

Министерство науки и высшего образования РФ
ФГБОУ ВО «Омский государственный технический университет»
Кафедра «Автоматизированные системы обработки информации и
управления»

ОТЧЁТ по лабораторной
работе №4

По дисциплине: Объектно-ориентированное программирование
студента Мархилия Фёдора группы ПИН-212

Пояснительная записка
Шифр работы От-2068998-43-ПИН-212-10 ПЗ
Специальность 09.03.04

Старший преподаватель

А.А. Кабанов

Студент

Ф. Е. Мархиль

Омск 2022

Задание

Промоделировать аналоговые часы (со стрелками) с кнопками для увеличения/уменьшения времени на час/минуту.

Цель работы

Изучить построение приложений с использованием графического интерфейса библиотек java.awt и javax.swing.

Ход работы

Импортирую нужные классы.

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.GridLayout;
import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
```

Создаю класс Clock, создаю переменные и изменяю функцию-конструктор. Создаю методы для изменения и чтения переменных.

```
6 usages
class Clock {
    2 usages
    private int hours;
    2 usages
    private int minutes;

    1 usage
    public Clock(int hours, int minutes) { //функция-конструктор
        this.setHours(hours);
        this.setMinutes(minutes);
    }

    2 usages
    public void setMinutes(int minutes) {
        this.minutes = minutes % 60;
    }

    3 usages
    public int getMinutes() {
        return minutes;
    }

    2 usages
    public void setHours(int hours) {
        this.hours = hours % 24;
    }

    2 usages
    public int getHours() {
        return hours;
    }
}
```

Класс Main наследует свойства класса JPanel. Также я создаю объект класса Clock.

```
2 usages
public class Main extends JPanel { //extends наследует свойства класса

7 usages
    private Clock clock;
```

В функции-конструкторе указываю размер окна и устанавливаю время по умолчанию.

```
1 usage
public Main(Clock clock) {
    setPreferredSize(new Dimension( width: 600, height: 400));
    setClock(clock);
}
```

Создаю метод для отрисовки.

```
@Override
public void paint(Graphics g) {
    super.paint(g); //для правильной отрисовки
    if (clock != null) {
        drawClock(g, clock);
    }
}
```

Далее идет метод, который возвращает точку, в которой линия должна заканчиваться.

```
3 usages
private Point getEndPoint(double angle, int radius) {
    Point o = new Point( x: getSize().width/2, y: getSize().height / 2);
    int x = (int) (o.x + radius * Math.cos(angle));
    int y = (int) (o.y - radius * Math.sin(angle));
    return new Point(x, y);
}
```

Основной метод, который рисует стрелки на основе времени. Мы рассчитываем радиус стрелок, определяем угол часовой стрелки на основе часов и минут и угол минутной стрелки на основе минут. Также прорисовываем их.

```

1 usage
private void drawClock(Graphics g, Clock clock) {
    Point o = new Point( x: getSize().width / 2, y: getSize().height / 2);
    int radiusClock = Math.min(o.x, o.y) - 20;
    int radiusMinute = radiusClock - 10;
    int radiusHour = radiusMinute - 20;
    double angle;

    angle = Math.PI / 2 - (clock.getHours() + clock.getMinutes() / 60.0) * Math.PI / 6;
    Point point = getEndPoint(angle, radiusHour);
    g.setColor(Color.GREEN);
    g.drawLine(o.x, o.y, point.x, point.y);

    angle = Math.PI / 2 - clock.getMinutes() * Math.PI / 30;
    point = getEndPoint(angle, radiusMinute);
    g.setColor(Color.GRAY);
    g.drawLine(o.x, o.y, point.x, point.y);
}

```

Дальше идет основной метод класса, в котором мы указываем параметры создания окна, создаем панель для кнопок, а также сами кнопки.

Указываем местоположение панели и включаем видимость. Также для кнопок часов и минут мы создаем слушателей(уведомляемый о событии объект) и создаем реакцию на нажатие кнопок.

```

public static void main(String[] args) throws Exception {
    JFrame f = new JFrame( title: "Часы");
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setLayout(new BorderLayout());

    final Main clockPanel = new Main(new Clock( hours: 0, minutes: 0));
    clockPanel.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY, thickness: 2));
    f.add(clockPanel, BorderLayout.CENTER);
    JPanel controlPanel = new JPanel();
    controlPanel.setBorder(BorderFactory.createTitledBorder("Настройки"));
    controlPanel.setLayout(new GridLayout( rows: 2, cols: 2, hgap: 6, vgap: 6));

    JButton but = (JButton) controlPanel.add(new JButton( text: "Часы +1"));
    but.putClientProperty("plus", true);
    ActionListener alHours = new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            JButton b = (JButton) e.getSource();
            boolean plus = ((Boolean) b.getClientProperty( key: "plus")).booleanValue();
            clockPanel.getClock().setHours(clockPanel.getClock().getHours() + (plus ? 1 : -1));
            clockPanel.repaint();
        }

