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

Кафедра «Компьютерные технологии»

А.И. Мясников

Моделирование кнопочного телефона с использованием SWITCH-технологии

Проектная документация

Проект создан в рамках
«Движения за открытую проектную документацию»
http://is.ifmo.ru

Санкт-Петербург
2003

Содержание

1. Введение

В рамках курса автоматного проектирования программ было предложено разработать автомат или группу автоматов как инструмент для решения задачи управления кнопочным телефоном. В результате получена модель кнопочного телефона, реализованная с использованием SWITCH-технологии (http://is.ifmo.ru). С помощью этой технологии промоделирована также работа автоматической телефонной станции (АТС).

Автоматный подход весьма удобен для реализации управляющих устройств. При его применении код получается простым и понятным, а вероятность ошибок снижается.

При реализации данного проекта программа заработала почти сразу.

Достоинство подхода состоит, в частности, в том, что проектирование вынесено в отдельный этап, где собственно и находится основная часть создания программы.

2. Постановка задачи

Разработать программу, используя автоматный подход, которая будет эмулировать работу кнопочного телефона. Для этого, в частности, необходимо:

- разработать автомат, эмулирующий работу телефона;
- разработать автомат, эмулирующий работу АТС;
- программно реализовать автомат;
- разработать визуализатор;
- выделить блоки проекта, не приспособленные для реализации с помощью автоматной технологии. Программирование этих блоков выполнено традиционным путем.

3. Описание

Данная программа реализована на стыке двух технологий на языке *Java*. Большая часть программы построена как каркас к описывающим систему автоматам. Собственно ядро программы, позволяющее запускаться этим автоматам и регулирующее их работу, реализовано с применением объектно-ориентированного программирования.

Телефон реагирует на набор событий, которые можно инициировать с помощью графического пользовательского интерфейса. Для набора телефонного номера существует два пути. Первый - набрать номер и поднять трубку. Телефон сам наберет валидный номер и соединится с абонентом (считается, что абонент всегда свободен). Валидным считается номер, длина которого превосходит семь символов (максимальный размер номера неограничен). Второй путь - сначала снять трубку, и только затем набрать валидный номер.

В проектируемой модели обе указанные возможности реализованы.

Кроме того, существует возможность сброса текущего номера, например, в случае ошибки.

Внешний вид эмулятора кнопочного телефона приведен на рис. 1. В верней части окна находится поле, предназначенное для показа набранного номера. В правой части интерфейса расположена зона информационных сообщений, где указано текущее состояние автомата, эмулирующего телефонный аппарат, и автомата, эмулирующего АТС.

В левой части интерфейса расположены кнопки, с их помощью можно набрать номер, снять/положить телефонную трубку и выйти из программы.

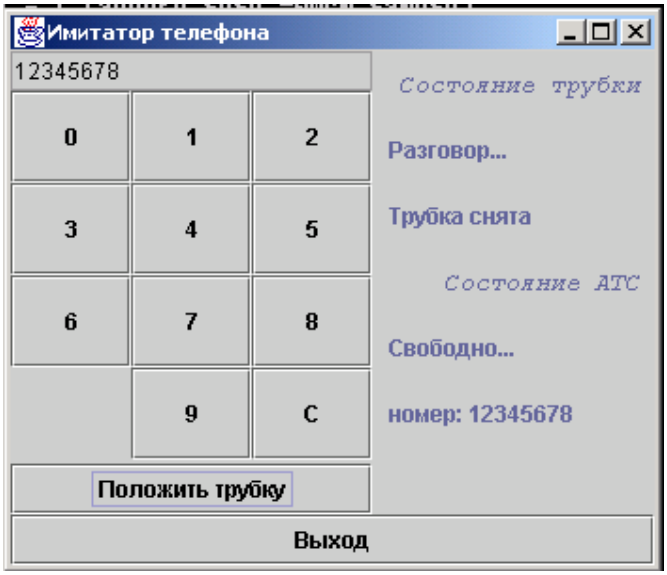

Рис. 1. Внешний вид эмулятора

4. Проектирование автоматов

4.1. Автомат, эмулирующий телефонный аппарат (A0)

Схема связей автомата приведена на рис.2., а граф переходов на рис. 3.

