

Санкт-Петербургский государственный институт точной механики и оптики (технический университет)

Кафедра «Компьютерные технологии»

В.А. Лазута

## **Графический пользовательский интерфейс для проекта “Immerssion”**

Объектно-ориентированное программирование с  
явным выделением состояний

Проектная документация

Проект создан в рамках  
“Движения за открытую проектную документацию”  
<http://is.ifmo.ru>

Санкт-Петербург  
2003

## Содержание

<b>ВВЕДЕНИЕ</b> .....	<b>3</b>
<b>1. ПОСТАНОВКА ЗАДАЧИ</b> .....	<b>3</b>
<b>2. ОПИСАНИЕ</b> .....	<b>4</b>
<b>3. ДИАГРАММА КЛАССОВ</b> .....	<b>6</b>
<b>4. КЛАСС «MenuVis»</b> .....	<b>7</b>
4.1. Словесное описание.....	7
4.2. Схема связей класса.....	8
4.3. Краткое описание методов.....	8
<b>5. КЛАСС «TlMenu»</b> .....	<b>8</b>
4.1. Словесное описание.....	8
4.2. Схема связей класса.....	9
4.3. Краткое описание методов.....	9
4.4. Автомат «Управление структурой меню».....	9
4.4.1. Словесное описание.....	9
4.4.2. Схема связей автомата.....	9
4.4.3. Граф переходов автомата.....	10
<b>6. КЛАСС «MenuDiagram»</b> .....	<b>11</b>
6.1. Словесное описание.....	11
6.2. Схема связей класса.....	11
6.3. Краткое описание методов.....	11
<b>7. КЛАСС «MenuDiagramItem»</b> .....	<b>11</b>
7.1. Словесное описание.....	11
7.2. Схема связей класса.....	12
7.3. Краткое описание методов.....	12
<b>8. ИСХОДНЫЕ ТЕКСТЫ ПРОГРАММЫ</b> .....	<b>13</b>
8.1. Файл «MenuVis.java».....	13
8.2. Файл «TlMenu.java».....	20
8.3. Файл «MenuDiagram.java».....	24
<b>9. ИСТОЧНИКИ</b> .....	<b>25</b>

# Введение

Для алгоритмизации и программирования задач логического управления А.А.Шалыто была предложена SWITCH-технология, которая применительно к событийным и объектно-ориентированным программам была разработана совместно с Н.И.Туккелем. Подробно ознакомиться с этой технологией и с конкретными примерами ее использования можно на сайтах <http://is.ifmo.ru> и <http://www.softcraft.ru>.

Эту технологию удобно использовать для построения системы управления структурой вложенных меню, так как при применении графов переходов структура меню может быть изоморфно отражена в исходном коде программы, и в нее достаточно легко вносить изменения (добавлять или удалять пункты меню).

## 1. Постановка задачи

Графический пользовательский интерфейс (выделен штриховкой на рис.1), описанный ниже, используется в проекте сетевой 3D-игры “Immerssion”. Он предназначен для изменения настроек игры и игрока, выбора протокола сетевой игры, сервера и т. д.

Графический интерфейс представляет собой вложенное меню, в котором каждый пункт является панелью управления, отображающей графические компоненты (кнопки, поля ввода текста и т.д.). Структура графических компонентов, их размещение и взаимодействие друг с другом и системой ввода построены по аналогии с библиотекой *VCL (Visual Components Library* – библиотека визуальных компонентов, которая используется в *Delphy* и *C Builder*). Навигация по меню (переключения между панелями и управления их отображением) выполнена на основе автоматного подхода.

Для представления работы меню написан *Java*-апплет, управление в котором осуществляет автомат.

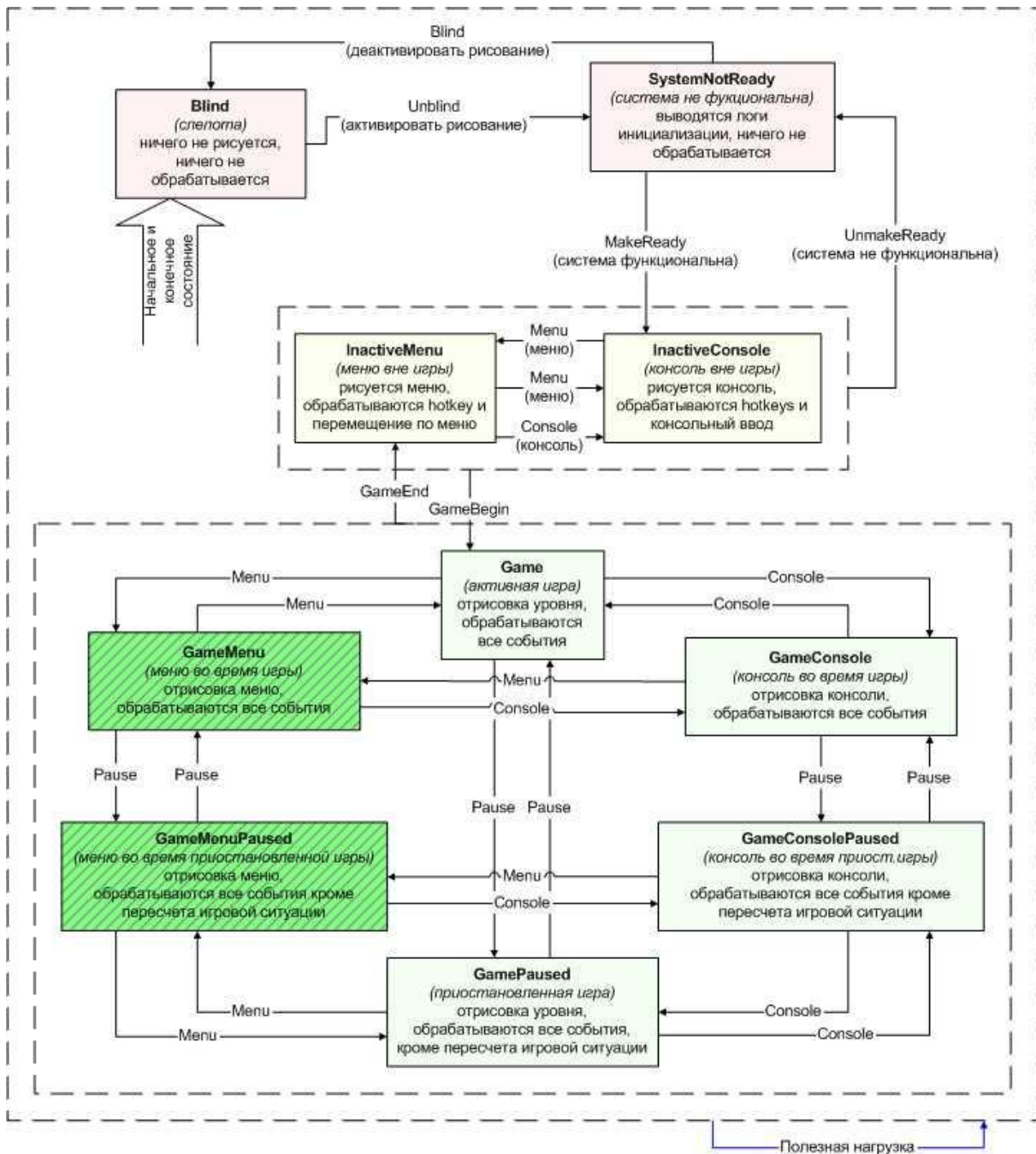


Рис. 1. Схема проекта “Immersion”

## 2. Описание метода

На рис. 2 справа приведено меню, реализованное в данной программе. Пункт “Inactive menu” не является пунктом меню (панелью с графическими компонентами). Это “входное/выходное” состояние меню, в котором оно само неактивно. В нем управление передается системе.