    };
    but.addActionListener(alHours);

    but = (JButton) controlPanel.add(new JButton( text: "Часы -1"));
    but.putClientProperty("plus", false);
    but.addActionListener(alHours);

    but = (JButton) controlPanel.add(new JButton( text: "Минуты +1"));
    but.putClientProperty("plus", true);
    ActionListener alMinutes = new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            JButton b = (JButton) e.getSource();
            boolean plus = ((Boolean) b.getClientProperty( key: "plus")).booleanValue();
            clockPanel.getClock().setMinutes(clockPanel.getClock().getMinutes() + (plus ? 1 : -1));
        }

    };
    but.addActionListener(alMinutes);
}

```

```

        @Override
        public void actionPerformed(ActionEvent e) {
            JButton b = (JButton) e.getSource();
            boolean plus = ((Boolean) b.getClientProperty( key: "plus")).booleanValue();
            clockPanel.getClock().setMinutes(clockPanel.getClock().getMinutes() + (plus ? 1 : -1));
            clockPanel.repaint();
        }

};
but.addActionListener(alMinutes);

but = (JButton) controlPanel.add(new JButton( text: "Минуты -1"));
but.putClientProperty("plus", false);
but.addActionListener(alMinutes);

f.add(controlPanel, BorderLayout.SOUTH);
f.pack();
f.setVisible(true);
}

```

Код и приложение:

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.GridLayout;
import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

class Clock {
    private int hours;
    private int minutes;

    public Clock(int hours, int minutes) { //функция-конструктор
        this.setHours(hours);
        this.setMinutes(minutes);
    }

    public void setMinutes(int minutes) {
        this.minutes = minutes % 60;
    }

    public int getMinutes() {
        return minutes;
    }

    public void setHours(int hours) {
        this.hours = hours % 24;
    }

    public int getHours() {
        return hours;
    }
}

```

```

public class Main extends JPanel { //extends наследует свойства класса

    private Clock clock;

    public Main(Clock clock) {
        setPreferredSize(new Dimension(600, 400));
        setClock(clock);
    }

    @Override
    public void paint(Graphics g) {
        super.paint(g); //для правильной отрисовки
        if (clock != null) {
            drawClock(g, clock);
        }
    }

    // public Point convertCoordinatesToPanel(Point point) {
    //     Point O = new Point(getSize().width / 2, getSize().height / 2);
    //     return new Point(O.x + point.x, O.y - point.y);
    // }

    private Point getEndPoint(double angle, int radius) {
        Point O = new Point(getSize().width/2, getSize().height / 2);
        int x = (int) (O.x + radius * Math.cos(angle));
        int y = (int) (O.y - radius * Math.sin(angle));
        return new Point(x, y);
    }

    private void drawClock(Graphics g, Clock clock) {
        Point O = new Point(getSize().width / 2, getSize().height / 2);
        int radiusClock = Math.min(O.x, O.y) - 20;
        int radiusMinute = radiusClock - 10;
        int radiusHour = radiusMinute - 20;
        double angle;

        angle = Math.PI / 2 - (clock.getHours() + clock.getMinutes() / 60.0)
* Math.PI / 6;
        Point point = getEndPoint(angle, radiusHour);
        g.setColor(Color.GREEN);
        g.drawLine(O.x, O.y, point.x, point.y);

        angle = Math.PI / 2 - clock.getMinutes() * Math.PI / 30;
        point = getEndPoint(angle, radiusMinute);
        g.setColor(Color.GRAY);
        g.drawLine(O.x, O.y, point.x, point.y);
    }

    public Clock getClock() {
        return clock;
    }

    public void setClock(Clock clock) {
        this.clock = clock;
    }

    public static void main(String[] args) throws Exception {
        JFrame f = new JFrame("Часы");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setLayout(new BorderLayout());

        final Main clockPanel = new Main(new Clock(0, 0));
        clockPanel.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY,
2));
    }
}

