



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения

**ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
по дисциплине
«Программирование на языке Java»**

**Тема: Создание программ с графическим интерфейсом пользователя на
Java.**

Выполнил студент группы ИКБО-16-20

Пак С.А.

Принял ассистент кафедры ИиППО

Русляков А.А.

Практические работы выполнены «_____» 2021г.

«Зачтено» «_____» 2021г

Москва 2021

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ.....	3
1. Общие сведения.....	3
2. JTextField.....	3
3. JTextArea.....	3
4. Layout менеджеры.....	3
4.1. BorderLayout.....	3
4.2. GridLayout.....	3
4.3. Null Layout.....	3
5. Меню.....	4
РЕШЕНИЕ ЗАДАЧИ.....	5
1. Постановка задачи.....	5
2. Программный код.....	5
3. Вывод программы.....	7
ВЫВОД.....	9

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

1. Общие сведения

Для создания графического интерфейса пользователя можно использовать стандартную Java библиотеку Swing или AWT. В этих библиотеках имеются различные классы, позволяющие создавать окна, кнопки, текстовые поля, меню и другие объекты.

2. JTextField

JTextFields - текстовое поле или поля для ввода текста (можно ввести только одну строку). Примерами текстовых полей являются поля для ввода логина и пароля, например, используемые, при входе в электронную почту.

```
JTextField jta = new JTextField(10)
```

В параметре конструктора задано число 10, это количество символов, которые могут быть видны в текстовом поле.

3. JTextArea

Компонент TextAreas похож на TextFields, но в него можно вводить более одной строки. В конструкторе можно передать количество рядов и столбцов для поля.

4. Layout менеджеры

4.1. BorderLayout

Разделяет компонент на 5 областей (WEST, EAST, NORTH, SOUTH и Center).

Метод для добавления в контейнер, который есть у менеджера BorderLayout отличается и выглядит следующим образом:

```
add( comp , BorderLayout.EAST)
```

Обратите внимание, что мы можем например добавить панели JPanel в эти области и затем добавлять компоненты этих панелей. Мы можем установить расположение этих JPanel используя другие менеджеры.

4.2. GridLayout

С помощью менеджера GridLayout компонент может принимать форму таблицы, где можно задать число строк и столбцов.

4.3. Null Layout

Иногда бывает нужно изменить размер и расположение компонента в контейнере. Таким образом, мы должны указать программе не использовать никакой менеджер компоновки, то есть

```
setLayout (null)
```

5. Меню

Добавление меню в программе Java проста. Java определяет три компонента:

- JMenuBar: который представляет собой компонент, который содержит меню;
- JMenu: который представляет меню элементов для выбора;
- JMenuItem: представляет собой элемент, который можно кликнуть из меню.

РЕШЕНИЕ ЗАДАЧИ

1. Постановка задачи

Вариант: 3

Задание: создать окно, реализовать анимацию, с помощью картинки, состоящей из нескольких кадров.