Java-апплет, визуализирующий работу меню, отображает переходы между панелями на схеме меню (правая часть на рис.2), возникающие при нажатии соответствующих кнопок на текущей панели управления (левая часть на рис.2).

На рис. 2, 3 для примера приведено меню в состояниях “Setup engine” и “Multi-player”. Эти состояния на указанных рисунках выделены зеленым цветом.

Из состояния “Setup engine” (рис. 2) можно перейти в следующие состояния:

- “Video” (переходу соответствует кнопка “Video”, порождающая соответствующее событие);
- “Sound” (переходу соответствует кнопка “Sound”);
- “Input” (переходу соответствует кнопка “Input”);
- “Main menu” (переходу соответствует кнопка “Back\_07”. Номер, использованный в названии кнопки, совпадает с номером состояния “Setup engine”, из которого выполняются переходы);
- “Inactive menu” (переходу соответствует кнопка “Exit”).

Аналогично строятся переходы из состояния “Multi-player” (рис.3) в указанные ниже состояния:

- “Multi-driver” (переходу соответствует кнопка “Multi-driver”);
- “Main menu” (переходу соответствует кнопка “Back\_03”);
- “Inactive menu” (переходу соответствует кнопка “Exit”).

Также ведется логирование работы автомата, управляющего структурой меню (нижняя часть рис. 2).

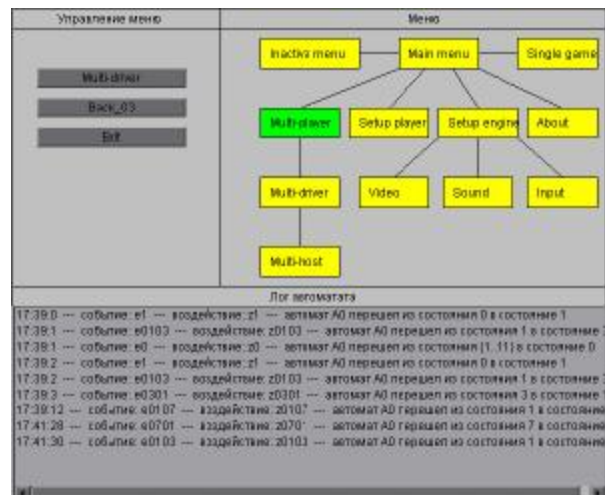
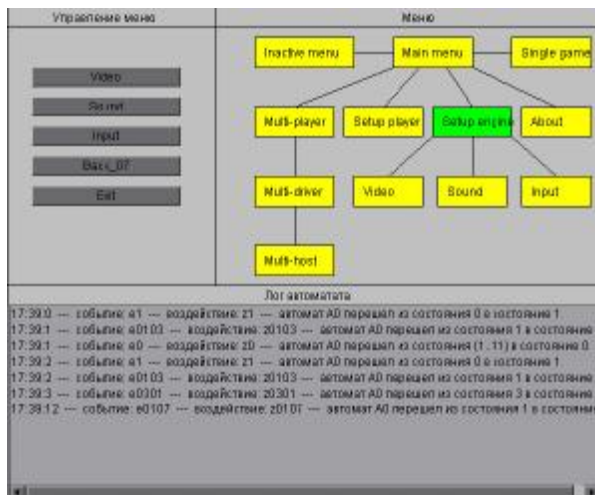


Рис.2. Визуализация работы меню (часть 1)      Рис.3. Визуализация работы меню (часть 2)

### 3. Диаграмма классов

Диаграмма классов приведена на рис.4. В этой диаграмме класс:

- MenuVis - создает экземпляры классов TlMenu и MenuDiagram и размещает их на панели апплета (разд. 4);
- TlMenu – реализует автомат управления структурой меню и его действия, осуществляет логирование работы автомата (разд. 5);
- MenuDiagram – создает массив экземпляров классов MenuItem (разд. 6);
- MenuItem – реализует пункт меню на схеме меню (разд. 7).

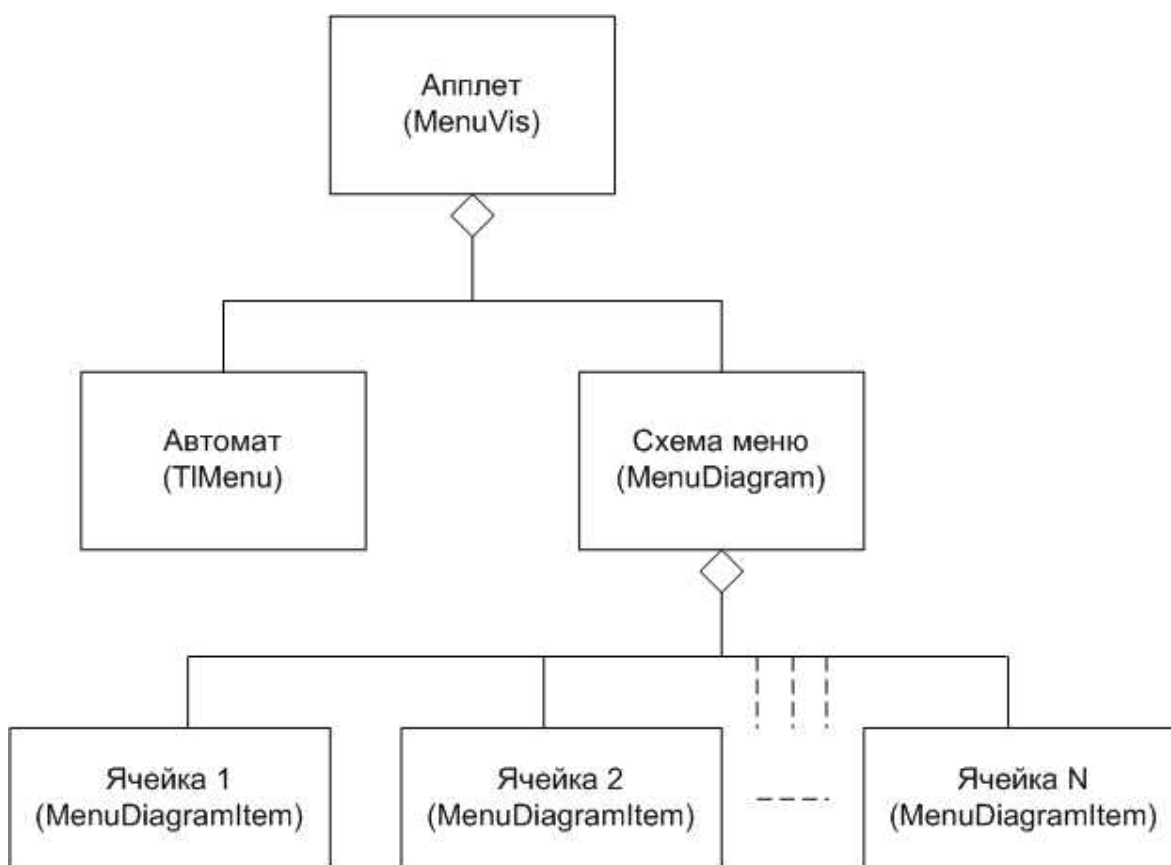


Рис. 4. Диаграмма классов

## 4. Класс “MenuVis”

### 4.1. Словесное описание

Создает экземпляры классов T1Menu и MenuDiagram и размещает их на панели апплета.

