**Excepciones 2: Piscina II**

En este ejercicio se reutiliza la aplicación desarrollada en el ejercicio anterior (Excepciones I). **Se practicará el diseño de una excepción personalizada.**

**1.-** Declara la clase **PiscinaNivelExcepcion** que extienda de la clase Exception. En esta clase declaramos un constructor:

public **PiscinaNivelExcepcion** (String descripción); En este constructor se pasa la cadena descripción al método constructor de la clase Exception.

**2.-** Modifica la clase **Piscina** para que lance la excepción propia:

**class Piscina**

{

**private int** nivel;

**public final int** MAX\_NIVEL;

**public** Piscina(**int** max)

{

**if** (max<0) max=0;

MAX\_NIVEL=max;

}

**public int** getNivel()

{

**return** nivel;

}

**public void** vaciar(**int** cantidad) **throws PiscinaNivelException**

{

**/\*Si (nivel-cantidad <0), se lanza la excepción.**

**La descripción será** "Vaciado excesivo" y nivel se queda igual.

**En otro caso** nivel será igual a nivel-cantidad. \*/

}

**public void** llenar(**int** cantidad) **throws PiscinaNivelException**

{

**/\*Si (nivel+cantidad >** MAX\_NIVEL**), se lanza la excepción.**

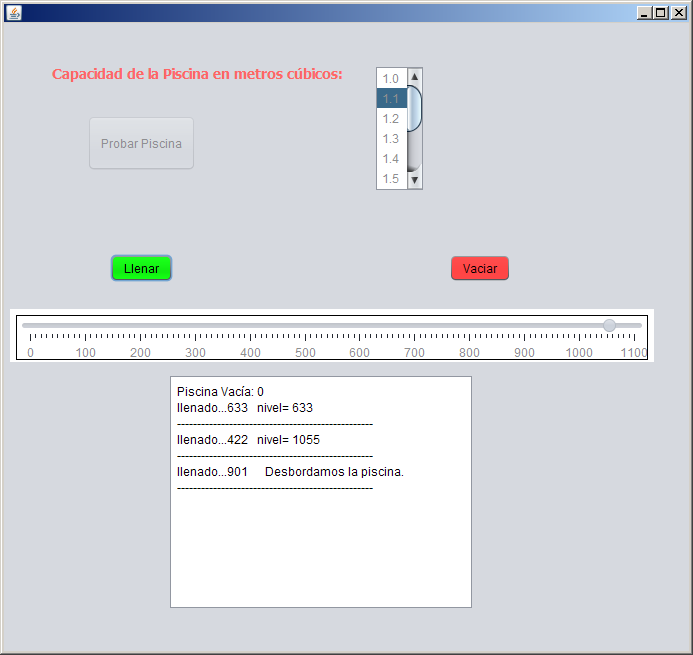
**La descripción será** "Vaciado excesivo" y nivel se queda igual.

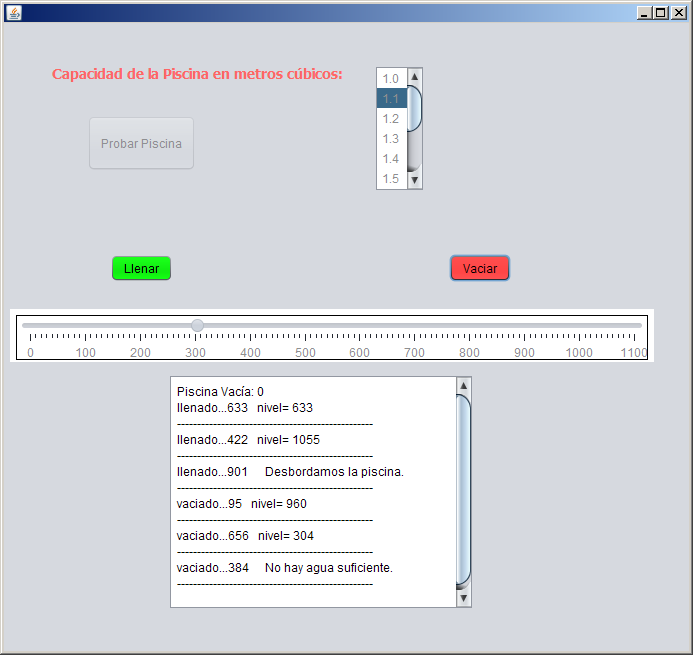
**En otro caso** nivel será igual a nivel+cantidad. \*/

}

}

3.- Reprogramar los botones **Llenar** y **Vaciar** para que recojan la excepción creada.