2. Программный код

Файл App.java:

```
package ru.mirea;

import java.io.File;
import java.io.IOException;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JButton;
import javax.swing.ImageIcon;

import javax.imageio.ImageIO;

public class App extends JFrame {
    private static final int WINDOW_WIDTH = 1400;
    private static final int WINDOW_HEIGHT = 900;

    private static final String IMG = "/home/ezhik/Documents/Java-MIREA/labs/Java-MIREA-lab-5/imgs/sekiro.jpg";

    /**
     * Создаёт окно приложения
     */
    private App() throws IOException {
        super("Application");

        setSize(WINDOW_WIDTH, WINDOW_HEIGHT);
        setLayout(null);

        BufferedImage image = ImageIO.read(new File(IMG));
        JLabel imageLabel = new JLabel(new ImageIcon(image));
```

```

int imageX = (WINDOW_WIDTH - image.getWidth()) / 2;
int imageY = (WINDOW_HEIGHT - image.getHeight()) / 2;

imageLabel.setLocation(imageX, imageY); //
расположить картинку в центре окна
imageLabel.setSize(image.getWidth(), image.getHeight());

JButton moveLeft = new JButton("Сдвинуть влево");
JButton moveRight = new JButton("Сдвинуть вправо");

moveLeft.addActionListener(new ActionListener() {
    /**
     * Сдвигает картинку влево на 5 пикселей
     * @param e событие
     */
    @Override
    public void actionPerformed(ActionEvent e) {
        imageLabel.setLocation(imageLabel.getX() - 5,
imageLabel.getY());
    }
});

moveRight.addActionListener(new ActionListener() {
    /**
     * Сдвигает картинку вправо на 5 пикселей
     * @param e событие
     */
    @Override
    public void actionPerformed(ActionEvent e) {
        imageLabel.setLocation(imageLabel.getX() + 5,
imageLabel.getY());
    }
});

moveLeft.setSize(160, 25);
moveRight.setSize(160, 25);

moveLeft.setLocation(15, WINDOW_HEIGHT - 60);
moveRight.setLocation(WINDOW_WIDTH - 15 -
moveRight.getWidth(), WINDOW_HEIGHT - 60);

add(imageLabel);
add(moveLeft);
add(moveRight);
}

public static void main(String[] args) throws IOException {

```

```

App win = new App();

win.setVisible(true);
win.setDefaultCloseOperation(EXIT_ON_CLOSE);
}
}

```

3. Вывод программы

По нажатии одной из кнопок картинка сдвигается на 5 пикселей либо вправо, либо влево (рис.1, 2).