### 4.2. Структурная схема класса

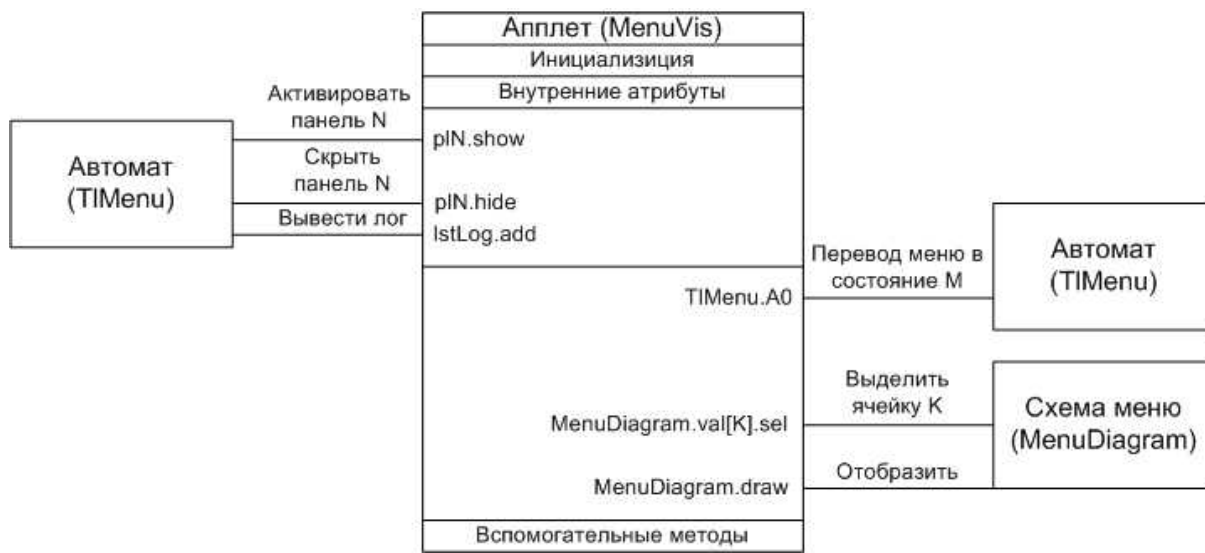


Рис. 5. Структурная схема класса “MenuVis”

### 4.3. Краткое описание методов

- `public void init()` – инициализация апплета и графических компонентов;
- `public void paint(Graphics g)` – отображение графических компонентов, структуры меню и окна логирования работы автомата;
- `public void actionPerformed(ActionEvent e)` – реакция на нажатие кнопок.

## 5. Класс “TlMenu”

### 5.1. Словесное описание

Реализует автомат управления структурой меню и его действия, осуществляет логирование работы автомата.

### 5.2. Структурная схема класса

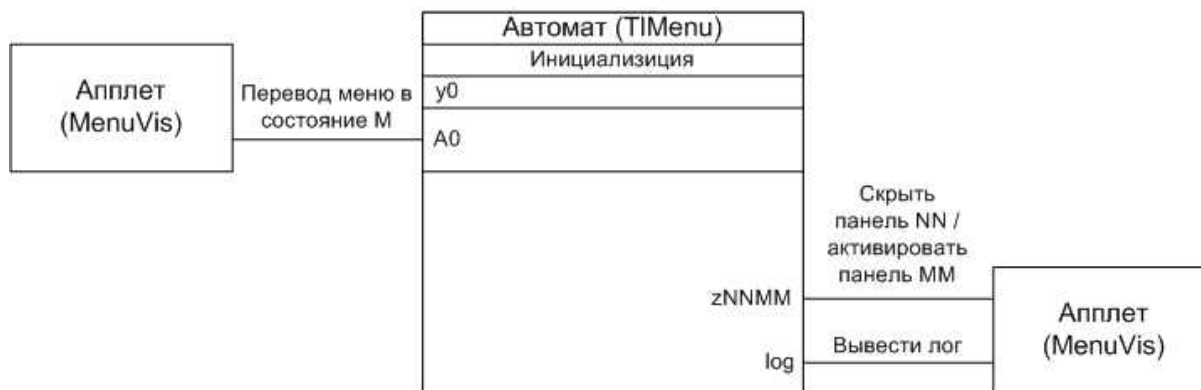


Рис. 6. Структурная схема класса “TlMenu”

### 5.3. Краткое описание методов

- `TlMenu(MenuVis MV)` – конструктор класса `TlMenu`;
- `public void A0(int e)` – автоматный метод;
- `private void zNNMM()` – методы выходных воздействий автомата, где `NN` -- номер состояния, из которого осуществляется переход, а `MM` -- номер состояния, в которое возможен переход;
- `private String log(String e, String z, String a0, String a1)` – генерация строки логирования действия автомата.

### 5.4. Автомат “Управление структурой меню”

#### 5.4.1. Словесное описание

Автомат осуществляет управление меню, при этом он реализует переходы между пунктами меню - активирует/скрывает соответствующие панели.

Автомат реализован как открытый метод `A0(int e)`. Его состояние хранится в защищенной переменной `y0`. Выходные воздействия автомата реализованы в виде защищенных методов `zNNMM()`.

Схема связей автомата изображена на рис.6, а его граф переходов – на рис.7.



