**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий Кафедра Информатики и информационных технологий***

# направление подготовки

**09.03.02 «Информационные системы и технологии»**

ЛАБОРАТОРНАЯ РАБОТА № 7

**Дисциплина:** Технология кроссплатформенного программирования

**Тема:** Создание программ с графическим интерфейсом пользователя на языке Java. Классы пакета Swing.

# Выполнил(а): студент(ка) группы

Хайруллина Алина Ринатовна

(Фамилия И.О.)

**Дата, подпись**

(Дата) (Подпись)

**Проверил:**

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись**

(Дата) (Подпись)

# Замечания:

**Москва 2023**

# ЛАБОРАТОРНАЯ РАБОТА № 7

**Задачи: I. Обязательная задача для всех:** Выписать все классы и методы, использованные в примере, и их описание из справки, и еще несколько полезных методов для каждого класса.

* BorderLayout – размещает элементы в один из пяти регионов (см. рис. 5), указанный при добавлении элемента в контейнер: наверх, вниз, влево, вправо, в центр (или север, юг, запад, восток, центр ); при этом, если в какой-либо регион не был добавлен элемент, то этот регион не отображается в окне;
* FlowLayout – размещает элементы по порядку в том же направлении, что и ориентация контейнера (слева направо по умолчанию), применяя один из пяти видов выравнивания, указанного при создании менеджера (по центру по умолчанию). Данный менеджер используется по умолчанию BorderLayout. в большинстве контейнерах;
* GridLayout – размещает элементы таблично. Количество столбцов и строк указывается при создании менеджера. По умолчанию одна строка, а число столбцов равно числу элементов;
* BoxLayout – размещает элементы по вертикали или по горизонтали. Обычно он используется не напрямую, а через контейнерный класс Box, который имеет дополнительные возможности;
* CardLayout – размещает элементы в виде колоды карт, т.е. в текущий момент может быть активным и видимым только один элемент (который лежит наверху колоды).

# JPanel – класс для работы с панелями. Панель – это своеобразный контейнер, к которому можно применить нужный компоновщик и поместить туда необходимые элементы, причем саму панель видно не будет, если только не установить специальные параметры.

# JButton – кнопка.

# JCheckBox – кнопка-флажок;

# JComboBox – выпадающий список;

# JLabel – метка, надпись;

# JList – список;

# JPasswordField – текстовое поле для скрытого ввода;

# JProgressBar – компонент для отображения числа в некотором диапазоне;

# JRadioButton – переключатели, радио-кнопки, обычно используются с компонентом ButtonGroup;

# JSlider – компонент, позволяющий выбрать значение из заданного диапазона;

# JSpinner – компонент, позволяющий выбрать значение из указанной последовательности;

# JTable – таблица;

# JTextField – однострочное текстовое поле;

# JTextArea – многострочное текстовое поле;

# JTree – дерево элементов (иерархически упорядоченных).

1. Классы и методы из AWT и Swing:

* JFrame: Это класс из библиотеки Swing, представляющий основное окно приложения.
  + setDefaultCloseOperation(int operation): Устанавливает операцию, которая будет выполнена при закрытии окна.
  + setSize(int width, int height): Устанавливает размер окна.
  + add(Component comp, Object constraints): Добавляет компонент в контейнер (окно).
  + setVisible(boolean b): Делает окно видимым или невидимым.
* JPanel: Это класс из библиотеки Swing, представляющий панель для размещения компонентов.
  + setLayout(LayoutManager manager): Устанавливает менеджер компоновки для панели.
  + add(Component comp): Добавляет компонент на панель.
* JButton: Это класс из библиотеки Swing, представляющий кнопку.
  + setText(String text): Устанавливает текст на кнопке.
* JLabel: Это класс из библиотеки Swing, представляющий метку (надпись).
  + setText(String text): Устанавливает текст метки.
* JCheckBox: Это класс из библиотеки Swing, представляющий флажок.
  + setText(String text): Устанавливает текст флажка.
* JRadioButton: Это класс из библиотеки Swing, представляющий радиокнопку.
  + setText(String text): Устанавливает текст радиокнопки.
* GridLayout: Это класс из библиотеки AWT, представляющий менеджер компоновки для размещения компонентов в таблице.
  + GridLayout(int rows, int cols): Создает объект GridLayout с указанным числом строк и столбцов.
* BoxLayout: Это класс из библиотеки Swing, представляющий менеджер компоновки для расположения компонентов в одной строке или одном столбце.
  + BoxLayout(Container target, int axis): Создает объект BoxLayout для указанного контейнера и оси (по вертикали или горизонтали).
* Box: Это класс из библиотеки Swing, представляющий контейнер для компонентов, позволяющий управлять их расположением.
  + createVerticalStrut(int height): Создает распорку (пустое пространство) вертикального размера.
  + createHorizontalStrut(int width): Создает распорку (пустое пространство) горизонтального размера.
  + createVerticalGlue(): Создает пружину для увеличения расстояния между компонентами.
* Component: Это базовый класс для всех компонентов AWT и Swing.