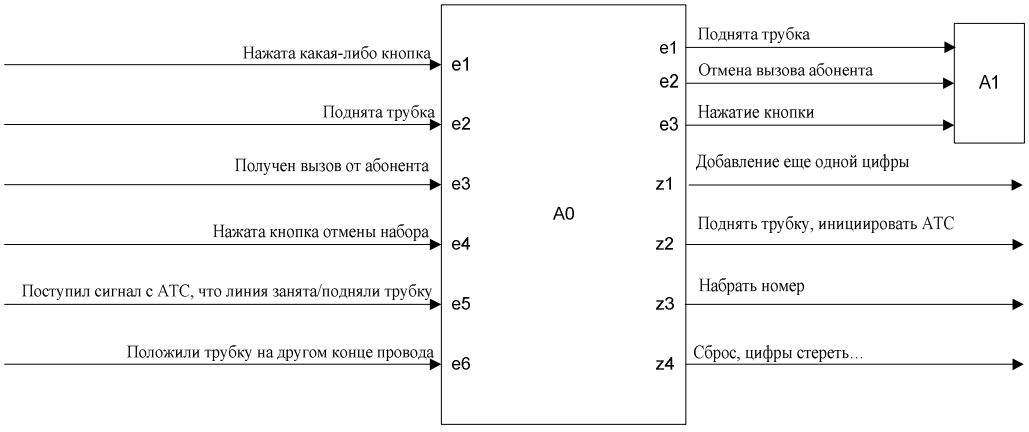

Рис.2. Схема связей автомата, эмулирующего телефонный аппарат

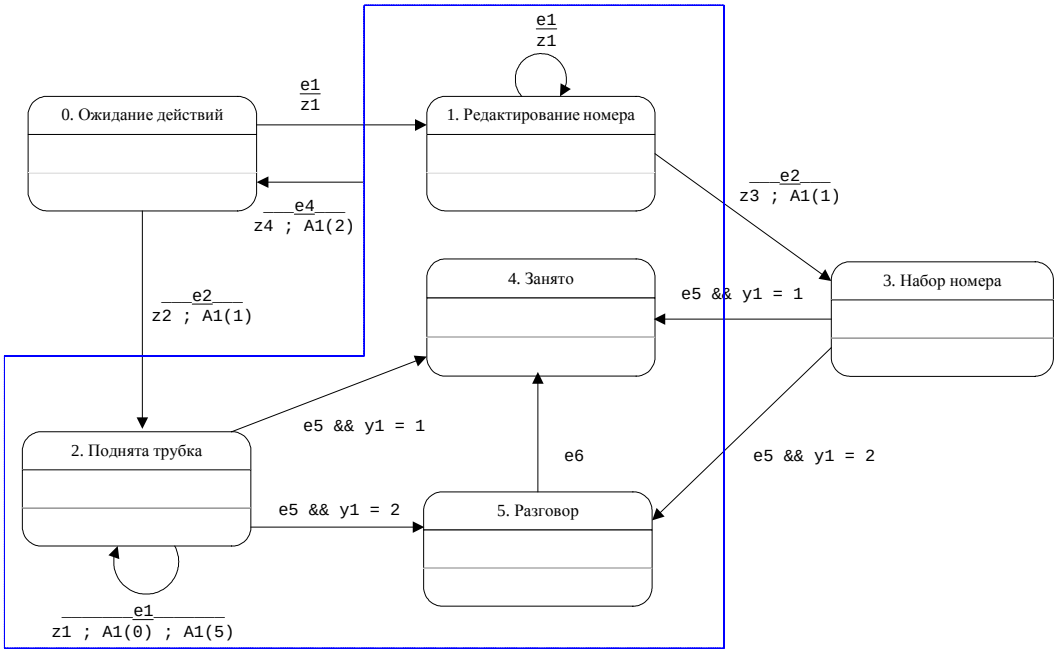

Рис. 3. Автомат, эмулирующий телефонный аппарат (A0)

4.2. Автомат, эмулирующий телефонную станцию (A1)

Схема связей автомата приведена на рис.4., а граф переходов на рис. 5.

