Timer and stopwatch project

```
package abc;

public class main {

	public static void main(String[] args) {


	Stopwatch stopwatch = new Stopwatch();

	}

      }
```

```
package abc;

import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

public class Stopwatch implements ActionListener{

JFrame frame = new JFrame();

JButton startButton = new JButton("START");

JButton resetButton = new JButton("RESET");

JLabel timeLabel = new JLabel();

int elapsedTime = 0;

int seconds =0;

int minutes =0;

int hours =0;

boolean started = false;

String seconds_string = String.format("%02d", seconds);

String minutes_string = String.format("%02d", minutes);
```

```
String hours_string = String.format("%02d", hours);

Timer timer = new Timer(1000, new ActionListener() {

public void actionPerformed(ActionEvent e) {

elapsedTime=elapsedTime+1000;

hours = (elapsedTime/3600000);

minutes = (elapsedTime/60000) % 60;

seconds = (elapsedTime/1000) % 60;

seconds_string = String.format("%02d", seconds);

minutes_string = String.format("%02d", minutes);

hours_string = String.format("%02d", hours);

timeLabel.setText(hours_string+":"+minutes_string+":"+seconds_string);

}

});

Stopwatch(){

timeLabel.setText(hours_string+":"+minutes_string+":"+seconds_string);

timeLabel.setBounds(100,100,200,100);

timeLabel.setFont(new Font("Verdana",Font.PLAIN,35));

timeLabel.setBorder(BorderFactory.createBevelBorder(1));

timeLabel.setOpaque(true);

timeLabel.setHorizontalAlignment(JTextField.CENTER);

startButton.setBounds(100,200,100,50);

startButton.setFont(new Font("Ink Free",Font.PLAIN,20));

startButton.setFocusable(false);

startButton.addActionListener(this);

resetButton.setBounds(200,200,100,50);

resetButton.setFont(new Font("Ink Free",Font.PLAIN,20));

resetButton.setFocusable(false);

resetButton.addActionListener(this);
```

```
frame.add(startButton);

frame.add(resetButton);

frame.add(timeLabel);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

frame.setSize(420,420);

frame.setLayout(null);

frame.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==startButton) {

if(started==false) {

started=true;

startButton.setText("STOP");

start();

}

else {

started=false;

startButton.setText("START");

stop();

}

}

if(e.getSource()==resetButton) {

started=false;

startButton.setText("START");

reset();

}

}
```

```
void start() {
timer.start();
}
void stop() {
timer.stop();
}
void reset() {
timer.stop();
elapsedTime=0;
seconds =0;
minutes=0;
hours=0;
seconds_string = String.format("%02d", seconds);
minutes_string = String.format("%02d", minutes);
hours_string = String.format("%02d", hours);
timeLabel.setText(hours_string+":"+minutes_string+":"+seconds_string);
}
}
```

```
package abc;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Timer;
public class TimerProject extends Frame {
private Label timerLabel;
private Button startButton;
```

```
private int secondsRemaining = 0;

private Timer countdownTimer;

public TimerProject() {

setTitle("Countdown Timer");

setSize(300, 150);

setLayout(new FlowLayout());

timerLabel = new Label("Time Remaining: " + secondsToTime(secondsRemaining));

timerLabel.setFont(new Font("Arial", Font.PLAIN, 20));

add(timerLabel);

startButton = new Button("Start Timer");

startButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

startButton.setEnabled(false);

secondsRemaining = 60; // Set the countdown time in seconds

updateTimerLabel();

countdownTimer = new Timer(1000, new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

secondsRemaining--;

updateTimerLabel();

if (secondsRemaining == 0) {

stopTimer();

}

}

});

countdownTimer.start();

}
```

```
});

add(startButton);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowClosing(java.awt.event.WindowEvent windowEvent) {

System.exit(0);

}

});

}

private String secondsToTime(int seconds) {

int mins = seconds / 60;

int secs = seconds % 60;

return String.format("%02d:%02d", mins, secs);

}

private void updateTimerLabel() {

timerLabel.setText("Time Remaining: " + secondsToTime(secondsRemaining));

}

private void stopTimer() {

countdownTimer.stop();

startButton.setEnabled(true);

updateTimerLabel();

}

public static void main(String[] args) {

TimerProject timerProject = new TimerProject();

timerProject.setVisible(true);

}

}
```