**Excepciones 1: Piscina I**

Sea la clase siguiente:

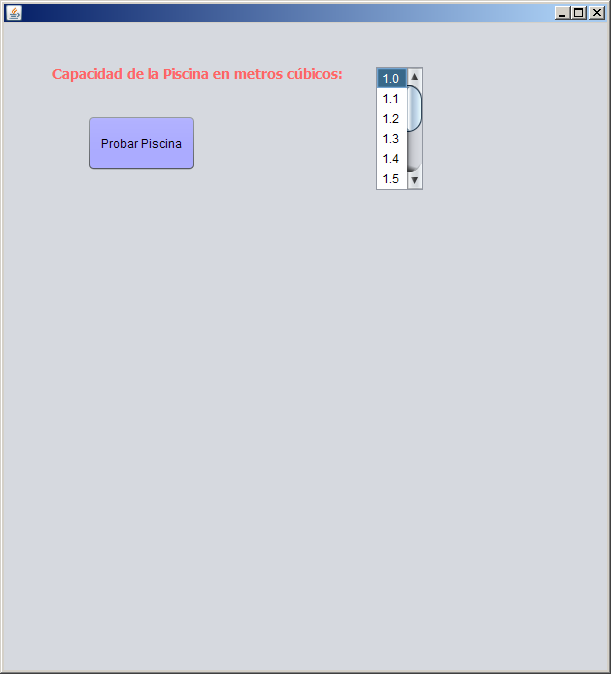


**nivel:** indica la cantidad de litros de agua que tiene la piscina.

**MAX\_NIVEL:** indica la capacidad máxima en litros que tiene la piscina.

**1.-** Modifica los métodos vaciar(…) y llenar(…) de manera que lancen una excepción cuando al vaciar el nivel quede por debajo de cero y cuando al llenar el nivel quede por encima de MAX\_NIVEL.

**2.-** Diseña una aplicación que nos permita elegir una piscina por su capacidad en metros cúbicos y a continuación se probará la piscina.



Probar la Piscina consiste en:

**a.-** Leer de la lista la capacidad de la piscina elegida y calcular los litros que tiene.

**b.-** Modificar las propiedades de un jSlider que representa la piscina. Se modificarán sus propiedades como se indica a continuación:

jS.setMinimum(0); //Valor Mínimo

jS.setMaximum(litrosPiscina); //Valor Máximo

jS.setMajorTickSpacing(100); //Definir marcas mayores cada 100 unidades

jS.setMinorTickSpacing(10); //Definir marcas menores cada 10 unidades

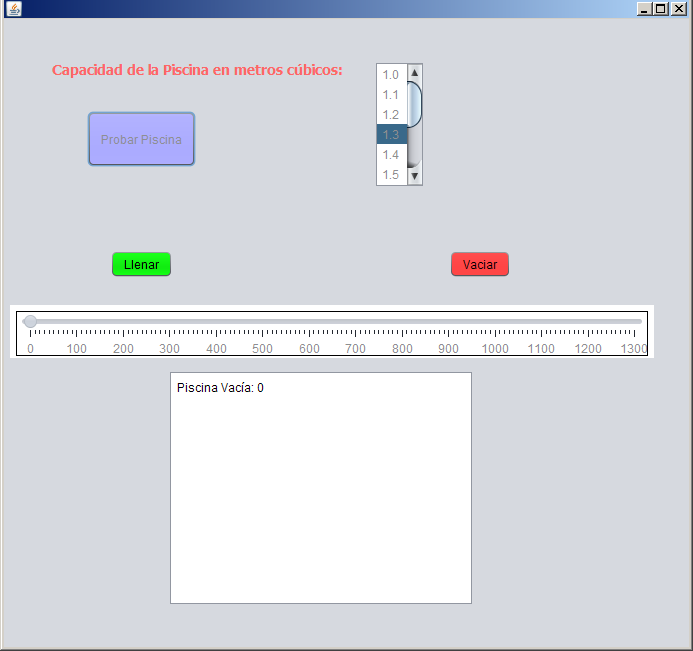
jS.setValue(0); //Asignación del valor inicial

jS.setPaintTicks(true); //Mostrar marcas mayores y menores

jS.setPaintLabels(true); //Mostrar numeración de marcas mayores

jS.setEnabled(false); //desactivar jSlider

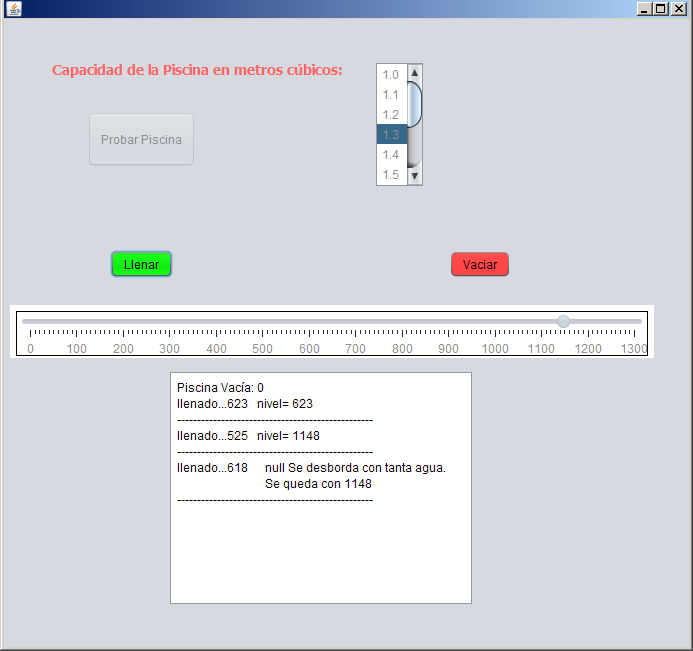
**c.-** Hacer visible el panel que contiene la piscina, el área de texto y los botones que llenan y vacían la piscina.