## 5.4.2. Схема связей автомата

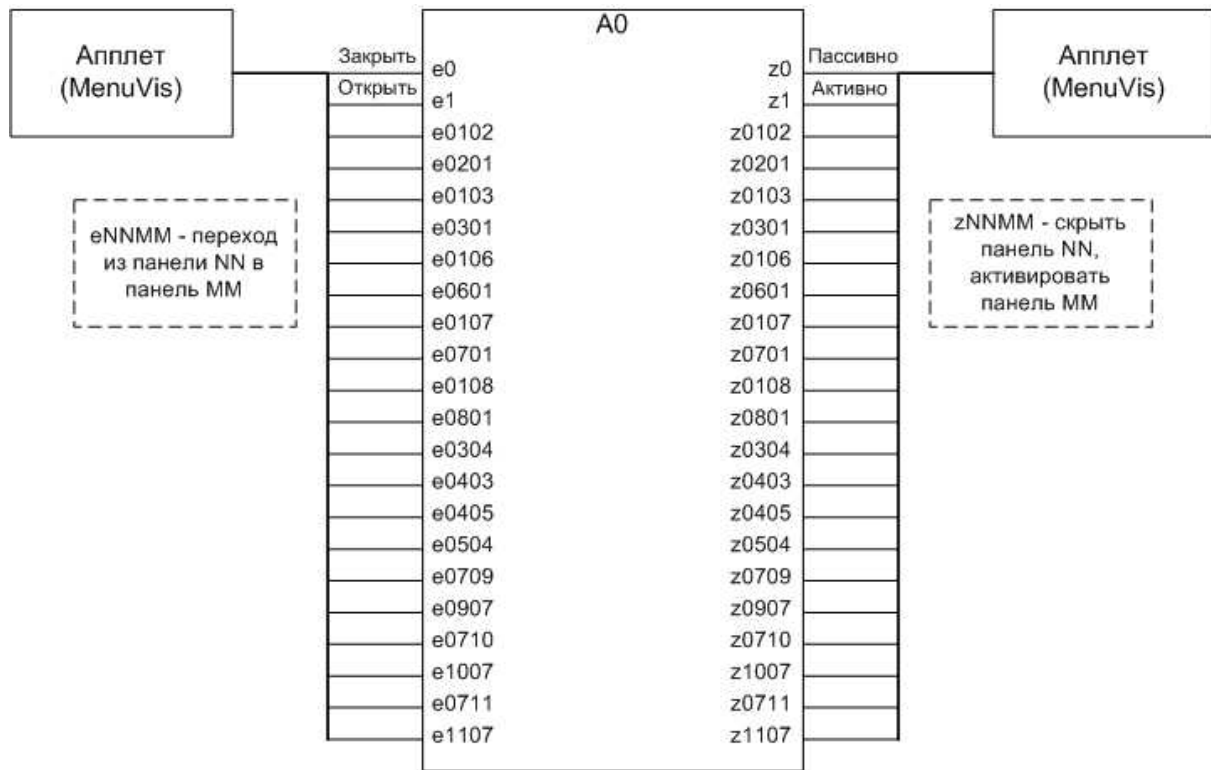


Рис. 7. Схема связей автомата A0

Событие вида eNNMM означает переход в меню с панели (из состояния) с номером NN на панель (в состояние) с номером MM.

Выходное воздействие автомата вида zNNMM скрывает панель с номером NN и активирует панель с номером MM.

### 5.4.3. Граф переходов автомата

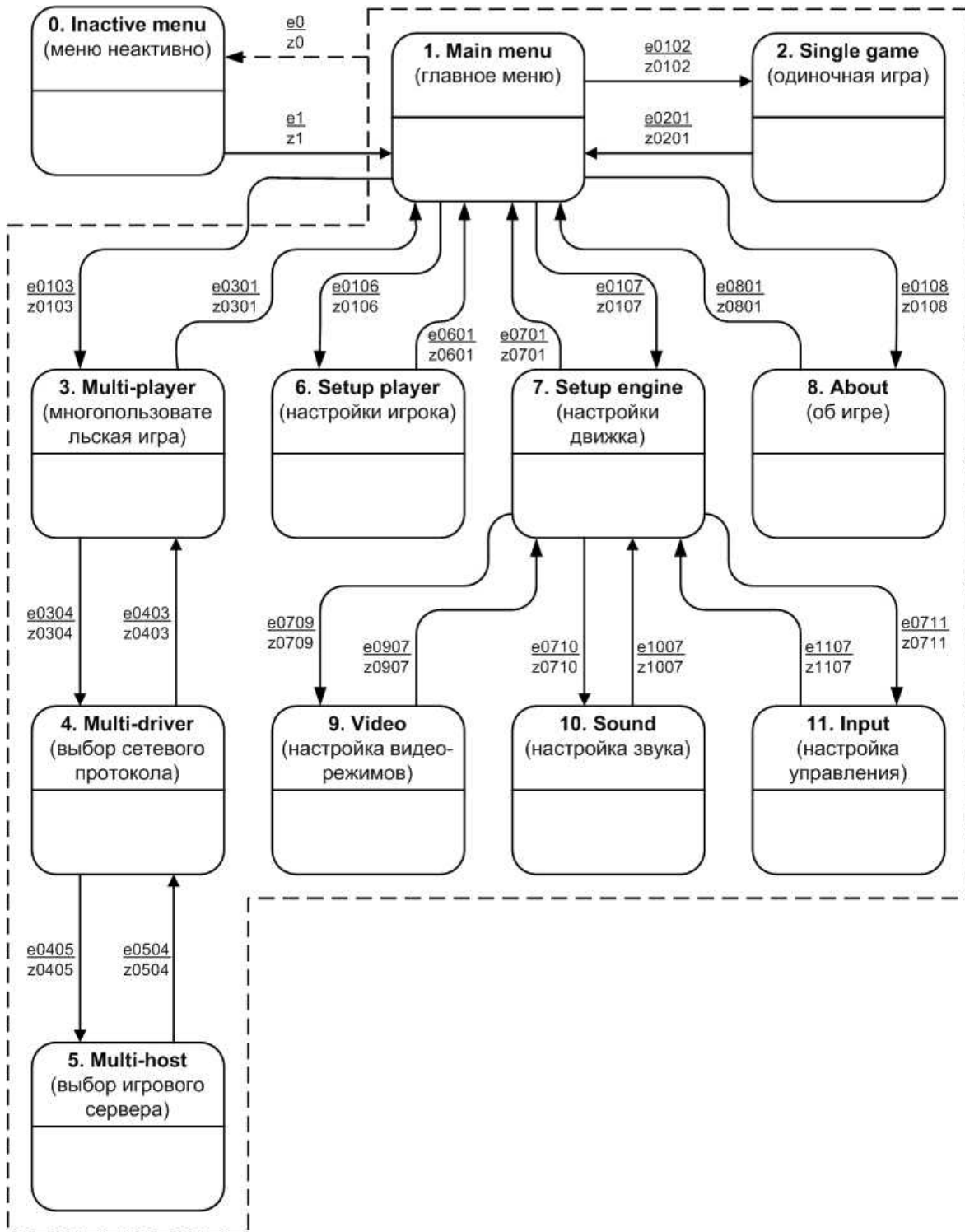


Рис. 8. Граф переходов автомата А0

## 6. Класс “MenuDiagram”

### 6.1. Словесное описание

Создает массив экземпляров классов MenuItem, отвечает за их отображение.

