5/18/2015

ADVANCE PROGRAMMING

LAB TASK 11

FAHEEM AZFAR

F-43030

BS(SE)-22B

**Q1.**

Design a java applet that displays a string “hello” scrolling horizontally on the drawing panel when the user click start button using Timer class in swing package.

The animation should stop on pressing stop button.

SOLUTION:

**import** java.awt.Container;

**import** java.awt.FlowLayout;

**import** java.awt.Graphics;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JApplet;

**import** javax.swing.JButton;

**import** javax.swing.Timer;

**public** **class** Question1 **extends** JApplet **implements** ActionListener

{

JButton b1,b2;

Timer t;

**int** x;

**public** **void** init()

{

b1=**new** JButton("Start");

b2=**new** JButton("Stop");

Container c=getContentPane();

FlowLayout f=**new** FlowLayout(FlowLayout.***CENTER***);

c.setLayout(f);

c.add(b1);

c.add(b2);

b1.addActionListener(**this**);

b2.addActionListener(**this**);

t=**new** Timer(50,**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e){

x=x+5;

**if**(x==getWidth())

{

x=0;

}

repaint();

}

});

}

**public** **void** actionPerformed(ActionEvent e)

{

**if**(e.getSource()==b1)

{

t.start();

}

**else** **if**(e.getSource()==b2)

{

t.stop();

}

}

**public** **void** paint(Graphics g)

{

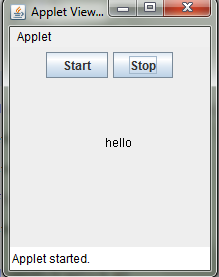
**super**.paint(g);

g.drawString("hello ", x, 100);

}

}

OUTPUT:



**Q2.**

Provide event handling for the microwave interface as follows

1. When user click ***food to be placed here button*** image of some food should be load there on button.
2. User selects the time by pressing digits representing minutes and it should be displayed on the text field above. Make sure the text field should be uneditable as its just display user the time and doesn’t allow any user input.
3. When the user press start button ***the food to be placed here button*** should become disable and the minutes in the text field start decreasing till it reached to zero(show message dialog with a message your food is ready now) or user press stop button.

When user click stop button the food to be placed here button should get enabled again and the text field gets clear for ne

SOLUTION:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Microwave extends JApplet implements ActionListener {

JButton arr[],b;

JTextField t;

int counter;

ImageIcon icon;

Thread a;

public void init()

{

setSize(400,200);

t=new JTextField(10);

t.setEditable(false);

arr=new JButton[12];

for(int i=0;i<12;i++)

{

arr[i]=new JButton();

}

b=new JButton("Food to be placed here");

arr[0]=new JButton("1");

arr[1]=new JButton("2");

arr[2]=new JButton("3");

arr[3]=new JButton("4");

arr[4]=new JButton("5");

arr[5]=new JButton("6");

arr[6]=new JButton("7");

arr[7]=new JButton("8");

arr[8]=new JButton("9");

arr[9]=new JButton("0");

arr[10]=new JButton("start");

arr[11]=new JButton("stop");

arr[11].setEnabled(false);

JPanel p=new JPanel();

p.setLayout(new GridLayout(4,3));

for(int i=0;i<12;i++)

{

p.add(arr[i]);

}

b.addActionListener(this);

JPanel p2=new JPanel();

p2.setLayout(new BorderLayout());

p2.add(t,BorderLayout.NORTH);

p2.add(p);

Container c=getContentPane();

c.setLayout(new GridLayout(1,2));

c.add(b);

c.add(p2);

for(int i=0;i<12;i++)

{

arr[i].addActionListener(this);

}

a=new Thread()

{

public void run()

{

while(counter>=0)

{

counter--;

t.setText(""+counter);

if(counter==0)

{

t.setText("Food is ready");

arr[10].setEnabled(true);

arr[11].setEnabled(false);

}

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

validate();

}

}

};

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==arr[0])

{

t.setText(""+t.getText()+"1");

}

else if(e.getSource()==arr[1])

{

t.setText(""+t.getText()+"2");

}

else if(e.getSource()==arr[2])

{

t.setText(""+t.getText()+"3");

}

else if(e.getSource()==arr[3])

{

t.setText(""+t.getText()+"4");

}

else if(e.getSource()==arr[4])

{

t.setText(""+t.getText()+"5");

}

else if(e.getSource()==arr[5])

{

t.setText(""+t.getText()+"6");

}

else if(e.getSource()==arr[6])

{

t.setText(""+t.getText()+"7");

}

else if(e.getSource()==arr[7])

{

t.setText(""+t.getText()+"8");

}

else if(e.getSource()==arr[8])

{

t.setText(""+t.getText()+"9");

}

else if(e.getSource()==arr[9])

{

t.setText(""+t.getText()+"0");

}

else if(e.getSource()==arr[10])

{

counter=Integer.parseInt(t.getText());

arr[10].setEnabled(false);

arr[11].setEnabled(true);

a.start();

}

else if(e.getSource()==b)

{

ImageIcon icon = new ImageIcon("images.jpg");

Image im=icon.getImage();

im=im.getScaledInstance(200, 200, java.awt.Image.SCALE\_SMOOTH);

icon = new ImageIcon(im);

b.setIcon(icon);

validate();

}

else if(e.getSource()==arr[11])

{

t.setText("");

arr[10].setEnabled(true);

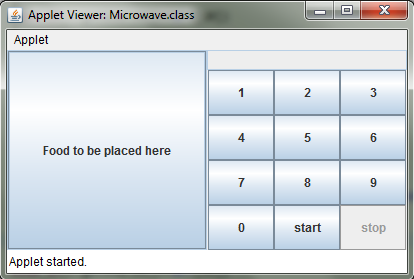
a.stop();

}

}

}

OUTPUT:



**Q3.**

Write a class Area that calculates the area of a rectangle in main function, using the following formula:

area= length\* breadth

The program reads the length (L) and the breadth (B) from input Dialog and Test the following:

**a)** The program should handle a NumberFormatException (Unable to parse string into int etc) if the sides are not legal values.The program should check the information read from the command line and give suitable feedback in case of exception occurs.

***Test the output on valid and invalid inputs.***

**b)** Modify part a by introducing throws clause in its declaration.(Do not remove try catch)

***Test the output on valid and invalid inputs.***

**c)** Remove try –catch block and leave the throws clause as it is.

**d)** Modify part c such that in catch print the exception message using e.getMessage().(do not remove throws clause)

***Test the output on valid and invalid inputs.***

SOLUTION:

**import** javax.swing.JOptionPane;

**public** **class** Area

{

**int** length=0;

**int** breadth=0;

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

{

Area a=**new** Area();

JOptionPane.*showMessageDialog*(**null**, "AREA"+(a.length\*a.breadth));

}

**public** **void** Question3() **throws** NumberFormatException

{

**while** (**true**){

**try**

{

String a=JOptionPane.*showInputDialog*(**null**,"Enter length");

length=Integer.*parseInt*(a);

}

**catch**(NumberFormatException e)

{

String a=e.getMessage();

JOptionPane.*showMessageDialog*(**null**,"WRONG INPUT");

}

}

**while**(**true**)

{

**try**

{

String b=JOptionPane.*showInputDialog*(**null**,"Enter breadth");

breadth=Integer.*parseInt*(b);

}

**catch**(NumberFormatException e)

{

String b=e.getMessage();

JOptionPane.*showMessageDialog*(**null**,"WRONG INPUT");

}

}

}

}

}