

Санкт-Петербургский государственный университет информационных технологий,
механики и оптики

Кафедра “Компьютерные технологии”

Е.В. Первушин, А.А. Шальто

Моделирование банкомата

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

Содержание	2
Введение.....	3
1. Постановка задачи	3
2. Диаграмма классов.....	5
3. Класс “TBankomatForm”	6
4. Класс “TAutomats”	7
4.1. Словесное описание	7
4.2. Нумерация и перечень событий (e)	7
4.3. Нумерация и перечень входных переменных (x).....	7
4.4. Нумерация и перечень выходных воздействий (z)	8
4.5. Автомат “Управление банкоматом” (A0).....	9
4.6. Автомат “Местонахождение денег” (A1).....	11
4.7. Автомат “Местонахождение карты” (A2)	12
5. Класс “TLog”	13
5.1. Словесное описание	13
5.2. Прототипы методов и их краткое описание.....	13
6. Класс “TCard”	14
7. Листинги программы	15
7.1. Bnk.pas	15
7.2. Automats.pas.....	21
7.3. Log_unit.pas	35
7.4. Cards.pas.....	38
Заключение	39

Введение

Для алгоритмизации и программирования задач логического управления и событийных систем была предложена SWITCH-технология. Подробно ознакомиться с этой технологией и конкретными примерами ее использования можно на сайте <http://is.ifmo.ru>.

Технология удобна для задач управления техническими объектами, такими как, например, банкомат, рассматриваемый в настоящей работе. Это связано с тем, что при использовании автоматного подхода, в частности, удается повысить централизацию логики управления в программном коде. Другое достоинство подхода состоит в том, что код является изоморфным графу переходов, по которому он строился. Это позволяет для понимания логики работы программы (ее поведения) не обращаться к текстам программ, а рассматривать лишь графы переходов.

В данной работе совместно применяются объектно-ориентированное и автоматное программирование, названное “*объектно-ориентированное программирование с явным выделением состояний*”. В рассматриваемом примере среди используемых классов только один является автоматным.

Визуализатор выполнен в виде приложения в среде *Delphi*.

1. Постановка задачи

Цель работы состоит в создании программы управления банкоматом. На панели (рис.1) находятся десять кнопок с цифрами, кнопки “Сброс”, “Отмена”, “Ввод”, “Начало”, “Протокол”, а также восемь обезличенных кнопок. Для вывода информации на панели имеется дисплей. Карта вводится через приемник карт (CARD). Чек выдается устройством его выдачи (RECEIPT), а деньги выдаются в правой нижней части панели.

