Trabajo Práctico 7 – Excepciones

1. Captura y Creación de Excepciones

En este ejemplo implementamos una Matriz de Objetos y una Excepción asociada para controlar los posibles errores.
La clase MatrizObjetos soporta objetos de cualquier clase como elementos que la componen, está implementada sobre la clase Vector del paquete java.util, se tomo un array de objetos vectores, la cantidad de filas y columnas esta limitada por dos variables estáticas pertenecientes a la clase MatrizObjetos.

Captura

El método SetRowCol() de la clase MatrizObjetos captura un posible error al acceder al array cuerpo, observe que el acceso al array se realiza dentro de un bloque try { } , y si ocurre un error se captura la Excepción en el bloque catch. El código es este :

```
try {
cuerpo[row].add(col,obj);
}
catch(Exception e){
System.out.println ("ERRRRROOOOOOOR!!!!!" );
System.out.println(e);
}
```

finally

Cuando ocurre un error java lanza una excepción e imprime en la consola información de la Excepción lanzada, deteniendo la ejecución del programa, si se controlan estas excepciones es posible hacer que el programa siga en ejecución, esto implica que en algunos casos se requiera realizar operaciones aun cuando existe un error. Esto se realiza con finally.
En el método GetRowCol() se muestra la utilización del bloque finally, el código es este :

```
Object obj = new Object();
try {
obj = cuerpo[row].elementAt(col);
}
catch(Exception e){
System.out.println("ERROR -> :"+e );
}
finally {
return obj;
}
```

Observe que este método siempre retornará un Object exista o no un error.

Lanzar una Excepción

El Constructor de la clase MatrizObjetos lanza una excepción, observe que hay que indicarle al método que excepción podrá lanzar, esto se realiza con la palabra throws en la declaración del método, y a continuación el nombre del la clase Excepción que se utilizará.

```
MatrizObjetos(int columnas, int filas) throws MatrizException {
```

Para lanzar la excepción se utiliza la palabra throw como se muestra en el siguiente código :

```
if(columnas > max_cols) {
throw new MatrizException("Exede la cantidad de columnas"); }
```

Crear una Excepción de control para una clase

Cada clase creada que posiblemente pueda lanzar errores tendrá una Clase Excepción asociada, esta se implementa como herencia de la clase Exception como en este ejemplo :

```
public class MatrizException extends Exception {
```

Aquí se muestra la declaración de la Clase MatrizExcepcion que se utiliza para controlar los errores en la clase MatrizObjetos.

Completar la clase MatrizObjetos y MatrizException de acuerdo a :

a) Lanzar una excepción en el método SetRowCol() de la clase MatrizObjetos si se quiere acceder a una posición que no es correcta en el array cuerpo.
b) Lanzar las Excepciones con el número de error, este número esta de acuerdo al array de errores en la clase MatrizException. Observe que la clase tiene dos constructores. El que recibe un String se utiliza cuando el error controlado no esta definido en el array de errores.
c) Completar la clase MatrizObjetos y agregar nombres de errores a la clase MatrizException

A continuación se muestran la clase MatrizObjetos, MatrizException y una Clase de prueba para el esquema.

Clase MatrizObjetos

```
import java.util.*;
public class MatrizObjetos {
static int max_rows = 200;
static int max_cols = 200;
private Vector cuerpo[];
MatrizObjetos(int columnas, int filas) throws MatrizException {
if(columnas > max_cols) {
throw new MatrizException("Exede la cantidad de columnas"); }
else {
cuerpo = new Vector[filas];
for(int i=0; i<filas; i++) cuerpo[i]= new Vector(2); }
}
public void SetRowCol(int row, int col, Object obj){
try {
cuerpo[row].add(col,obj); }
catch(Exception e){
System.out.println ("ERRRRROOOOOOOR!!!!!" );
System.out.println(e);
}}
public Object GetRowCol(int row, int col){
Object obj = new Object();
try {
obj = cuerpo[row].elementAt(col); }
catch(Exception e){
System.out.println("ERROR -> :"+e ); }
finally {
return obj; }
}
public String toString(){
String staux = new String("");
for(int i=0;i<cuerpo.length;i++)
for(int j=0;j< cuerpo[i].size();j++){
staux = staux + cuerpo[i].elementAt(j);
}
return staux;
}}
```

Clase control de Error para la Matriz de objetos

```
public class MatrizException extends Exception {
private static String Errores[] ={"Excede las columnas", "Excede las filas" } ;
private String name;
public MatrizException(String nom){ name = nom; }
public MatrizException( int numError){ name = Errores[numError] ; }
public String toString() { return "ERROR MATRIZ :"+name; }
}
```

Clase de prueba para la Matriz de objetos y control de Excepciones

```
class App {
public static void main(String args[]){
try {
MatrizObjetos ma = new MatrizObjetos(402,2);
String st1 = new String(" esto es un String ");
Integer entero = new Integer(10);
ma.SetRowCol(0, 0, st1);
ma.SetRowCol(1, 0, entero);
System.out.println( ma.GetRowCol(0,0) );
ma.SetRowCol(10,0,st1);
System.out.println(ma);
}
catch(MatrizException e){
System.out.println( e );
}}}
```