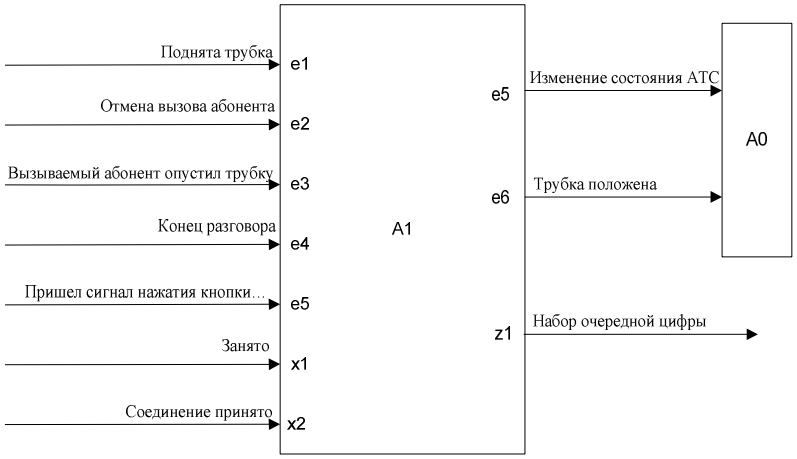

Рис. 4. Схема связей автомата, эмулирующего телефонную станцию

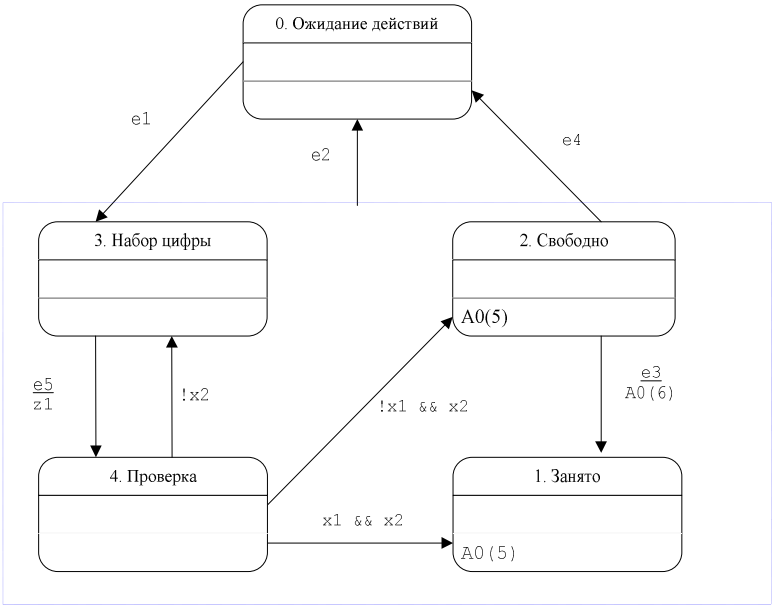

Рис. 5. Автомат, эмулирующий телефонную станция (A1)

5. Реализация программного кода

Перечень классов, используемых в программе, приведен на рис. 6. Исходные тексты программы - в Приложении.

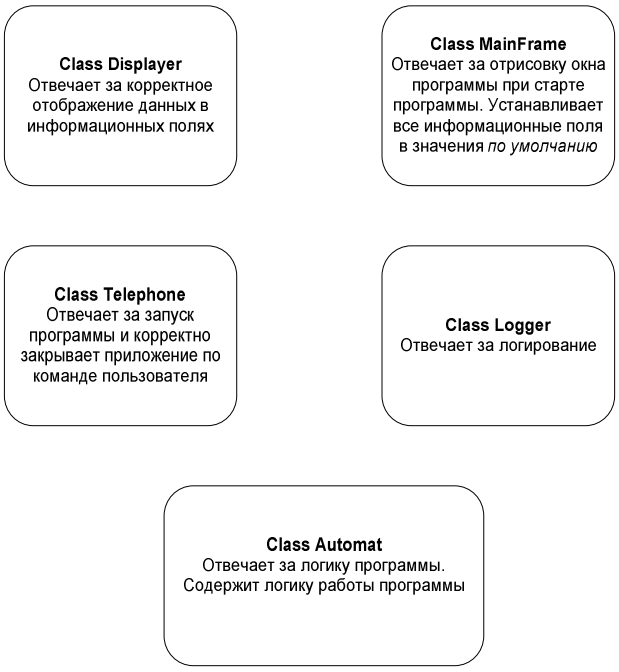

Рис. 6. Перечень классов программы

6. Заключение

Методика автоматного проектирования находится в начале жизненного пути. Она еще будет меняться и доводится до ума. Объектно-ориентированное проектирование не позволяет поддерживать современные большие проекты в приличной форме. Любой крупный проект обзаводится целым шлейфом из ошибок в программном коде, корень которых лежит в самом подходе к проектированию. Часть этих ошибок позволяет устранить автоматный подход. Само проектирование программы становится более наглядным и очевидным. Человек склонен ошибаться, и поэтому стоит предоставлять ему информацию в наиболее удобной для анализа форме, которой в части поведения являются графы переходов.