```

```

        f.add(clockPanel, BorderLayout.CENTER);
        JPanel controlPanel = new JPanel();

        controlPanel.setBorder(BorderFactory.createTitledBorder("Настройки"));
        controlPanel.setLayout(new GridLayout(2, 2, 6, 6));

        JButton but = (JButton) controlPanel.add(new JButton("Часы +1"));
        but.putClientProperty("plus", true);
        ActionListener alHours = new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                JButton b = (JButton) e.getSource();
                boolean plus = ((Boolean)
b.getClientProperty("plus")).booleanValue();

                clockPanel.getClock().setHours(clockPanel.getClock().getHours() + (plus ? 1 :
-1));

                clockPanel.repaint();
            }

        };
        but.addActionListener(alHours);

        but = (JButton) controlPanel.add(new JButton("Часы -1"));
        but.putClientProperty("plus", false);
        but.addActionListener(alHours);

        but = (JButton) controlPanel.add(new JButton("Минуты +1"));
        but.putClientProperty("plus", true);
        ActionListener alMinutes = new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                JButton b = (JButton) e.getSource();
                boolean plus = ((Boolean)
b.getClientProperty("plus")).booleanValue();

                clockPanel.getClock().setMinutes(clockPanel.getClock().getMinutes() + (plus ?
1 : -1));

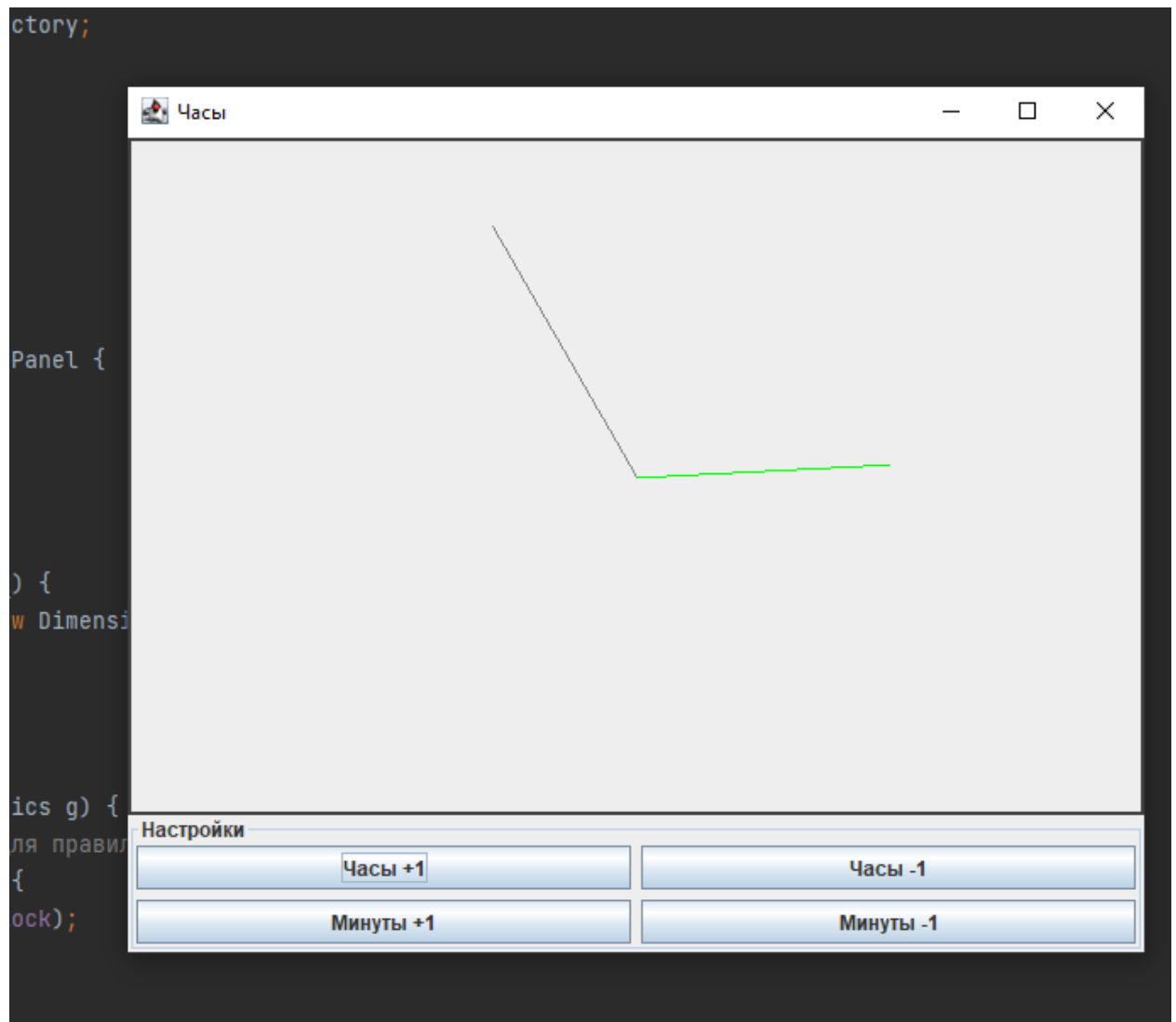
                clockPanel.repaint();
            }

        };
        but.addActionListener(alMinutes);

        but = (JButton) controlPanel.add(new JButton("Минуты -1"));
        but.putClientProperty("plus", false);
        but.addActionListener(alMinutes);

        f.add(controlPanel, BorderLayout.SOUTH);
        f.pack();
        f.setVisible(true);
    }
}

```



Вывод

В ходе выполнения данной работы я получил опыт работы с графическим интерфейсом библиотек `java.awt` и `javax.swing`.