### 6.2. Структурная схема класса

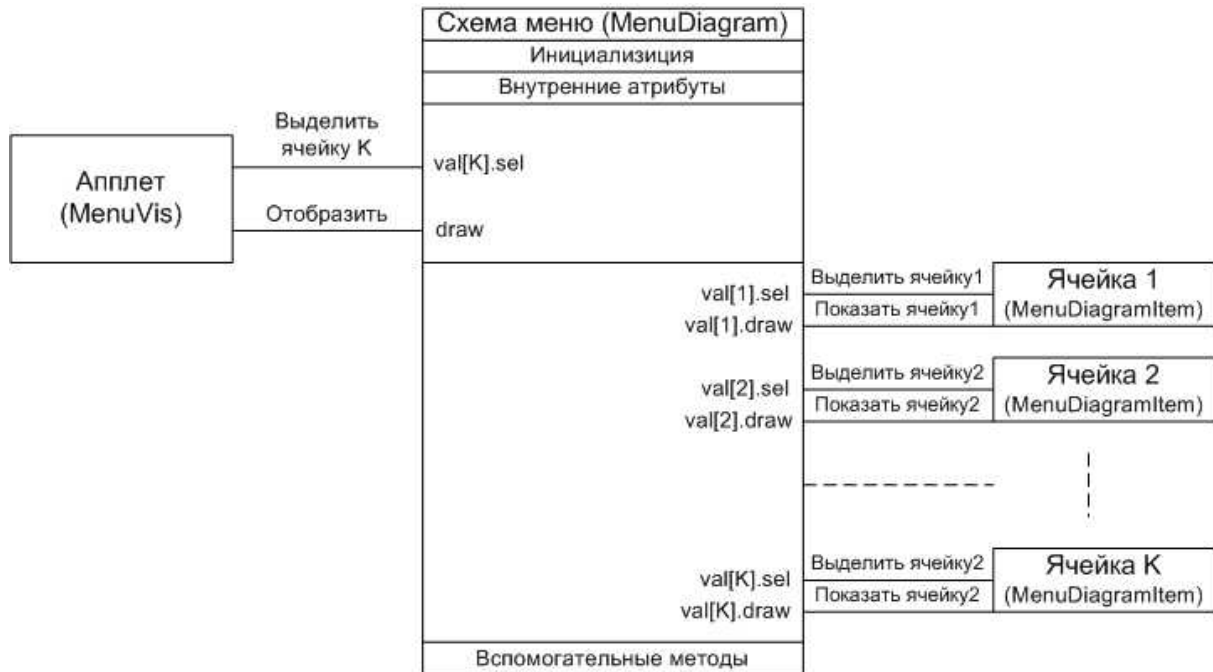


Рис. 9. Структурная схема класса “MenuDiagram”

### 6.3. Краткое описание методов

- public void draw(Graphics g) – отображает схему меню;
- public void unselectAll() – делает невыделенными все пункты меню;
- MenuDiagram(int num) – конструктор класса MenuDiagram.

## 7. Класс “MenuItem”

### 7.1. Словесное описание

Реализует пункт меню на схеме меню.

## 7.2. Структурная схема класса



Рис. 10. Структурная схема класса “MenuItem”

## 7.3. Краткое описание методов

Эти методы имеют следующие названия:

- `public void draw()` – отображает пункт меню;
- `MenuItem(int x, int y, int dx, int dy, String val)` – конструктор класса MenuItem.

## 8. Исходные тексты программы

### 8.1. Файл MenuVis.java

```
//Описание и реализация класса MenuVis
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

//Класс MenuVis
public class MenuVis extends Applet implements ActionListener
{
    //Графические компоненты

        //Кнопки переходов между панелями
        Button btn0p11;
        Button btn1p10;
        Button btn1p12;
        Button btn1p13;
        Button btn1p16;
        Button btn1p17;
        Button btn1p18;
        Button btn2p10;
        Button btn2p11;
        Button btn3p10;
        Button btn3p11;
        Button btn3p14;
        Button btn4p10;
        Button btn4p13;
        Button btn4p15;
        Button btn5p10;
        Button btn5p14;
        Button btn6p10;
        Button btn6p11;
        Button btn7p10;
        Button btn7p11;
        Button btn7p19;
        Button btn7p110;
        Button btn7p111;
        Button btn8p10;
        Button btn8p11;
        Button btn9p10;
        Button btn9p17;
        Button btn10p10;
        Button btn10p17;
        Button btn11p10;
        Button btn11p17;

        //Панели (пункты меню), на которых размещены графические компоненты
        Panel p10;
        Panel p11;
        Panel p12;
        Panel p13;
        Panel p14;
        Panel p15;
        Panel p16;
        Panel p17;
        Panel p18;
        Panel p19;
        Panel p110;
        Panel p111;

        //Экземпляр класса TlMenu
        TlMenu tMenu;
        //Экземпляр класса MenuDiagram
        MenuDiagram mVis;List lstLog;

        Image imgMenu;
        Graphics g;
        Graphics graphMenu;

//Инициализация графических компонентов, панелей и классов MenuDiagram и TlMenu
    public void init()
    {
        tMenu=new TlMenu(this);
```

```

mVis=new MenuDiagram(12);
mVis.val[0]=new MenuDiagramItem(250,30,100,30,"Inactive menu");
mVis.val[1]=new MenuDiagramItem(390,30,80,30,"Main menu");
mVis.val[2]=new MenuDiagramItem(510,30,80,30,"Single game");
mVis.val[3]=new MenuDiagramItem(250,100,80,30,"Multi-player");
mVis.val[4]=new MenuDiagramItem(250,170,80,30,"Multi-driver");
mVis.val[5]=new MenuDiagramItem(250,240,80,30,"Multi-host");
mVis.val[6]=new MenuDiagramItem(340,100,80,30,"Setup player");
mVis.val[7]=new MenuDiagramItem(430,100,80,30,"Setup engine");
mVis.val[8]=new MenuDiagramItem(520,100,70,30,"About");
mVis.val[9]=new MenuDiagramItem(350,170,70,30,"Video");
mVis.val[10]=new MenuDiagramItem(435,170,70,30,"Sound");
mVis.val[11]=new MenuDiagramItem(520,170,70,30,"Input");
mVis.val[0].sel=true;

setLayout(null);

pl0=new Panel();
pl0.setLocation(10,50);
pl0.setSize(200,200);
pl0.setLayout(null);

btn0p11=new Button("Activate menu");
btn0p11.addActionListener(this);
btn0p11.setLocation(15,10);
btn0p11.setSize(150,20);
pl0.add(btn0p11);
add(pl0);

pl1=new Panel();
pl1.setLocation(10,50);
pl1.setSize(180,200);
pl1.setLayout(null);

btn1p10=new Button("Exit");
btn1p10.addActionListener(this);
btn1p10.setLocation(15,160);
btn1p10.setSize(150,20);
pl1.add(btn1p10);
btn1p12=new Button("Single game");
btn1p12.addActionListener(this);
btn1p12.setLocation(15,10);
btn1p12.setSize(150,20);
pl1.add(btn1p12);
btn1p13=new Button("Multi-player");
btn1p13.addActionListener(this);
btn1p13.setLocation(15,40);
btn1p13.setSize(150,20);
pl1.add(btn1p13);
btn1p16=new Button("Setup player");
btn1p16.addActionListener(this);
btn1p16.setLocation(15,70);
btn1p16.setSize(150,20);
pl1.add(btn1p16);
btn1p17=new Button("Setup engine");
btn1p17.addActionListener(this);
btn1p17.setLocation(15,100);
btn1p17.setSize(150,20);
pl1.add(btn1p17);
btn1p18=new Button("About");
btn1p18.addActionListener(this);
btn1p18.setLocation(15,130);
btn1p18.setSize(150,20);
pl1.add(btn1p18);
add(pl1);
pl1.hide();

pl2=new Panel();
pl2.setLocation(10,50);
pl2.setSize(180,200);
pl2.setLayout(null);

btn2p10=new Button("Exit");
btn2p10.addActionListener(this);
btn2p10.setLocation(15,40);
btn2p10.setSize(150,20);
pl2.add(btn2p10);
add(pl2);