**SOLUCIÓN:**

**//1º Definición de la excepción propia.**

package excepciones2;

class PiscinaNivelException extends Exception

{

//int nivel;

public PiscinaNivelException (String descripcion)

{

super (descripcion);

//nivel=valor;

}

}

**//2º Lanzamiento de la excepción cuando se produce una situación anómala.**

package excepciones2;

class Piscina

{

private int nivel;

public final int MAX\_NIVEL;

public Piscina(int max)

{

if (max<0) max=0;

MAX\_NIVEL=max;

}

public int getNivel()

{

return nivel;

}

public void vaciar(int cantidad) throws PiscinaNivelException

{

if (nivel-cantidad < 0)

throw new PiscinaNivelException("No hay agua suficiente.");

else

nivel=nivel-cantidad;

}

public void llenar(int cantidad) throws PiscinaNivelException

{

if (nivel+cantidad > MAX\_NIVEL)

throw new PiscinaNivelException("Desbordamos la piscina.");

else

nivel=nivel+cantidad;

}

}

package excepciones2;

import javax.swing.JOptionPane;

public class **MiPiscina** extends javax.swing.JFrame

{

Piscina P;

public MiPiscina()

{

initComponents();

lcapacidad.setSelectedIndex(0);

PanelPiscina.setVisible(false);

}

**. . . . . . . . . . . . . . . .**

private void **btnpiscinaActionPerformed**(java.awt.event.ActionEvent evt)

{

String Mensaje="";

String Capacidad=(String)lcapacidad.getSelectedValue();

lcapacidad.setEnabled(false);

double tamaño=0;

int litrosPiscina=0;

try{

tamaño=Double.parseDouble(Capacidad);

litrosPiscina=(int)(tamaño\*1000);

}catch (NumberFormatException e1)

{

Mensaje+= "Tamaño no válido.\n";

JOptionPane.showMessageDialog(null, Mensaje);

}

P = new Piscina(litrosPiscina);

btnpiscina.setEnabled(false);

//preparamos el deslizador jS

jS.setMinimum(0);

jS.setMaximum(litrosPiscina);

jS.setMajorTickSpacing(100);

jS.setMinorTickSpacing(10);

jS.setValue(0);

jS.setPaintTicks(true);

jS.setPaintLabels(true);

jS.setEnabled(false);

salida.append("Piscina Vacía: "+P.getNivel()+"\n");

PanelPiscina.setVisible(true);

}

//3º En la clase de más alto nivel se programan secciones de código

// que recogen la excepción creada.

private void **btnLlenarActionPerformed**(java.awt.event.ActionEvent evt)

{

int mas;

try

{

mas=(int)(Math.random()\*1000);

salida.append("llenado..." + mas+" ");

P.llenar(mas);

salida.append ("nivel= "+ P.getNivel()+"\n");

jS.setValue(P.getNivel());

salida.append ("-------------------------------------------------\n");

}

catch (Exception e)

{

salida.append ("\t"+e.getMessage()+"\n");

salida.append ("-------------------------------------------------\n");

}

}

private void **btnVaciarActionPerformed**(java.awt.event.ActionEvent evt)

{

int menos;

try

{

menos=(int)(Math.random()\*1000);

salida.append("vaciado..." + menos+" ");

P.vaciar(menos);

salida.append ("nivel= "+ P.getNivel()+"\n");

jS.setValue(P.getNivel());

salida.append ("-------------------------------------------------\n");

}

catch (Exception e)

{

salida.append ("\t"+e.getMessage() +"\n");

salida.append ("-------------------------------------------------\n");

}

}

public static void **main**(String args[])

{

**. . . . . . . . . . .**

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable()

{

public void run()

{

new MiPiscina().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JPanel PanelPiscina;

private javax.swing.JButton btnLlenar;

private javax.swing.JButton btnVaciar;

private javax.swing.JButton btnpiscina;

private javax.swing.JLabel jLabel1;

private javax.swing.JSlider jS;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JList lcapacidad;

private javax.swing.JPanel paneSlider;

private javax.swing.JTextArea salida;

// End of variables declaration

}