7. Источники

1. *Туккель Н.И., Шалыто А.А.* Система управления танком для игры Robocode (http://is.ifmo.ru/?i0=projects&i1=tanks)
2. *Туккель Н.И., Шалыто А.А.* Система управления дизель-генератором (http://is.ifmo.ru/?i0=projects&i1=dg)
3. *Наумов А.С., Шалыто А.А.* Система управления лифтом (http://is.ifmo.ru/?i0=projects&i1=elevator)

Приложение

Класс Automat.java

Класс, реализующий оба автомата.

```
package ru.ezhiki.srez.telephone;

import java.util.Vector;
import java.util.logging.Level;

public class Automat
{
    Logger Log = Logger.getLogger("current");
    public int y0 = 0;
    public int y1 = 0;

    public Vector numbersDialed = null;
    public Vector numbersDisplayed = null;

    public int lastNumberPressed = 0;

    public boolean isRecieverUp = false;

    public Automat()
    {
        numbersDialed = new Vector();
        numbersDisplayed = new Vector();
    }

    // Автомат A0
    public int A0(int e)
    {
        Log.logl( Level.INFO, "A0 (состояние - " + y0 + ") Начинает обработку события e
             "+e+" " );
        switch (e)
        {
            case 1:
                Log.log( Level.INFO, "(нажата какая-либо кнопка)" );
            break;
            case 2:
                Log.log( Level.INFO, "(поднята трубка)" );
            break;
```

```
            case 3:
                Log.log( Level.INFO, "(получен вызов от абонента)" );
            break;
            case 4:
                Log.log( Level.INFO, "(нажата кнопка отмены набора (положить трубку))" );
            break;
            case 5:
                Log.log( Level.INFO, "(поступил сигнал с LTT, что линия занята/подняли
                        трубку)" );
            break;
            case 6:
                Log.log( Level.INFO, "(положили трубку на том конце провода)" );
            break;
        }
        switch (y0)
        {
            case 0:
                if (e == 1)                { z0_1();               y0 = 1;}
                else if (e == 2)           { z0_2(); A1(1);        y0 = 2;}
            break;

            case 1:
                if (e == 1)                { z0_1();                 }
                else if (e == 2)           { z0_3(); A1(1);        y0 = 3;}
                else if (e == 4)           { z0_4(); A1(2);        y0 = 0;}
            break;

            case 2:
                if (e == 1)                { z0_1(); A1(0); A1(5);   }
                else if (e == 4)           { z0_4(); A1(2);        y0 = 0;}
                else if (e == 5 && y1 == 1) {                      y0 = 4;}
                else if (e == 5 && y1 == 2) {                      y0 = 5;}
            break;

            case 3:
                if (e == 5 && y1 == 1)     {                       y0 = 4;}
                else if (e == 5 && y1 == 2) {                      y0 = 5;}
            break;

            case 4:
                if (e == 4)                { z0_4(); A1(2);        y0 = 0;}
            break;

            case 5:
                if (e == 4)                { z0_4(); A1(2);        y0 = 0;}
                else if (e == 6)           {                       y0 = 4;}
            break;
        }
        return y0;
    }

    private void z0_1()
    {
        numbersDisplayed.add(new Integer(lastNumberPressed));
    }

    private void z0_2()
    {
        isRecieverUp = true;
    }

    private void z0_3()
    {
        for (int i = 0; i < numbersDisplayed.size() - 1; i++)
        {
            A1(5);
            A1(0);
        }
        A1(5);
        A1(0);
    }

    private void z0_4()
    {
        A1(3);
        numbersDisplayed.clear();
        numbersDialed.clear();
    }
```

```
    // Автомат A1
    public int A1(int e)
    {
      Log.log( Level.INFO, "A1 (состояние - " + y1 + ") начинает обработку события e"+e );
       switch (e)
       {
           case 1:
               Log.log( Level.INFO, "(поднята трубка)" );
           break;
           case 2:
               Log.log( Level.INFO, "(отмена вызова абонента)" );
           break;
           case 3:
               Log.log( Level.INFO, "(вызываемый опустил трубку)" );
           break;
           case 4:
               Log.log( Level.INFO, "(конец разговора)" );
           break;
           case 5:
               Log.log( Level.INFO, "(пришел сигнал нажатия кнопки…)" );
           break;
       }
       switch (y1)
       {
           case 0:
               if (e == 1)                {          y1 = 3;}
           break;

           case 1:
               A0(5);
               if (e == 2)                {          y1 = 0;}
           break;

           case 2:
               A0(5);
               if (e == 2)                {          y1 = 0;}
               else if (e == 3)           { A0(6);   y1 = 1;}
               else if (e == 4)           {          y1 = 0;}
           break;

           case 3:
               if (e == 2)                {          y1 = 0;}
               if (e == 5)                {z1_1();   y1 = 4;}
           break;

           case 4:
               if (e == 2)                {          y1 = 0;}
               else if (x1_1() && x1_2())  {          y1 = 1;}
               else if (!x1_1() && x1_2()) {          y1 = 2;}
               else if (!x1_2())          {          y1 = 3;}
           break;
       }
       return y1;
    }

    private void z1_1()
    {
        numbersDialed.add(numbersDisplayed.get(numbersDialed.size()));
    }

    private boolean x1_1()
    {
        return false;
    }

    private boolean x1_2()
    {
        return numbersDialed.size() >= 7;
    }
}
```