```

```

btn2pl1=new Button("Back_02");
btn2pl1.addActionListener(this);
btn2pl1.setLocation(15,10);
btn2pl1.setSize(150,20);
pl2.add(btn2pl1);
add(pl2);
pl2.hide();

pl3=new Panel();
pl3.setLocation(10,50);
pl3.setSize(180,200);
pl3.setLayout(null);

btn3pl0=new Button("Exit");
btn3pl0.addActionListener(this);
btn3pl0.setLocation(15,70);
btn3pl0.setSize(150,20);
pl3.add(btn3pl0);
btn3pl4=new Button("Multi-driver");
btn3pl4.addActionListener(this);
btn3pl4.setLocation(15,10);
btn3pl4.setSize(150,20);
pl3.add(btn3pl4);
btn3pl1=new Button("Back_03");
btn3pl1.addActionListener(this);
btn3pl1.setLocation(15,40);
btn3pl1.setSize(150,20);
pl3.add(btn3pl1);
add(pl3);
pl3.hide();

pl4=new Panel();
pl4.setLocation(10,50);
pl4.setSize(180,200);
pl4.setLayout(null);

btn4pl0=new Button("Exit");
btn4pl0.addActionListener(this);
btn4pl0.setLocation(15,70);
btn4pl0.setSize(150,20);
pl4.add(btn4pl0);
btn4pl5=new Button("Multi-host");
btn4pl5.addActionListener(this);
btn4pl5.setLocation(15,10);
btn4pl5.setSize(150,20);
pl4.add(btn4pl5);
btn4pl3=new Button("Back_04");
btn4pl3.addActionListener(this);
btn4pl3.setLocation(15,40);
btn4pl3.setSize(150,20);
pl4.add(btn4pl3);
add(pl4);
pl4.hide();

pl5=new Panel();
pl5.setLocation(10,50);
pl5.setSize(180,200);
pl5.setLayout(null);

btn5pl0=new Button("Exit");
btn5pl0.addActionListener(this);
btn5pl0.setLocation(15,40);
btn5pl0.setSize(150,20);
pl5.add(btn5pl0);
btn5pl4=new Button("Back_05");
btn5pl4.addActionListener(this);
btn5pl4.setLocation(15,10);
btn5pl4.setSize(150,20);
pl5.add(btn5pl4);
add(pl5);
pl5.hide();

pl6=new Panel();
pl6.setLocation(10,50);
pl6.setSize(180,200);
pl6.setLayout(null);

```

```

btn6p10=new Button("Exit");
btn6p10.addActionListener(this);
btn6p10.setLocation(15,40);
btn6p10.setSize(150,20);
p16.add(btn6p10);
btn6p11=new Button("Back_06");
btn6p11.addActionListener(this);
btn6p11.setLocation(15,10);
btn6p11.setSize(150,20);
p16.add(btn6p11);
add(p16);
p16.hide();

p17=new Panel();
p17.setLocation(10,50);
p17.setSize(180,200);
p17.setLayout(null);

btn7p10=new Button("Exit");
btn7p10.addActionListener(this);
btn7p10.setLocation(15,130);
btn7p10.setSize(150,20);
p17.add(btn7p10);
btn7p19=new Button("Video");
btn7p19.addActionListener(this);
btn7p19.setLocation(15,10);
btn7p19.setSize(150,20);
p17.add(btn7p19);
btn7p110=new Button("Sound");
btn7p110.addActionListener(this);
btn7p110.setLocation(15,40);
btn7p110.setSize(150,20);
p17.add(btn7p110);
btn7p111=new Button("Input");
btn7p111.addActionListener(this);
btn7p111.setLocation(15,70);
btn7p111.setSize(150,20);
p17.add(btn7p111);
btn7p11=new Button("Back_07");
btn7p11.addActionListener(this);
btn7p11.setLocation(15,100);
btn7p11.setSize(150,20);
p17.add(btn7p11);
add(p17);
p17.hide();

p18=new Panel();
p18.setLocation(10,50);
p18.setSize(180,200);
p18.setLayout(null);

btn8p10=new Button("Exit");
btn8p10.addActionListener(this);
btn8p10.setLocation(15,40);
btn8p10.setSize(150,20);
p18.add(btn8p10);
btn8p11=new Button("Back_08");
btn8p11.addActionListener(this);
btn8p11.setLocation(15,10);
btn8p11.setSize(150,20);
p18.add(btn8p11);
add(p18);
p18.hide();

p19=new Panel();
p19.setLocation(10,50);
p19.setSize(180,200);
p19.setLayout(null);

btn9p10=new Button("Exit");
btn9p10.addActionListener(this);
btn9p10.setLocation(15,40);
btn9p10.setSize(150,20);
p19.add(btn9p10);
btn9p17=new Button("Back_09");
btn9p17.addActionListener(this);

```



```

btn9pl7.setLocation(15,10);
btn9pl7.setSize(150,20);
pl9.add(btn9pl7);
add(pl9);
pl9.hide();

pl10=new Panel();
pl10.setLocation(10,50);
pl10.setSize(180,200);
pl10.setLayout(null);

btn10pl0=new Button("Exit");
btn10pl0.addActionListener(this);
btn10pl0.setLocation(15,40);
btn10pl0.setSize(150,20);
pl10.add(btn10pl0);
btn10pl7=new Button("Back_10");
btn10pl7.addActionListener(this);
btn10pl7.setLocation(15,10);
btn10pl7.setSize(150,20);
pl10.add(btn10pl7);
add(pl10);
pl10.hide();

pl11=new Panel();
pl11.setLocation(10,50);
pl11.setSize(180,200);
pl11.setLayout(null);

btn11pl0=new Button("Exit");
btn11pl0.addActionListener(this);
btn11pl0.setLocation(15,40);
btn11pl0.setSize(150,20);
pl11.add(btn11pl0);
btn11pl7=new Button("Back_11");
btn11pl7.addActionListener(this);
btn11pl7.setLocation(15,10);
btn11pl7.setSize(150,20);
pl11.add(btn11pl7);
add(pl11);
pl11.hide();

lstLog=new List();
lstLog.setBounds(1,300,600,200);
add(lstLog);
g=getGraphics();
imgMenu=createImage(600, 300);
graphMenu=imgMenu.getGraphics();
}

```

//Отображение апплета

```

public void paint(Graphics g)
{
    mVis.draw(graphMenu);
    graphMenu.setColor(Color.black);
    graphMenu.drawLine(350,45,390,45);
    graphMenu.drawLine(470,45,510,45);
    graphMenu.drawLine(390,60,290,100);
    graphMenu.drawLine(470,60,555,100);
    graphMenu.drawLine(417,60,380,100);
    graphMenu.drawLine(444,60,470,100);
    graphMenu.drawLine(290,130,290,170);
    graphMenu.drawLine(290,200,290,240);
    graphMenu.drawLine(430,130,385,170);
    graphMenu.drawLine(470,130,470,170);
    graphMenu.drawLine(510,130,555,170);
    graphMenu.drawRect(0,0,210,280);
    graphMenu.drawRect(210,0,389,280);
    graphMenu.drawRect(0,20,599,500);
    graphMenu.drawString("Меню", 85, 15);
    graphMenu.drawString("Структура меню", 365, 15);
    graphMenu.drawString("Лог автомата", 260, 295);

    g.drawImage(imgMenu, 0, 0, null);
}