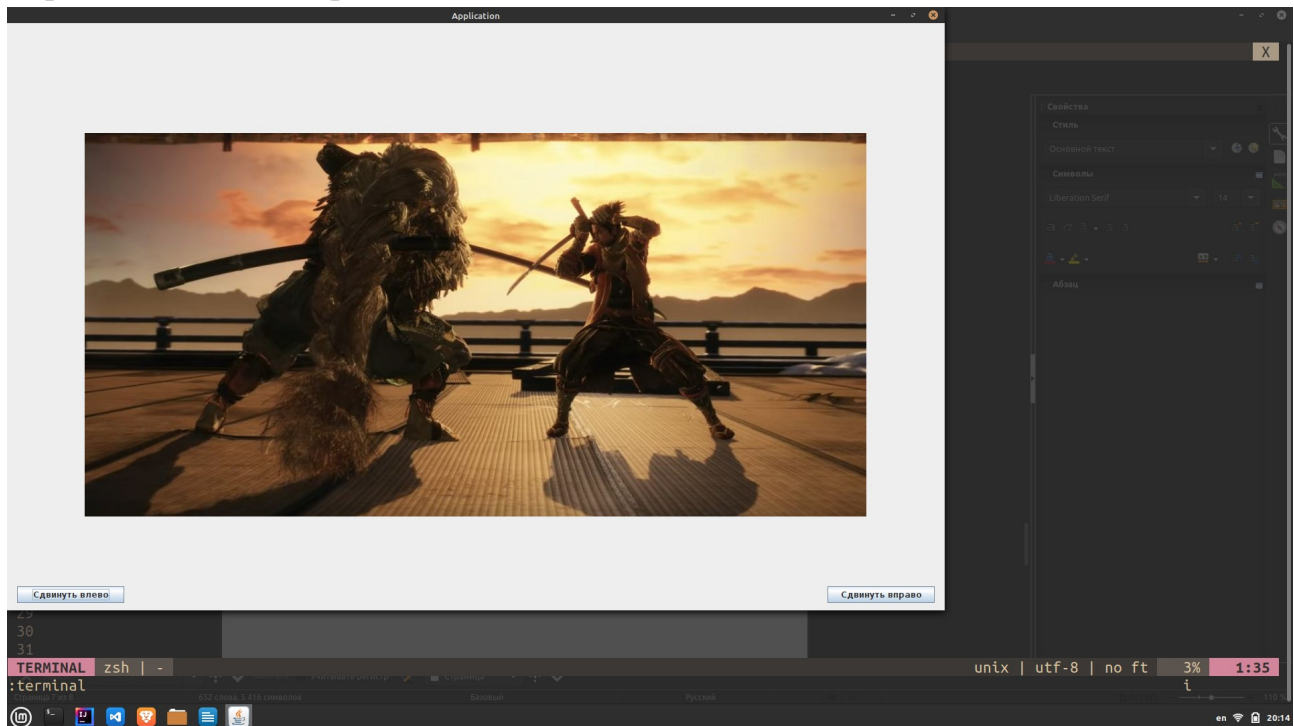


Рис.1 Выходные данные программы ч.1

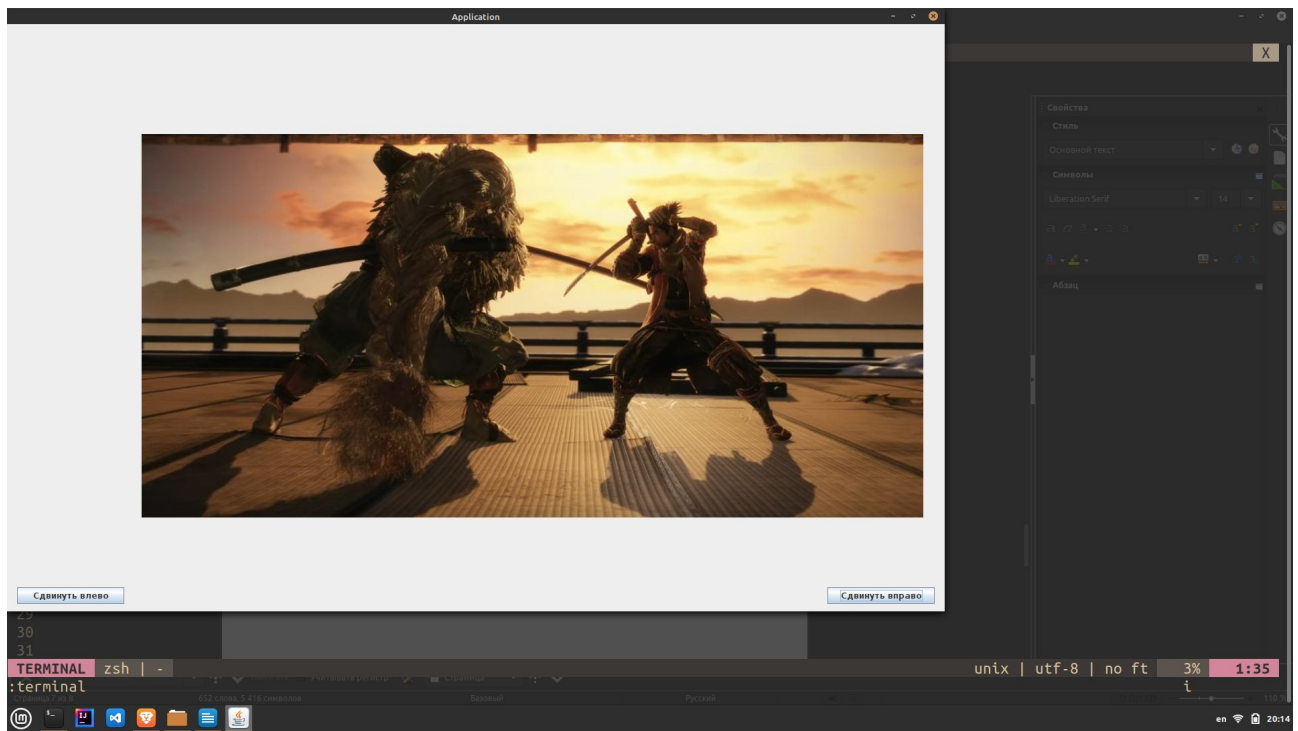


Рис.2 Выходные данные программы ч.2

ВЫВОД

В ходе выполнения научился создавать графический интерфейс пользователя, освоил на практике работу с различными объектами для создания ГИП, менеджерами размещения компонентов.