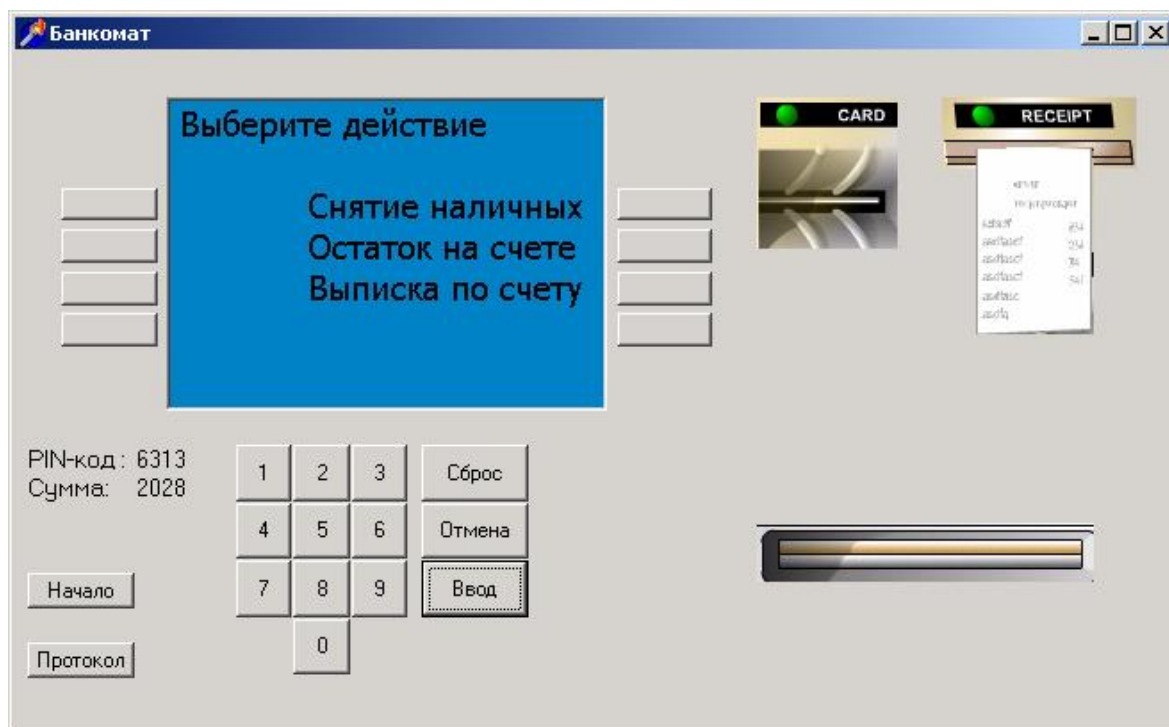


Рис. 1. Внешний вид визуализатора банкомата

PIN-код (персональный код) клиента генерируется при каждом обращении к банкомату. Также генерируется и сумма денег, находящаяся на счету клиента. Эта сумма находится в пределах от 100 до 5100.

Кнопка “Начало” нажимается, если в банкомате у клиента осталось, по его мнению, слишком мало денег, и он хочет начать “жить” сначала. При этом у него изменяется PIN-код и сумма, хранящаяся на счете.

При нажатии на кнопку “Протокол” на экране появляется окно, в котором осуществляется протоколирование (логирование) работы программы (рис.2).

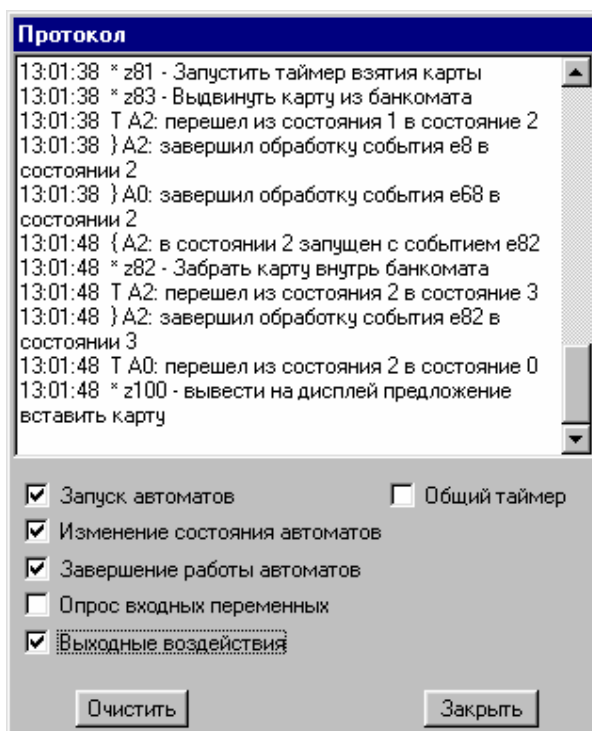


Рис.2. Протоколирование

Процесс работы с банкоматом состоит в следующем.