```

//Обработчики событий

```
public void actionPerformed(ActionEvent e)
{
    if ( e.getActionCommand().equals(btn0p11.getLabel()))
    {
        tMenu.A0(1);
        mVis.unselAll();
        mVis.val[1].sel=true;
    }
    if ( e.getActionCommand().equals(btn1p10.getLabel()))
    {
        tMenu.A0(0);
        mVis.unselAll();
        mVis.val[0].sel=true;
    }
    if ( e.getActionCommand().equals(btn1p12.getLabel()))
    {
        tMenu.A0(102);
        mVis.unselAll();
        mVis.val[2].sel=true;
    }
    if ( e.getActionCommand().equals(btn1p13.getLabel()))
    {
        tMenu.A0(103);
        mVis.unselAll();
        mVis.val[3].sel=true;
    }
    if ( e.getActionCommand().equals(btn1p16.getLabel()))
    {
        tMenu.A0(106);
        mVis.unselAll();
        mVis.val[6].sel=true;
    }
    if ( e.getActionCommand().equals(btn1p17.getLabel()))
    {
        tMenu.A0(107);
        mVis.unselAll();
        mVis.val[7].sel=true;
    }
    if ( e.getActionCommand().equals(btn1p18.getLabel()))
    {
        tMenu.A0(108);
        mVis.unselAll();
        mVis.val[8].sel=true;
    }
    if ( e.getActionCommand().equals(btn2p10.getLabel()))
    {
        tMenu.A0(0);
        mVis.unselAll();
        mVis.val[0].sel=true;
    }
    if ( e.getActionCommand().equals(btn2p11.getLabel()))
    {
        tMenu.A0(201);
        mVis.unselAll();
        mVis.val[1].sel=true;
    }
    if ( e.getActionCommand().equals(btn3p10.getLabel()))
    {
        tMenu.A0(0);
        mVis.unselAll();
        mVis.val[0].sel=true;
    }
    if ( e.getActionCommand().equals(btn3p11.getLabel()))
    {
        tMenu.A0(301);
        mVis.unselAll();
        mVis.val[1].sel=true;
    }
    if ( e.getActionCommand().equals(btn3p14.getLabel()))
    {
        tMenu.A0(304);
        mVis.unselAll();
        mVis.val[4].sel=true;
    }
    if ( e.getActionCommand().equals(btn4p10.getLabel()))
    {
        tMenu.A0(0);
        mVis.unselAll();
        mVis.val[0].sel=true;
    }
}
```

```

    }
    if ( e.getActionCommand().equals(btn4p13.getLabel()))
    {
        tMenu.A0(403);
        mVis.unselAll();
        mVis.val[3].sel=true;
    }
    if ( e.getActionCommand().equals(btn4p15.getLabel()))
    {
        tMenu.A0(405);
        mVis.unselAll();
        mVis.val[5].sel=true;
    }
    if ( e.getActionCommand().equals(btn5p10.getLabel()))
    {
        tMenu.A0(0);
        mVis.unselAll();
        mVis.val[0].sel=true;
    }
    if ( e.getActionCommand().equals(btn5p14.getLabel()))
    {
        tMenu.A0(504);
        mVis.unselAll();
        mVis.val[4].sel=true;
    }
    if ( e.getActionCommand().equals(btn6p10.getLabel()))
    {
        tMenu.A0(0);
        mVis.unselAll();
        mVis.val[0].sel=true;
    }
    if ( e.getActionCommand().equals(btn6p11.getLabel()))
    {
        tMenu.A0(601);
        mVis.unselAll();
        mVis.val[1].sel=true;
    }
    if ( e.getActionCommand().equals(btn7p10.getLabel()))
    {
        tMenu.A0(0);
        mVis.unselAll();
        mVis.val[0].sel=true;
    }
    if ( e.getActionCommand().equals(btn7p11.getLabel()))
    {
        tMenu.A0(701);
        mVis.unselAll();
        mVis.val[1].sel=true;
    }
    if ( e.getActionCommand().equals(btn7p19.getLabel()))
    {
        tMenu.A0(709);
        mVis.unselAll();
        mVis.val[9].sel=true;
    }
    if ( e.getActionCommand().equals(btn7p110.getLabel()))
    {
        tMenu.A0(710);
        mVis.unselAll();
        mVis.val[10].sel=true;
    }
    if ( e.getActionCommand().equals(btn7p111.getLabel()))
    {
        tMenu.A0(711);
        mVis.unselAll();
        mVis.val[11].sel=true;
    }
    if ( e.getActionCommand().equals(btn8p10.getLabel()))
    {
        tMenu.A0(0);
        mVis.unselAll();
        mVis.val[0].sel=true;
    }
    if ( e.getActionCommand().equals(btn8p11.getLabel()))
    {
        tMenu.A0(801);
        mVis.unselAll();
        mVis.val[1].sel=true;
    }
    if ( e.getActionCommand().equals(btn9p10.getLabel()))

```

```

        {
            tMenu.A0(0);
            mVis.unselAll();
            mVis.val[0].sel=true;
        }
    if ( e.getActionCommand().equals(btn9pl7.getLabel()))
        {
            tMenu.A0(907);
            mVis.unselAll();
            mVis.val[7].sel=true;
        }
    if ( e.getActionCommand().equals(btn10pl0.getLabel()))
        {
            tMenu.A0(0);
            mVis.unselAll();
            mVis.val[0].sel=true;
        }
    if ( e.getActionCommand().equals(btn10pl7.getLabel()))
        {
            tMenu.A0(1007);
            mVis.unselAll();
            mVis.val[7].sel=true;
        }
    if ( e.getActionCommand().equals(btn11pl0.getLabel()))
        {
            tMenu.A0(0);
            mVis.unselAll();
            mVis.val[0].sel=true;
        }
    if ( e.getActionCommand().equals(btn11pl7.getLabel()))
        {
            tMenu.A0(1107);
            mVis.unselAll();
            mVis.val[7].sel=true;
        }
        mVis.draw(graphMenu);
        g.drawImage(imgMenu, 0, 0, null);
    }
}

```

## 8.1. Файл TlMenu.java

```

//Реализация автоматного класса
import java.util.*;

//Инициализация
public class TlMenu
{
    MenuVis MV;

    TlMenu(MenuVis MV)
    {
        this.MV=MV;
        y0=0;
    }

//Генерация строки логирования действия автомата
    private String log(String e, String z, String a0, String a1)
    {
        Date logTime=new Date();
        return(" "+logTime.getHours()+":"+logTime.getMinutes()+":"+logTime.getSeconds()+
            " --- событие: e"+e+" --- воздействие: z"+z+" --- автомат A0 перешел из
состояния "
            +a0+" в состояние "+a1);
    }

//Состояние автомата
    private int y0;