Класс Displayer.java

Класс, который отвечает за отрисовку корректных значений всех информационных полей в дилоговом окне приложения.

```
package ru.ezhiki.srez.telephone;
```

```
import java.util.logging.Level;

public class Displayer extends Thread
{
    Logger Log = Logger.getLogger("current");
    private MainFrame mainFramenull;
    private Automat automat = null;

    private long delay = 10;

    private long numbersDisplayed = 0;
    private int statePhone-1;
    private int stateATS = -1;

    public Displayer(MainFrame mainFrame, Automat automat)
    {
        this.mainFramemainFrame;
        this.automat = automat;
    }

    public void run()
    {
        super.run();

        while (true)
        {
            if (numbersDisplayed != automat.numbersDisplayed.size())
            {
                numbersDisplayed = automat.numbersDisplayed.size();
                String numbers = "";

                for (int i = 0; i < automat.numbersDisplayed.size(); i++)
                {
                    numbers += ((Integer)(automat.numbersDisplayed.get(i)));
                }
                mainFrame.editNumbers.setText(numbers);
            }

            if (statePhone != automat.y0)
            {
                statePhoneautomat.y0;

                if (statePhone == 0 || statePhone == 1)
                {
                    mainFrame.labelPhoneState2.setText("   Трубка положена");
                    mainFrame.buttonRecieverUpDown.setText("Поднять трубку ");
                    Log.log( Level.INFO, "Статус Трубки: Трубка положена" );
                }
                else
                {
                    mainFrame.labelPhoneState2.setText("   Трубка снята");
                    mainFrame.buttonRecieverUpDown.setText(" Положить трубку ");
                    Log.log( Level.INFO, "Статус Трубки: Трубка снята" );
                }

                switch (statePhone)
                {
                    case 0:
                        mainFrame.labelPhoneState1.setText("   Ожидание...");
                        Log.log( Level.INFO, "Статус Телефона: Ожидание" );
                    break;

                    case 1:
                        mainFrame.labelPhoneState1.setText("   Редактирование...");
                        Log.log( Level.INFO, "Статус Телефона: Редактирование" );
                    break;

                    case 2:
                        mainFrame.labelPhoneState1.setText("   Поднята трубка...");
                        Log.log( Level.INFO, "Статус Телефона: Поднята трубка" );
                    break;

                    case 3:
                        mainFrame.labelPhoneState1.setText("   Набираете  номер...");
                        Log.log( Level.INFO, "Статус Телефона: Набираете  номер" );
                    break;
```

```
                        case 4:
                            mainFrame.labelPhoneState1.setText("   Занято...");
                            Log.log( Level.INFO, "Статус Телефона: Занято" );
                        break;

                        case 5:
                            mainFrame.labelPhoneState1.setText("   Разговор...");
                            Log.log( Level.INFO, "Статус Телефона: Разговор" );
                        break;
                    }
                }

                if (stateATS != automat.y1 || stateATS == 3)
                {
                    stateATS = automat.y1;
                    String number = "   номер: ";

                    switch (stateATS)
                    {
                        case 0:
                            mainFrame.labelATSState1.setText("    Ожидание...");
                            Log.log( Level.INFO, "Статус АТС: Ожидание" );
                        break;

                        case 1:
                            mainFrame.labelATSState1.setText("   Занято...");
                            Log.log( Level.INFO, "Статус АТС: Занято" );
                        break;

                        case 2:
                            mainFrame.labelATSState1.setText("   Свободно...");
                            Log.log( Level.INFO, "Статус АТС: Свободно" );
                        break;

                        case 3:
                            mainFrame.labelATSState1.setText("   Набор номера...");
                        break;

                        case 4:
                            mainFrame.labelATSState1.setText("   Проверка...");
                            Log.log( Level.INFO, "Статус АТС: Проверка" );
                        break;
                    }

                    for (int i = 0; i < automat.numbersDisplayed.size(); i++)
                    {
                        number += ((Integer)(automat.numbersDisplayed.get(i)));
                    }
                    mainFrame.labelATSState2.setText(number);
                }

                try
                {
                    sleep(delay);
                }
                catch (InterruptedException e)
                {
                    e.printStackTrace();
                }
            }
        }
    }
```

