*/\*\*  
 \* Данная программа реаллизует графическую программу "таймер"  
 \* С использованием ActionListener-ов, класса awt и swing.  
 \* все комментарии были реаллизованы простими строчными объяснениями, не javadoc (для удобства).  
 \* Использование методов описано в коде достаточно подробно  
 \*  
 \** ***@author*** *Botsan Nicholas  
 \** ***@since*** *29.03.22  
 \** ***@version*** *3.0  
 \*  
 \*/*import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
  
public class timer implements ActionListener{  
 JFrame frame = new JFrame();  
 JButton startButton = new JButton("START");  
 JButton resetButton = new JButton("RESET");  
 JLabel timerLabel = new JLabel();  
 boolean isStarted = false;  
  
 int timeElapse=0;  
 int seconds=0;  
 int minutes=0;  
 int hours=0;  
  
  
 String secondsString = String.*format*("%02d", seconds);//2 - число позиций на число, 0 - заполнить пропуски 0-ми  
 String minutesString = String.*format*("%02d", minutes);  
 String hoursString = String.*format*("%02d", hours);  
  
  
  
  
 Timer timerr = new Timer(1000, new ActionListener() { //Если бы мы указали this, у нас бы вызывался тот же метод, что и для кнопок (Для ActionListener который имплементируется нашим классом)  
 @Override  
 public void actionPerformed(ActionEvent e){  
 timeElapse+=1000;//+1 sec //timeElapse отсчитывает, сколько миллисекунд прошло с начала запуска таймера, а мы уже далее расчитываем из этого времени минуты секунды и часы...  
 hours = (timeElapse/3600000);  
 minutes = (timeElapse/60000) % 60;  
 seconds = (timeElapse/1000) % 60;  
  
 //При обновлении числовых переменных их строки автоматически не изменятся => надо изменить их ручками  
 secondsString = String.*format*("%02d", seconds);  
 minutesString = String.*format*("%02d", minutes);  
 hoursString = String.*format*("%02d", hours);  
  
 timerLabel.setText(hoursString + ":" + minutesString + ":" + secondsString);//не обновляется сам, поэтому ты должен его обновить сам...  
 //Ты в лэйбел выводиль именно отформатированные стринги, поэтому если ты их не обновишь, секунды будут идти, а лейбел не будет обновляться  
  
  
 }  
 });//Через каждые 1000 миллисекунд будет срабатывать ActionListener, который вызвывает метод ActionPerformed, но не стандартный, а свой - переопределенный  
  
  
  
  
  
 public timer(){  
 int frameW = 320;  
 int frameH = 200;  
 Toolkit toolkit = Toolkit.*getDefaultToolkit*();  
 Dimension screen = toolkit.getScreenSize();  
  
 //toolkit.beep(); //Для звука при запуске окна  
  
  
  
  
  
 timerLabel.setText(hoursString + ":" + minutesString + ":" + secondsString);  
 timerLabel.setFont(new Font("Vergina", Font.*PLAIN*, 50));  
 timerLabel.setBounds(frameW/2 -158, frameH/2 - 97, 300, 97);  
  
 timerLabel.setBorder(BorderFactory.*createBevelBorder*(1));  
 timerLabel.setOpaque(true);  
 timerLabel.setHorizontalAlignment(JTextField.*CENTER*);  
  
  
  
  
  
  
  
  
  
 startButton.setBounds(frameW/2 -157, frameH/2+5, 148, 50);  
 startButton.setFont(new Font("Vergina", Font.*BOLD*, 30));  
 startButton.setFocusable(false); //Если его оставить, кнопка будет реагировать на клавиатуру и мышь  
 startButton.addActionListener(this);  
 //startButton.doClick();//появляется по клику  
  
  
 resetButton.setBounds(frameW/2 - 7, frameH/2+5, 148, 50);  
 resetButton.setFont(new Font("Vergina", Font.*BOLD*, 30));  
 resetButton.setFocusable(false);  
 resetButton.addActionListener(this);  
  
  
  
  
  
  
  
  
 frame.add(timerLabel);  
 frame.add(startButton);  
 frame.add(resetButton);  
  
 //Совет. Сначала настрой фрейм, по потом пиши настройку всех объектов перед фреймом (по есть настройка фрейма в конце). Проведи аналогию с проектом - ты же его сначала добавляешь в него текст, картинки, графики и лишь потом "афишируешь его". В Java так же...  
  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.setBounds(screen.width/2-frameW/2, screen.height/2-frameH/2, frameW, frameH);  
 frame.setLayout(null);  
 frame.setVisible(true);  
  
 frame.setTitle("Timer 2.0");  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e){  
 if(e.getSource() == startButton) Start();  
 else if(e.getSource() == resetButton) Reset();  
 }  
  
 public void Start(){  
 if(!isStarted){  
 isStarted = true;  
 startButton.setText("STOP");  
 timerr.start();  
 }  
 else  
 {  
 isStarted = false;  
 startButton.setText("START");  
 timerr.stop();  
 }  
 }  
  
 public void Reset(){  
 if(isStarted){  
  
 timeElapse = 0;  
 hours = 0;  
 minutes = 0;  
 seconds = 0;  
  
 secondsString = String.*format*("%02d", seconds);  
 minutesString = String.*format*("%02d", minutes);  
 hoursString = String.*format*("%02d", hours);  
  
 timerLabel.setText(hoursString + ":" + minutesString + ":" + secondsString);  
 }  
 else  
 {  
 timeElapse = 0;  
 hours = 0;  
 minutes = 0;  
 seconds = 0;  
  
 secondsString = String.*format*("%02d", seconds);  
 minutesString = String.*format*("%02d", minutes);  
 hoursString = String.*format*("%02d", hours);  
  
 timerLabel.setText(hoursString + ":" + minutesString + ":" + secondsString);  
 }  
 }  
  
}