```

```

//Функция автомата
void A0(int e)
{
    switch (y0)
    {
    case 0:
        if (e==1) { z1(); y0=1; }
        break;
    case 1:
        if (e==0) { z0(); y0=0; }
        else if (e==0102) { z0102(); y0=2; }
        else if (e==0103) { z0103(); y0=3; }
        else if (e==0106) { z0106(); y0=6; }
        else if (e==0107) { z0107(); y0=7; }
        else if (e==0108) { z0108(); y0=8; }
        break;
    case 2:
        if (e==0) { z0(); y0=0; }
        else if (e==0201) { z0201(); y0=1; }
        break;
    case 3:
        if (e==0) { z0(); y0=0; }
        else if (e==0301) { z0301(); y0=1; }
        else if (e==0304) { z0304(); y0=4; }
        break;
    case 4:
        if (e==0) { z0(); y0=0; }
        else if (e==0403) { z0403(); y0=3; }
        else if (e==0405) { z0405(); y0=5; }
        break;
    case 5:
        if (e==0) { z0(); y0=0; }
        else if (e==0504) { z0504(); y0=4; }
        break;
    case 6:
        if (e==0) { z0(); y0=0; }
        else if (e==0601) { z0601(); y0=1; }
        break;
    case 7:
        if (e==0) { z0(); y0=0; }
        else if (e==0701) { z0701(); y0=1; }
        else if (e==0709) { z0709(); y0=9; }
        else if (e==0710) { z0710(); y0=10; }
        else if (e==0711) { z0711(); y0=11; }
        break;
    case 8:
        if (e==0) { z0(); y0=0; }
        else if (e==0801) { z0801(); y0=1; }
        break;
    case 9:
        if (e==0) { z0(); y0=0; }
        else if (e==0907) { z0907(); y0=7; }
        break;
    case 10:
        if (e==0) { z0(); y0=0; }
        else if (e==1007) { z1007(); y0=7; }
        break;
    case 11:
        if (e==0) { z0(); y0=0; }
        else if (e==1107) { z1107(); y0=7; }
        break;
    }
}

//Реализация выходных воздействий автомата

//Переводит меню в пассивное состояние(Inactive menu) (скрывает все панели)
private void z0()
{
    MV.pl1.hide();
    MV.pl2.hide();
    MV.pl3.hide();
    MV.pl4.hide();
    MV.pl5.hide();
    MV.pl6.hide();
    MV.pl7.hide();
    MV.pl8.hide();
    MV.pl9.hide();
    MV.pl10.hide();
}

```

```

        MV.pl11.hide();
        MV.pl0.show();
        MV.lstLog.add(log("0","0","{1..11}","0"));
    }

    //Переводит меню в активное состояние(Main меню) (активирует панель Main меню)
    private void z1()
    {
        MV.pl0.hide();
        MV.pl1.show();
        MV.lstLog.add(log("1","1","0","1"));
    }

    //Выходное воздействие вида zNNMM переводят меню из состояния NN в состояние MM
    (скрывает панель с номером NN и активирует панель с номером MM)
    private void z0102()
    {
        MV.pl1.hide();
        MV.pl2.show();
        MV.lstLog.add(log("0102","0102","1","2"));
    }

    private void z0103()
    {
        MV.pl1.hide();
        MV.pl3.show();
        MV.lstLog.add(log("0103","0103","1","3"));
    }

    private void z0106()
    {
        MV.pl1.hide();
        MV.pl6.show();
        MV.lstLog.add(log("0106","0106","1","6"));
    }

    private void z0107()
    {
        MV.pl1.hide();
        MV.pl7.show();
        MV.lstLog.add(log("0107","0107","1","7"));
    }

    private void z0108()
    {
        MV.pl1.hide();
        MV.pl8.show();
        MV.lstLog.add(log("0108","0108","1","8"));
    }

    private void z0201()
    {
        MV.pl2.hide();
        MV.pl1.show();
        MV.lstLog.add(log("0201","0201","2","1"));
    }

    private void z0301()
    {
        MV.pl3.hide();
        MV.pl1.show();
        MV.lstLog.add(log("0301","0301","3","1"));
    }

    private void z0304()
    {
        MV.pl3.hide();
        MV.pl4.show();
        MV.lstLog.add(log("0304","0304","3","4"));
    }

    private void z0403()
    {
        MV.pl4.hide();
        MV.pl3.show();
        MV.lstLog.add(log("0403","0403","4","3"));
    }

    private void z0405()
    {

```

```

        MV.pl4.hide();
        MV.pl5.show();
        MV.lstLog.add(log("0405","0405","4","5"));
    }

    private void z0504()
    {
        MV.pl5.hide();
        MV.pl4.show();
        MV.lstLog.add(log("0504","0504","5","4"));
    }

    private void z0601()
    {
        MV.pl6.hide();
        MV.pl1.show();
        MV.lstLog.add(log("0601","0601","6","1"));
    }

    private void z0701()
    {
        MV.pl7.hide();
        MV.pl1.show();
        MV.lstLog.add(log("0701","0701","7","1"));
    }

    private void z0709()
    {
        MV.pl7.hide();
        MV.pl9.show();
        MV.lstLog.add(log("0709","0709","7","9"));
    }

    private void z0710()
    {
        MV.pl7.hide();
        MV.pl10.show();
        MV.lstLog.add(log("0710","0710","7","10"));
    }

    private void z0711()
    {
        MV.pl7.hide();
        MV.pl11.show();
        MV.lstLog.add(log("0711","0711","7","11"));
    }

    private void z0801()
    {
        MV.pl8.hide();
        MV.pl1.show();
        MV.lstLog.add(log("0801","0801","8","1"));
    }

    private void z0907()
    {
        MV.pl9.hide();
        MV.pl7.show();
        MV.lstLog.add(log("0907","0907","9","7"));
    }

    private void z1007()
    {
        MV.pl10.hide();
        MV.pl7.show();
        MV.lstLog.add(log("1007","1007","10","7"));
    }

    private void z1107()
    {
        MV.pl11.hide();
        MV.pl7.show();
        MV.lstLog.add(log("1107","1107","11","7"));
    }
}

```

## 8.1. Файл MenuDiagram.java

```
//Реализация классов MenuDiagram и MenuItem
import java.awt.*;

//Класс MenuDiagram
public class MenuDiagram
{

//Массив экземпляров класса MenuItem
    MenuItem val[];
    int num;

//Инициализация
    MenuDiagram(int num)
    {
        this.num=num;
        val=new MenuItem[num];
    }

//Отображение
    public void draw(Graphics g)
    {
        for (int i=0; i<num; i++)
            val[i].draw(g);
    }

//Деактивация пунктов меню
    public void unselAll()
    {
        for (int i=0; i<num; i++)
            val[i].sel=false;
    }
}

//Класс MenuItem
class MenuItem
{
    int x,y,dx,dy;
    boolean sel;
    String val;

//Инициализация
    MenuItem(int x, int y, int dx, int dy, String val)
    {
        this.x=x;
        this.y=y;
        this.dx=dx;
        this.dy=dy;
        this.val=val;
        sel=false;
    }

//Отображение
    public void draw(Graphics g)
    {
        if (sel) g.setColor(Color.green);
        else g.setColor(Color.yellow);
        g.fillRect(x,y,dx,dy);
        g.setColor(Color.black);
        g.drawRect(x,y,dx,dy);
        g.drawString(""+val,x+10,y+(int)(dy/2)+5);
    }
};
```



## 9. Источники

1. <http://is.ifmo.ru>
2. <http://www.softcraft.ru>
3. *Шалыто А. А.* SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.