Класс MainFrame.java

Класс создает и настраивает средствами языка *Java* диалоговое окно для работы программы. В последствии информационные поля будут обновляться классом *Displayer.java*.

```
package ru.ezhiki.srez.telephone;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.WindowEvent;

class ButtonActionListener implements ActionListener
{
```

```
            private int idx = 0;
            private Automat automat = null;

            public ButtonActionListener(int idx, Automat automat)
            {
                this.idx = idx;
                this.automat = automat;
            }

            public void actionPerformed(ActionEvent e)
            {
                automat.lastNumberPressed = idx;
                automat.A0(1);
            }
        }

        public class MainFrame extends JFrame
        {
            public Automat automat = new Automat();
            private Displayer displayer = null;

            private JButton buttons[] = new JButton[11];
            private JButton buttonExit = new JButton(" Выход ");
            public JButton buttonRecieverUpDown = new JButton(" Поднять трубку ");
            public JTextField editNumbers = new JTextField();

            private JLabel labelBeep = new JLabel(" Не гудит ");
            public JLabel labelPhoneState1 = new JLabel("  ", JLabel.LEFT);
            public JLabel labelPhoneState2 = new JLabel("  ", JLabel.LEFT);
            public JLabel labelATSState1 = new JLabel("  ", JLabel.LEFT);
            public JLabel labelATSState2 = new JLabel("  ", JLabel.LEFT);

            public MainFrame()
            {
                super("Имитатор телефона ");
                initButtons();
                createLayout();
                createActions();

                displayer = new Displayer(this, automat);

                pack();
                setSize(350, 300);

                Dimension screenSizeToolkit.getDefaultToolkit().getScreenSize();
                Dimension frameSizegetSize();
                if (frameSize.height > screenSize.height)
                {
                    frameSize.height = screenSize.height;
                }
                if (frameSize.width > screenSize.width)
                {
                    frameSize.width = screenSize.width;
                }
                setLocation((screenSize.width - frameSize.width)/2, (screenSize.height -
frameSize.height)/2);

                displayer.start();
            }

            private void initButtons()
            {
                buttons[0] = new JButton(" 0 ");
                buttons[1] = new JButton(" 1 ");
                buttons[2] = new JButton(" 2 ");
                buttons[3] = new JButton(" 3 ");
                buttons[4] = new JButton(" 4 ");
                buttons[5] = new JButton(" 5 ");
                buttons[6] = new JButton(" 6 ");
                buttons[7] = new JButton(" 7 ");
                buttons[8] = new JButton(" 8 ");
                buttons[9] = new JButton(" 9 ");
                buttons[10] = new JButton(" C ");

                editNumbers.setEditable(false);
            }

            private void createLayout()
            {
                JPanel panelMain = new JPanel(new BorderLayout());
```

```
    JPanel panelButtons = new JPanel(new GridLayout(4, 3));
