/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Name: Sarah Redmon  
Date: 2/2/19  
Instructor: Ms. Tucker  
Class: Jeopardy  
Purpose: To show a GUI timer for a Jeopardy final question  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*/

import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
import java.applet.AudioClip;  
import java.net.URL;  
import javax.swing.JButton;  
import javax.swing.Timer;

public class JeopardyPanel extends JPanel  
{  
    /\*------------------------------------------------------------------------------  
    Initializes variables, JLabels, JRadioButtons, JButtons, etc. for program  
    ------------------------------------------------------------------------------  
    \*/  
    private int timer;  
    private Timer time;  
    private JLabel jeopardyHeading, timerLabel, noteLabel;  
    private JButton stopButton, playButton, exitButton;  
    private JRadioButton fiveSec, thirtySec;  
    private AudioClip[] music;  
    private AudioClip current;  
      
    public JeopardyPanel()  
    {  
        /\*------------------------------------------------------------------------------  
        Layout is set as Grid (2 by 4)  
        ------------------------------------------------------------------------------  
        \*/  
        setLayout(new GridLayout (2, 4));  
          
        /\*------------------------------------------------------------------------------  
        Time object is created to make timer subtract by one every second (HAS HIDDEN LOOP)  
        ------------------------------------------------------------------------------  
        \*/  
        time = new Timer(1000, new TimerListener());  
          
        /\*------------------------------------------------------------------------------  
        Music is imported and array is made to distinguish null from file  
        ------------------------------------------------------------------------------  
        \*/  
        URL url1;  
        url1 = null;  
          
        try  
        {  
            url1 = new URL ("file", "localhost", "jeopardy.au");  
        }  
        catch (Exception exception) {}  
      
        music = new AudioClip[2];  
        music[0] = null;  
        music[1] = JApplet.newAudioClip (url1);  
          
        current = null;  
          
        /\*------------------------------------------------------------------------------  
        Heading is made as JLabel  
        ------------------------------------------------------------------------------  
        \*/  
        jeopardyHeading = new JLabel ("JEOPARDY TIMER");  
        jeopardyHeading.setFont (new Font ("Comic Sans MS", Font.BOLD, 18));  
          
        /\*------------------------------------------------------------------------------  
        JRadioButtons are made (5 seconds & 30 seconds)  
        ------------------------------------------------------------------------------  
        \*/  
        fiveSec = new JRadioButton ("Stop after 5 seconds");  
        fiveSec.setBackground (Color.pink);  
        thirtySec = new JRadioButton ("Stop after 30 seconds");  
        thirtySec.setBackground (Color.pink);  
          
        /\*------------------------------------------------------------------------------  
        Play, Stop, & Exit Buttons are made & action listeners added into Button Listener  
        ------------------------------------------------------------------------------  
        \*/  
        playButton = new JButton ("Play", new ImageIcon ("play.gif"));  
        playButton.setBackground (Color.green);  
        stopButton = new JButton ("Stop", new ImageIcon ("stop.gif"));  
        stopButton.setBackground (Color.red);  
        exitButton = new JButton ("EXIT", new ImageIcon ("App-x-icon.png"));  
        exitButton.setBackground (Color.gray);  
          
        stopButton.addActionListener (new ButtonListener());  
        playButton.addActionListener (new ButtonListener());  
        exitButton.addActionListener (new ButtonListener());  
          
        /\*------------------------------------------------------------------------------  
        Timer & Note Labels are made to show the countdown & notices  
        ------------------------------------------------------------------------------  
        \*/  
        timerLabel = new JLabel ("");  
        timerLabel.setFont (new Font ("Helvetica", Font.BOLD, 125));  
        noteLabel = new JLabel ("");  
        noteLabel.setFont (new Font ("Helvetica", Font.BOLD, 20));  
          
