

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ** **РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Уральский государственный экономический университет»**

**(УрГЭУ)**

**ЛАБОРАТОРНАЯ РАБОТА**

**по дисциплине** Объектно-Ориентированное Программирование

Тема 14. Графический пользовательский интерфейс, отслеживание событий

Институт непрерывного и дистанционного образования

Направление подготовки

09.03.01 Информатика и

вычислительная техника

Направленность (профиль)

Программное обеспечение

автоматизированных систем

Кафедра

Информационных технологий и

статистик

Студент

Кушниренко Дарья Александровна\_

Группа ИНО ЗБ ПОАС-23-1

Руководитель

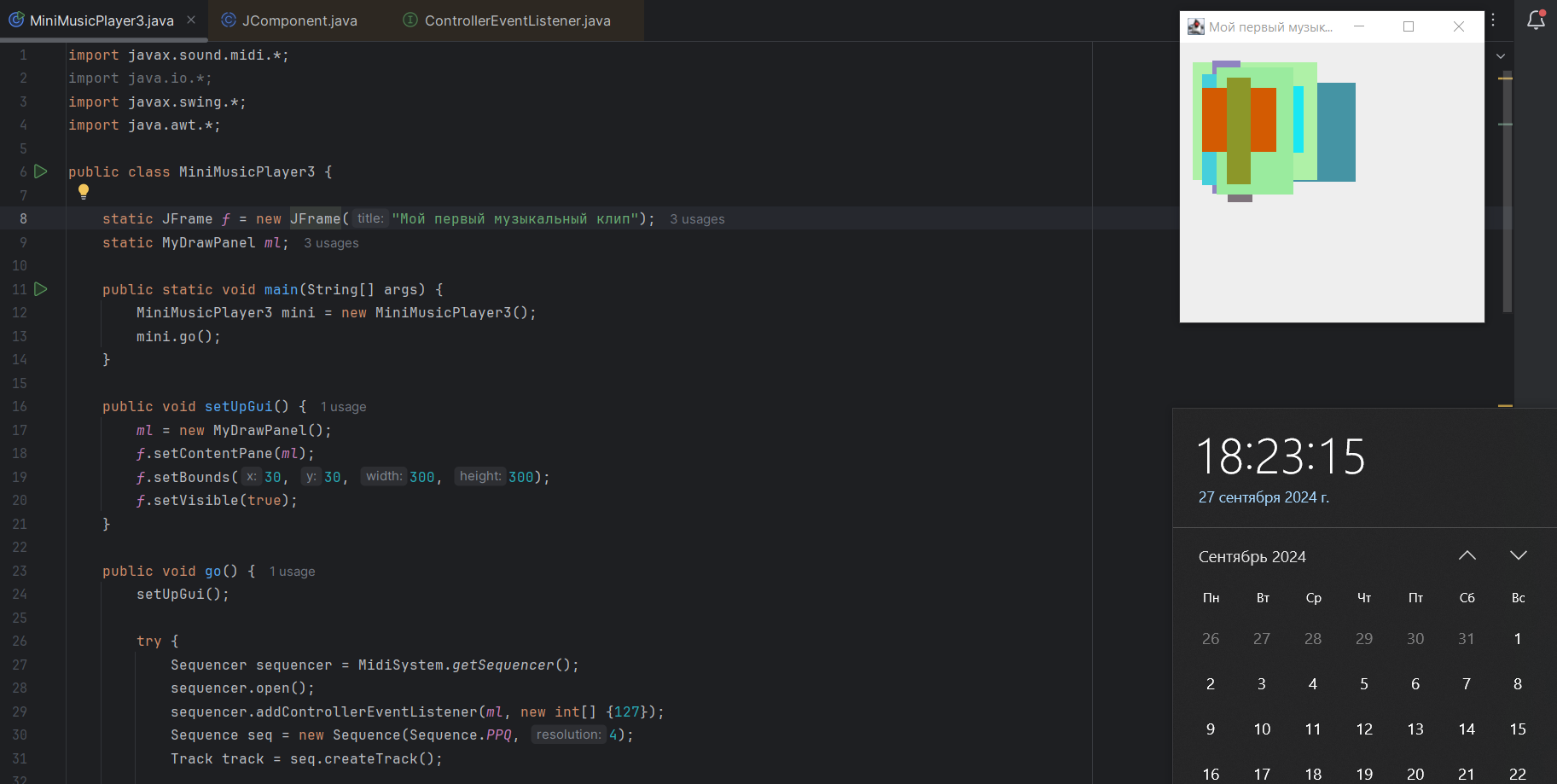
Панов Михаил Александрович

Екатеринбург  
2024 г.

**Кушниренко Д.А., группа: ИНО ЗБ ПОАС-23-1**

**Задание 1**

import javax.sound.midi.\*;  
import java.io.\*;  
import javax.swing.\*;  
import java.awt.\*;  
  
public class MiniMusicPlayer3 {  
  
 static JFrame *f* = new JFrame("Мой первый музыкальный клип");  
 static MyDrawPanel *ml*;  
  
 public static void main(String[] args) {  
 MiniMusicPlayer3 mini = new MiniMusicPlayer3();  
 mini.go();  
 }  
  
 public void setUpGui() {  
 *ml* = new MyDrawPanel();  
 *f*.setContentPane(*ml*);  
 *f*.setBounds(30, 30, 300, 300);  
 *f*.setVisible(true);  
 }  
  
 public void go() {  
 setUpGui();  
  
 try {  
 Sequencer sequencer = MidiSystem.*getSequencer*();  
 sequencer.open();  
 sequencer.addControllerEventListener(*ml*, new int[] {127});  
 Sequence seq = new Sequence(Sequence.*PPQ*, 4);  
 Track track = seq.createTrack();  
  
 int r = 0;  
 for (int i = 0; i < 60; i += 4) {  
  
 r = (int) ((Math.*random*() \* 50) + 1);  
 track.add(makeEvent(144, 1, r, 100, i));  
 track.add(makeEvent(176, 1, 127, 0, i));  
 track.add(makeEvent(128, 1, r, 100, i + 2));  
 } //Конец цикла  
 sequencer.setSequence(seq);  
 sequencer.start();  
 sequencer.setTempoInBPM(120);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
  
 public MidiEvent makeEvent(int comd, int chan, int one, int two, int tick) {  
 MidiEvent event = null;  
 try {  
 ShortMessage a = new ShortMessage();  
 a.setMessage(comd, chan, one, two);  
 event = new MidiEvent(a, tick);  
  
 } catch (Exception e) {  
 }  
 return event;  
 } //Закрываем метод  
  
 class MyDrawPanel extends JPanel implements ControllerEventListener {  
 boolean msg = false;  
  
 public void controlChange(ShortMessage event) {  
 msg = true;  
 repaint();  
 }  
  
 public void paintComponent(Graphics g) {  
 if (msg) {  
  
 Graphics2D g2 = (Graphics2D) g;  
 int r = (int) (Math.*random*() \* 250);  
 int gr = (int) (Math.*random*() \* 250);  
 int b = (int) (Math.*random*() \* 250);  
  
 g.setColor(new Color(r, gr, b));  
  
 int ht = (int) ((Math.*random*() \* 120) + 10);  
 int width = (int) ((Math.*random*() \* 120) + 10);  
  
 int x = (int) ((Math.*random*() \* 40) + 10);  
 int y = (int) ((Math.*random*() \* 40) + 10);  
  
 g.fillRect(x, y, ht, width);  
 msg = false;  
 }  
 }  
 }  
}

****