    for (int i = 0; i < 9; i++)
        panelButtons.add(buttons[i]);

    panelButtons.add(new JPanel());
    panelButtons.add(buttons[9]);
    panelButtons.add(buttons[10]);

    JPanel panelPhonenew JPanel(new BorderLayout());
    panelPhone.add(BorderLayout.NORTH, editNumbers);
    panelPhone.add(BorderLayout.CENTER, panelButtons);
    panelPhone.add(BorderLayout.SOUTH, buttonRecieverUpDown);

    panelPhone.setPreferredSize(new Dimension(200, 250));

    JPanel panelStates = new JPanel(new GridLayout(7, 1));
    Font font = new Font("Monospaced", Font.ITALIC, 14);
    JLabel lab = new JLabel(" Состояние трубки ", JLabel.RIGHT);
    lab.setFont(font);
    panelStates.add(lab);
    panelStates.add(labelPhoneState1);
    panelStates.add(labelPhoneState2);
    JLabel lab2 = new JLabel(" Состояние LTT ", JLabel.RIGHT);        lab2.setFont(font);
    panelStates.add(lab2);
    panelStates.add(labelATSState1);
    panelStates.add(labelATSState2);

    panelStates.setPreferredSize(new Dimension(150, 250));

    panelMain.add(BorderLayout.CENTER, panelPhone);
    panelMain.add(BorderLayout.SOUTH, buttonExit);
    panelMain.add(BorderLayout.EAST, panelStates);

    getContentPane().setLayout(new BorderLayout());
    getContentPane().add(BorderLayout.CENTER, panelMain);
}

private void createActions()
{
    buttonExit.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            exitWindow();
        }
    });

    for (int i = 0; i < 10; i++)
        buttons[i].addActionListener(new ButtonActionListener(i, automat));

    buttons[10].addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            automat.A0(4);
        }
    });

    buttonRecieverUpDown.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            if (automat.y0 == 0 || automat.y0 == 1)
                automat.A0(2);
            else
                automat.A0(4);
        }
    });
}

protected void processWindowEvent(WindowEvent e)
{
    super.processWindowEvent(e);
    if (e.getID() == WindowEvent.WINDOW_CLOSING)
    {
        exitWindow();
    }
}
```

```
    private void exitWindow()
    {
        System.exit(0);
    }
}
```

Класс Telephone.java

Основной класс. Передает управлению автомату. После его отработки корректно завершает приложение по команде пользователя.

```
package ru.ezhiki.srez.telephone;

public class Telephone
{
    public static void main(String[] args)
    {
        MainFrame mainFramenew MainFrame();
        mainFrame.show();
    }
}
```

Класс Logger.java

Класс, ответственный за логирование.

```
package ru.ezhiki.srez.telephone;
import java.util.logging.Level;

public class Logger {

    public void log( Level l, String msg )
    {
        System.out.print(msg+"\n");
    }

    public static Logger getLogger(String s) {
        return new Logger();
    }

    public void logl(Level info, String msg) {
        System.out.print(msg);
    }
}
```

8. Пример лога

Пусть задан следующий сценарий.

1. Телефон в начальном состоянии. Трубка положена.
2. Трубка поднимается.
3. Набирается номер из 8 цифр (12345678).
4. Разговор
5. Трубка кладется.
6. Снова набирается номер (тот же самый, что и в пункте 3)
7. Трубка поднимается, номер автоматически набирается.
8. Разговор
9. Трубка снова кладется на телефон.