**3.-** Programar los botones **Llenar** y **Vaciar**.

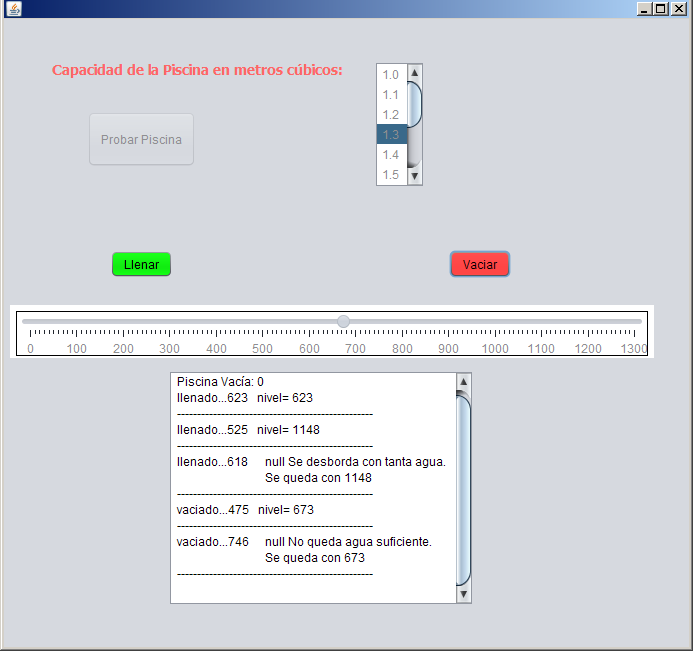
**a.-** El botón llenar simboliza que queremos echar agua en la piscina. Se generará un número aleatorio de litros (entre 1y 1000) y ejecuta el método **llenar(…)**.

* Si esta cantidad sobrepasa NIVEL\_MAX entonces se produce una excepción que se captura con **try-catch**. Si se produce una excepción se emite mensaje de texto asociado a la excepción más el nivel de la piscina. El mensaje se escribe en el área de texto.
* Si esta cantidad no sobrepasa NIVEL\_MAX el nivel aumenta y se debe reflejar en el deslizador. La operación que se realiza y el nuevo nivel aparecen escrito en el área de texto.

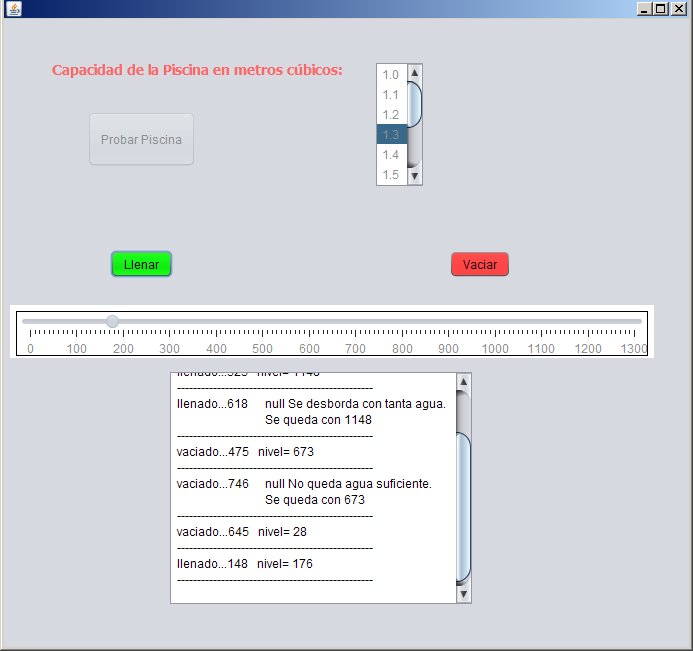


**b.-** El botón vaciar simboliza que queremos sacar agua de la piscina. Se generará un número aleatorio de litros (entre 1y 1000) y ejecuta el método **vaciar(…)**.

* Si la cantidad a sacar es mayor que el nivel de agua que tiene la piscina entonces se produce una excepción que se captura con **try-catch**. Si se produce una excepción se emite mensaje de texto asociado a la excepción más el nivel de la piscina. El mensaje se escribe en el área de texto.
* Si esta cantidad es menor que el nivel de agua que tiene la piscina entonces el nivel disminuye y se debe reflejar en el deslizador. La operación que se realiza y el nuevo nivel aparecen escrito en el área de texto.



Después de vaciar y llenar varias veces.



**SOLUCIÓN:**

class Piscina

{

private int nivel;

public final int MAX\_NIVEL;

public Piscina(int max)

{

if (max<0) max=0;

MAX\_NIVEL=max;

System.out.println("Max Nivel: "+max+" "+MAX\_NIVEL);

}

public int getNivel()

{

return nivel;

}

public void vaciar(int cantidad) throws Exception

{

if (nivel-cantidad < 0)

throw new Exception();

else

nivel=nivel-cantidad;

}

public void llenar(int cantidad) throws Exception

{

if (nivel+cantidad > MAX\_NIVEL)

throw new Exception();

else

nivel=nivel+cantidad;

}

}

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);

}

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() +" Se desborda con tanta agua.\n \tSe queda con " + P.getNivel()+"\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() +" No queda agua suficiente.\n \tSe queda con " + P.getNivel()+"\n");

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

}

}

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

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

{

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

}