Факультет ИКСС

**Лабораторная работа №2**

Курс: 3-ий

Группа: ИКПИ-42

Дисциплина: Теория автоматов и формальных языков

Выполнил: Топорков Н. Ю.

Преподаватель: Выборнова А. И.

# Постановка задачи

Даны три кнопки:



Реализовать схему конечного автомата:



# Снимки программы

|  |  |  |
| --- | --- | --- |
| Исходное состояние | (Выкл)--ВВ->(Вкл) | (Вкл)--a->(А) |
| (А)--a->( Вкл) | (Вкл)--c->(C) | (С)--c->(Вкл) |
| (Вкл)--ВВ->( Выкл) | (А)--ВВ->( Выкл) | (С)--ВВ->( Выкл) |
| (А)--c->(С) | (С)--а->(А) |  |

# Код программы

## Main.java

**package** ru.eninja.flang;  
  
**import** javax.swing.\*;  
  
*/\*\*  
 \* Точка запуска программы  
 \*/***public class** Main {  
 **public static void** main(String[] args) {  
 SwingUtilities.invokeLater(Frame::**new**);  
 }  
}

## Frame.java

**package** ru.eninja.flang;  
  
**import** com.seaglasslookandfeel.SeaGlassLookAndFeel;  
  
**import** javax.swing.\*;  
**import** java.awt.\*;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
  
  
*/\*\*  
 \* Класс формы  
 \*/***public class** Frame **extends** JFrame {  
  
 **private** JLabel currentStateLbl;  
 **private** JButton ioBtn;  
 **private** JButton aBtn;  
 **private** JButton cBtn;  
  
 **private** Switcher switcher;  
 **private** String currentState;  
  
 */\*\*  
 \* Конструктор окна  
 \*/* **public** Frame() {  
 **super**(**""**);  
 setTheme();  
 initWindow();  
 initComponents();  
 setFont();  
 initListeners();  
 addComponentsToFrame();  
 pack();  
 setVisible(**true**);  
 }  
  
 */\*\*  
 \* Установка темы окна  
 \*/* **private void** setTheme() {  
 **try** {  
 UIManager.setLookAndFeel(**new** SeaGlassLookAndFeel());  
 SwingUtilities.updateComponentTreeUI(**this**);  
 } **catch** (UnsupportedLookAndFeelException e) {  
 e.printStackTrace();  
 }  
 }  
  
 */\*\*  
 \* Инициализация окна  
 \*/* **private void** initWindow() {  
 setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);  
 setLayout(**new** BorderLayout());  
 setLocationRelativeTo(**null**);  
 setResizable(**false**);  
 }  
  
 */\*\*  
 \* Инициализация компонентов, которые будут распологаться на окне  
 \*/* **private void** initComponents() {  
 currentStateLbl = **new** JLabel(**"Текущее состояние"**);  
 ioBtn = **new** JButton(**"I/O"**);  
 aBtn = **new** JButton(**"A"**);  
 cBtn = **new** JButton(**"C"**);  
 }  
  
 */\*\*  
 \* Установка шрифта  
 \*/* **private void** setFont() {  
 **try** {  
 String fontFileName = **"OpenSans-Regular.ttf"**;  
 **float** fontSize = 18;  
  
 InputStream fontStream = getClass().getClassLoader().getResourceAsStream(fontFileName);  
 Font font = Font.createFont(Font.TRUETYPE\_FONT, fontStream);  
 font = font.deriveFont(Font.PLAIN, fontSize);  
  
 currentStateLbl.setFont(font);  
 ioBtn.setFont(font);  
 aBtn.setFont(font);  
 cBtn.setFont(font);  
 } **catch** (FontFormatException | IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 */\*\*  
 \* Инициализация и установка для элементов окна слушателя события (клика)  
 \*/* **private void** initListeners() {  
 switcher = **new** Switcher(**this**);  
 ioBtn.addActionListener(switcher);  
 aBtn.addActionListener(switcher);  
 cBtn.addActionListener(switcher);  
 }  
  
 */\*\*  
 \* Добавление компонентов на форму  
 \*/* **private void** addComponentsToFrame() {  
 JPanel lblPanel = **new** JPanel(**new** FlowLayout());  
 lblPanel.add(currentStateLbl);  
  
 JPanel btnPanel = **new** JPanel(**new** FlowLayout());  
 btnPanel.add(ioBtn);  
 btnPanel.add(aBtn);  
 btnPanel.add(cBtn);  
  
 add(lblPanel, BorderLayout.NORTH);  
 add(btnPanel, BorderLayout.CENTER);  
 }  
  
 */\*\*  
 \* Возвращает текущее состояние конечного автомата  
 \*/* **public** String getCurrentState() {  
 **return** currentState;  
 }  
  
 */\*\*  
 \* Устанавливает текущее состояние конечного автомата  
 \*/* **public void** setCurrentState(String state) {  
 **this**.currentState = state;  
 currentStateLbl.setText(**"Текущее состояние: "** + state);  
 }  
}

## Switcher.java

**package** ru.eninja.flang;  
  
**import** javax.swing.\*;  
**import** java.awt.event.ActionEvent;  
**import** java.awt.event.ActionListener;  
  
  
*/\*\*  
 \* Класс конечного автомата  
 \*/***public class** Switcher **implements** ActionListener {  
  
 */\* Форма \*/* **private** Frame **frame**;  
  
 */\* Состояния \*/* **private static final** String ***OFF*** = **"Выкл"**;  
 **private static final** String ***INC*** = **"Вкл"**;  
 **private static final** String ***A*** = **"A"**;  
 **private static final** String ***C*** = **"C"**;  
  
 */\* Переходы \*/* **private static final** String ***IO*** = **"I/O"**;  
 *// A = "A";  
 // C = "C";* **public** Switcher(Frame frame) {  
 **this**.**frame** = frame;  
 frame.setCurrentState(***OFF***);  
 }  
  
 */\*\*  
 \* Метод, вызываемый при клике на кнопку  
 \*/* @Override  
 **public void** actionPerformed(ActionEvent e) {  
 **if** (!(e.getSource() **instanceof** JButton)) {  
 **return**;  
 }  
 String btnText = ((JButton) e.getSource()).getText();  
  
 */\* Переходы между состояниями \*/* **switch** (**frame**.getCurrentState()) {  
 **case *OFF***:  
 **switch** (btnText) {  
 **case *IO***:  
 **frame**.setCurrentState(***INC***);  
 **break**;  
 }  
 **break**;  
 **case *INC***:  
 **switch** (btnText) {  
 **case *IO***:  
 **frame**.setCurrentState(***OFF***);  
 **break**;  
 **case *A***:  
 **frame**.setCurrentState(***A***);  
 **break**;  
 **case *C***:  
 **frame**.setCurrentState(***C***);  
 **break**;  
 }  
 **break**;  
 **case *A***:  
 **switch** (btnText) {  
 **case *A***:  
 **frame**.setCurrentState(***INC***);  
 **break**;  
 **case *C***:  
 **frame**.setCurrentState(***C***);  
 **break**;  
 **case *IO***:  
 **frame**.setCurrentState(***OFF***);  
 **break**;  
 }  
 **break**;  
 **case *C***:  
 **switch** (btnText) {  
 **case *IO***:  
 **frame**.setCurrentState(***OFF***);  
 **break**;  
 **case *C***:  
 **frame**.setCurrentState(***INC***);  
 **break**;  
 **case *A***:  
 **frame**.setCurrentState(***A***);  
 **break**;  
 }  
 **break**;  
 }  
 }  
}