1. Дополнительные полезные методы:

* pack(): Автоматически подбирает оптимальный размер окна на основе размеров его компонентов.
* setMinimumSize(Dimension minimumSize): Устанавливает минимальный размер окна.
* setSelected(boolean selected): Устанавливает выбранным или снимает выбор с компонента.
* setLayout(LayoutManager manager): Устанавливает менеджер компоновки для контейнера.

**Индивидуальные задания** (номер получается путем деления по модулю 7 и последующим прибавлением 1, номера из списка группы). Создать интерфейс по изображению (использовать разные виды компоновок, если одинаковых объектов больше 3, то использовать массив

Листинг (Вариант 1, 25 в списке – 7\*4 = 24, 24+1 = 25)

import java.awt.BorderLayout;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import javax.swing.JCheckBox;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JRadioButton;

import javax.swing.JSlider;

import javax.swing.JSpinner;

import javax.swing.JTextArea;

import javax.swing.JTextField;

class Program {

public static void addTextFields(JFrame jfr) { GridLayout gridLayout = new GridLayout(2, 3); JPanel textFieldsPanel = new JPanel(); textFieldsPanel.setLayout(gridLayout);

// textFieldsPanel.setSize(20, 10);

JTextField textFieldArray[] = new JTextField[6]; for (int i = 0; i < 6; i++) {

textFieldArray[i] = new JTextField("Текстовое поле " + (i + 1), 0);

}

for (JTextField i : textFieldArray) { textFieldsPanel.add(i);

}

jfr.add(textFieldsPanel, BorderLayout.NORTH);

}

public static void addLeftPanel(JFrame jfr) {

// centerPanel.

JPanel centerPanel = new JPanel();

GridLayout gridLayout2 = new GridLayout(10, 1); centerPanel.setLayout(gridLayout2);

JPanel radioButtonPanel = new JPanel(); radioButtonPanel.setLayout(new BorderLayout());

GridLayout gridLayout = new GridLayout(9, 1); JPanel leftPanel = new JPanel(); leftPanel.setLayout(new BorderLayout(20, 0));

JPanel checkBoxPanel = new JPanel(); checkBoxPanel.setLayout(gridLayout);

JCheckBox checkBoxArray[] = new JCheckBox[9]; for (int i = 0; i < 9; i++) {

checkBoxArray[i] = new JCheckBox("Выбор " + (i + 1));

}

for (JCheckBox i : checkBoxArray) { checkBoxPanel.add(i);

}

JPanel multyPanel = new JPanel(); multyPanel.setLayout(new FlowLayout());

JComboBox comboBox = new JComboBox(); multyPanel.add(comboBox);

JSpinner spinner = new JSpinner(); multyPanel.add(spinner);

JRadioButton radioButton1 = new JRadioButton("Радиокнопка 1"); multyPanel.add(radioButton1);

JRadioButton radioButton2 = new JRadioButton("Радиокнопка 2"); radioButtonPanel.add(radioButton2, BorderLayout.CENTER);

centerPanel.add(multyPanel); centerPanel.add(radioButtonPanel);

leftPanel.add(checkBoxPanel, BorderLayout.WEST); leftPanel.add(centerPanel, BorderLayout.EAST);

jfr.add(leftPanel, BorderLayout.WEST);

}

public static void addRightTextPanel(JFrame jfr) { JPanel rightTextPanel = new JPanel(); rightTextPanel.setLayout(new BorderLayout()); JTextArea textArea = new JTextArea(20, 25); textArea.setText("Текстовая область"); textArea.setLineWrap(true); textArea.setWrapStyleWord(true); rightTextPanel.add(textArea, BorderLayout.EAST); jfr.add(rightTextPanel, BorderLayout.EAST);

}

public static void addSlider(JFrame jfr) { JPanel sliderPanel = new JPanel(); sliderPanel.setLayout(new BorderLayout()); JSlider slider = new JSlider();

slider.setOrientation(JSlider.HORIZONTAL); sliderPanel.add(slider, BorderLayout.SOUTH); jfr.add(sliderPanel, BorderLayout.SOUTH);

}

public static void main(String[] args) {

JFrame frame = new JFrame("MyWindow"); frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// fix size frame.setSize(630, 400); frame.setResizable(false);

addTextFields(frame); addRightTextPanel(frame); addLeftPanel(frame); addSlider(frame);

frame.setVisible(true);

}

}