        /\*------------------------------------------------------------------------------  
        Adds in the components to be shown in GUI  
        ------------------------------------------------------------------------------  
        \*/  
        add (jeopardyHeading);  
        add (fiveSec);  
        add (thirtySec);  
        add (exitButton);  
        add (playButton);  
        add (stopButton);  
        add (timerLabel);  
        add (noteLabel);  
          
        /\*------------------------------------------------------------------------------  
        Display options  
        ------------------------------------------------------------------------------  
        \*/  
        setPreferredSize (new Dimension (700, 300));  
        setBackground (Color.cyan);  
    }  
      
    private class ButtonListener implements ActionListener  
    {  
      public void actionPerformed (ActionEvent event)  
      {  
         if (event.getSource() == playButton) {  
            if (fiveSec.isSelected()) {  
               /\*------------------------------------------------------------------------------  
                If play is clicked on and five sec is selected  
                \* Play music  
                \* Start time, set timer to 5, & show sec in Timer Label  
                \* Show nothing in Note Label  
                ------------------------------------------------------------------------------  
                \*/  
               music[1].play();  
               time.start();  
               timer = 5;  
               timerLabel.setText(String.valueOf(timer));  
               timerLabel.setFont (new Font ("Helvetica", Font.BOLD, 125));  
               noteLabel.setText("");  
               noteLabel.setFont (new Font ("Helvetica", Font.BOLD, 20));  
            }  
            else if (thirtySec.isSelected()) {  
               /\*------------------------------------------------------------------------------  
                If play is clicked on and thirty sec is selected  
                \* Play music  
                \* Start time, set timer to 30, & show sec in Timer Label  
                \* Show nothing in Note Label  
                ------------------------------------------------------------------------------  
                \*/  
               music[1].play();  
               time.start();  
               timer = 30;  
               timerLabel.setText(String.valueOf(timer));  
               timerLabel.setFont (new Font ("Helvetica", Font.BOLD, 125));  
               noteLabel.setText("");  
               noteLabel.setFont (new Font ("Helvetica", Font.BOLD, 20));  
            }  
        }     
          
        if (event.getSource() == stopButton) {  
            /\*------------------------------------------------------------------------------  
            If stop is clicked on  
            \* Put fiveSec & thirtySec Radio Buttons into Button Group and clear selection  
            \* Stop music  
            \* Stop time, set timer to 0, & show 0 in Timer Label  
            \* Show TIME STOPPED in Note Label  
            ------------------------------------------------------------------------------  
            \*/  
            ButtonGroup group = new ButtonGroup();  
            group.add (fiveSec);  
            group.add (thirtySec);  
            group.clearSelection();  
              
            music[1].stop();  
            time.stop();  
            timer = 0;  
            timerLabel.setText("0");  
            timerLabel.setFont (new Font ("Helvetica", Font.BOLD, 125));  
            noteLabel.setText("TIMER STOPPED");  
            noteLabel.setFont (new Font ("Helvetica", Font.BOLD, 20));  
        }  
          
        if (event.getSource() == exitButton) {  
            /\*------------------------------------------------------------------------------  
            If exit is clicked-closes program  
            ------------------------------------------------------------------------------  
            \*/  
            System.exit(0);  
        }  
      }  
    }  
      
    private class TimerListener implements ActionListener  
    {  
      public void actionPerformed (ActionEvent event)  
      {  
        if (timer != 0) {  
            /\*------------------------------------------------------------------------------  
            If timer is not 0  
            \* Subtract one from timer  
            \* Show timer in Timer Label  
            ------------------------------------------------------------------------------  
            \*/  
            timer = timer - 1;  
            timerLabel.setText(String.valueOf(timer));  
        }  
          
        if (timer == 0) {  
            /\*------------------------------------------------------------------------------  
            If timer is 0  
            \* Stop music  
            \* Show TIMER EXPIRED in Note Label  
            ------------------------------------------------------------------------------  
            \*/  
            music[1].stop();  
            noteLabel.setText("TIMER EXPIRED");  
            noteLabel.setFont (new Font ("Helvetica", Font.BOLD, 20));  
        }  
      }  
    }  
}