Этот сценарий рассматривает оба способа корректно набрать номер.
Ниже приведен протокол демонстрирующий, что программа реализует этот сценарий.

```
Статус Трубки: Трубка положена
Статус Телефона: Ожидание
Статус АТС: Ожидание
A0 (состояние - 0) Начинает обработку события e2 (поднята трубка)
A1 (состояние - 0) Начинает обработку события e1
```

(поднята трубка)
Статус Трубки: Трубка Снята
Статус Телефона: Поднята трубка
A0 (состояние - 2) Начинает обработку события e1 (нажата какая-либо кнопка)
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A0 (состояние - 2) Начинает обработку события e1 (нажата какая-либо кнопка)
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A0 (состояние - 2) Начинает обработку события e1 (нажата какая-либо кнопка)
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A0 (состояние - 2) Начинает обработку события e1 (нажата какая-либо кнопка)
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
Статус АТС: Проверка
A1 (состояние - 3) Начинает обработку события e0
A0 (состояние - 2) Начинает обработку события e1 (нажата какая-либо кнопка)
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A0 (состояние - 2) Начинает обработку события e1 (нажата какая-либо кнопка)
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A0 (состояние - 2) Начинает обработку события e1 (нажата какая-либо кнопка)
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 2) Начинает обработку события e0
A0 (состояние - 2) Начинает обработку события e5 (поступил сигнал с АТС, что линия занята/подняли трубку)
Статус Трубки: Трубка Снята
Статус Телефона: Разговор
Статус АТС: Свободно
A0 (состояние - 5) Начинает обработку события e1 (нажата какая-либо кнопка)
A0 (состояние - 5) Начинает обработку события e4 (нажата кнопка отмены набора (положить трубку))
A1 (состояние - 2) Начинает обработку события e3
(вызываемый опустил трубку)
A0 (состояние - 5) Начинает обработку события e5 (поступил сигнал с АТС, что линия занята/подняли трубку)
A0 (состояние - 5) Начинает обработку события e6 (положили трубку на том конце провода)
A1 (состояние - 1) Начинает обработку события e2
(отмена вызова абонента)
A0 (состояние - 4) Начинает обработку события e5 (поступил сигнал с АТС, что линия занята/подняли трубку)
Статус Трубки: Трубка положена
Статус Телефона: Ожидание
Статус АТС: Ожидание
A0 (состояние - 0) Начинает обработку события e1 (нажата какая-либо кнопка)
Статус Трубки: Трубка положена
Статус Телефона: Редактирование
A0 (состояние - 1) Начинает обработку события e1 (нажата какая-либо кнопка)

A0 (состояние - 1) Начинает обработку события e1 (нажата какая-либо кнопка)
A0 (состояние - 1) Начинает обработку события e1 (нажата какая-либо кнопка)
A0 (состояние - 1) Начинает обработку события e1 (нажата какая-либо кнопка)
A0 (состояние - 1) Начинает обработку события e1 (нажата какая-либо кнопка)
A0 (состояние - 1) Начинает обработку события e1 (нажата какая-либо кнопка)
A0 (состояние - 1) Начинает обработку события e1 (нажата какая-либо кнопка)
A0 (состояние - 1) Начинает обработку события e2 (поднята трубка)
A1 (состояние - 0) Начинает обработку события e1
(поднята трубка)
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e0
A1 (состояние - 3) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A1 (состояние - 4) Начинает обработку события e0
A1 (состояние - 2) Начинает обработку события e0
A0 (состояние - 3) Начинает обработку события e5 (поступил сигнал с АТС, что линия занята/подняли трубку)
A1 (состояние - 2) Начинает обработку события e5
(пришел сигнал нажатия кнопки:)
A0 (состояние - 5) Начинает обработку события e5 (поступил сигнал с АТС, что линия занята/подняли трубку)
A1 (состояние - 2) Начинает обработку события e0
A0 (состояние - 5) Начинает обработку события e5 (поступил сигнал с АТС, что линия занята/подняли трубку)
Статус Трубки: Трубка Снята
Статус Телефона: Разговор
Статус АТС: Свободно
A0 (состояние - 5) Начинает обработку события e4 (нажата кнопка отмены набора (положить трубку))
A1 (состояние - 2) Начинает обработку события e3
(вызываемый опустил трубку)
A0 (состояние - 5) Начинает обработку события e5 (поступил сигнал с АТС, что линия занята/подняли трубку)
A0 (состояние - 5) Начинает обработку события e6 (положили трубку на том конце провода)
A1 (состояние - 1) Начинает обработку события e2
(отмена вызова абонента)
A0 (состояние - 4) Начинает обработку события e5 (поступил сигнал с АТС, что линия занята/подняли трубку)
Статус Трубки: Трубка положена
Статус Телефона: Ожидание
Статус АТС: Ожидание