del **super**.

La clase quedara de la siguiente forma:

```
6
     package Frac;
7
8
9
10
      * @author mipe
11
12
     public class FraccionFormatExeption extends Exception {
13 -
         public FraccionFormatExeption(String ex) {
14
              super(ex);
15
          }
16
     }
17
```



Ahora que ya tenemos nuestra clase **FraccionFormatExeption** vamos a hacer uso de ella donde lo necesitemos en este caso será cuando el formato creado de una fracción sea incorrecto, el caso en el que una fracción puede ser incorrecta es que su denominador sea igual a cero. Así que la aplicaremos en el parseFraccion.

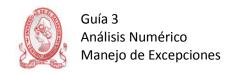
Para llamar la excepción si se cumple la condición incorrecta allí la llamaremos a travez del throw y le mandaremos en una cadena el error.

```
throw new FraccionFormatExeption("Denominador igual a cero");
```

De igual forma lo llamaremos en dado caso q al convertir la él numerador o el denominador a Long de error, o si al ingresar el String tiene una forma diferente a una fracción.

Nuestra función parseFracción quedara de la siguiente forma:

```
public static Fraccion parseFraccion(String a) throws FraccionFormatExeption {
107 🖃
              String f[] = a.split("/");
108
109
              try {
                  Fraccion resultado;
110
111
                  if (f.length == 2) {
112
                       if (!f[1].equals("0")) {
113
                           resultado = new Fraccion(Long.parseLong(f[0]), Long.parseLong(f[1]));
114
                           return resultado;
115
                       } else {
116
                           throw new FraccionFormatExeption("Denominador igual a cero");
117
                       }
118
                   } else if (f.length == 1) {
119
                       resultado = new Fraccion(Long.parseLong(f[0]), 1);
120
                       return resultado;
121
                   } else {
                       throw new FraccionFormatExeption("formato fraccion incorrecto");
122
123
                   }
124
125
              } catch (NumberFormatException ex) {
126
                   throw new FraccionFormatExeption("formato fraccion incorrecto");
127
              }
128
129
          }
```



A la firma de la función le colocamos **throws FraccionFormatExeption** que indica que en esa función se puede generar la excepción **FraccionFormatExeption**.

Ahora cuando hagamos uso del parseFraccion lo haremos encerrando el código en un **try** y lo que hara si se genera la excepción en un **catch**, ejemplo:

```
public class Main {
20
21
22 -
23
          * @param args the command line arguments
24
25 🖃
         public static void main(String[] args) {
26
             try {
                 String a = JOptionPane.showInputDialog("ingrese una fraccion de la forma num/den");
27
                 JOptionPane.showMessageDialog(null, "la fraccion ingresada es: \n" + Fraccion.parseFraccion(a));
28
             } catch (FraccionFormatExeption ex) {
29
                 JOptionPane.showMessageDialog(null, ex.toString());
30
31
32
         }
33
34
35
     }
36
```

Cree la porción de código anterior para terminar de comprender el funcionamiento. Y pruebe ingresando: 5/2, 5/s, 4/0, 7/8/8.

Ahora proceda a manejar los siguientes errores:

- En el constructor de la clase Fraccion no pueda ingresar el denominador igual a cero.
- Que no pueda sumar dos matrices de diferente dimensión.
- Que para multiplicar dos matrices filas de a sea igual a columnas de b y columnas de a sea igual a filas de b.
- Que para invertir una matriz debe de ser cuadrada.
- Y todas las excepciones que considere necesarias en matriz y fracción.