1. При запуске программы либо при нажатии кнопки “Начало” (если программа была запущена) на экране появляется PIN-код и сумма, находящаяся на счете пользователя. На дисплее формируется надпись “Вставьте карту”.
2. Щелчком левой кнопки мыши по изображению карты она перемещается в приемник карт. На дисплее появляется надпись “Введите ваш персональный код” (PIN).
3. С помощью оцифрованных кнопок вводится PIN-код. При нажатии на каждую оцифрованную кнопку на дисплее выводится символ “X”.
4. Нажимается кнопка “Ввод”. На дисплее появляется надпись “Выберите действие”, а соответствующим кнопкам присваиваются имена “Снятие наличных”, “Остаток на счете”, “Выписка по счету”.
5. При нажатии кнопок, соответствующих остатку на счете или выписке по счету, банкомат выдает чек и возвращает карту. На дисплее появляется надпись “Возьмите, пожалуйста, карту”. Если карту не удастся забрать в течение 20 секунд, то она

исчезает в приемнике карт, и ее можно получить, только заново нажав на кнопку “Начало”.

6. Для получения денег нажимается кнопка “Снятие наличных”, а на дисплее появляется надпись “Печатать чек?” Соответствующим обозначенным кнопкам присваиваются имена “Да” и “Нет”.
7. При нажатии любой из этих кнопок появляется надпись “Выберите сумму”, а обозначенным кнопкам присваиваются наименования: “3000” “2000” “1500” “1000” “500” “200” “100” и “Другая”.
8. При нажатии кнопки “Другая” на дисплее появляется надпись “Введите сумму”. По мере ее ввода, она отображается на дисплее и для ее получения необходимо нажать кнопку “Ввод”.
9. Появляется карта, которая должна быть взята в течение двадцати секунд, и чек, если он был затребован.
10. После взятия карты банкомат выдает деньги, которые также необходимо забрать в течение двадцати секунд.

Кнопка “Сброс” прерывает текущую операцию и возвращает карту, а кнопка “Отмена” позволяет ввести заново неверно набранную информацию.

2. Диаграмма классов

В программе основными являются класс *TBankomatForm*, предназначенный для визуализации работы банкомата, и класс *TAutomats*, который реализует все автоматы и является системой управления банкоматом.

Кроме основных, программа содержит также два вспомогательных класса.

Класс *TLog* протоколирует логику работу приложения и связан с соответствующим окном, предназначенным для вывода в него протокола.

Класс *TCard* предназначен для хранения информации о карте, и содержит ряд методов для управления картой.

Диаграмма классов приведена на рис. 3.



Рис. 3. Диаграмма классов

Ниже описывается каждый из перечисленных классов, причем наиболее подробно описан класс, содержащий автоматы.

3. Класс “TBankomatForm”

Класс является наследником встроенного в *Delphi* класса *TForm* и определяет окно визуализатора.

Класс *TBankomatForm* осуществляет визуализацию работы программы. При этом экземпляр этого класса получает внешние события и “препровождает” большинство из них экземпляру класса *TAutomats*, с помощью которого реализуется логика работы программы.

Рассматриваемый класс порождает форму, которая содержит дисплей (реализован в виде поля ввода типа *TMemo*), 23 кнопки (десять цифровых, три управляющих, восемь обозначенных и две дополнительные кнопки), а также изображения карты, денег и чека.

В обработчиках событий (нажатия на кнопки, на изображения и срабатывания таймера) вызываются соответствующие автоматы. Например, обработчик события *e21* “Нажатие кнопки 1” (кнопка типа *TButton* с названием *digit_1*) вызывает автомат *A0* с этим событием:

```

procedure TBankomatForm.digit_1Click(Sender: TObject);
begin
    aut.A0( 21 );
end;
  
```

По событиям, полученным от таймеров *MoneyTimer* и *CardTimer*, вызываются автоматы *A1* и *A2* с событиями *e72* и *e82* соответственно.

4. Класс “TAutomats”

4.1. Словесное описание

Класс реализует логику работы системы управления банкоматом. Конструктор принимает ссылку на поле дисплея (встроенный тип *TLines*).

В классе реализованы три автомата – методы вида *procedure An (e : integer)*, где *n* –номер автомата, *e* – событие. Состояние каждого автомата хранится в переменной *un*, начальное состояние всех автоматов – нулевое.

4.2. Нумерация и перечень событий (e)

0. Сработал общий таймер
1. Вставлена карта
3. Нажатие кнопки “Сброс”
4. Нажатие кнопки “Отмена”
5. Нажатие кнопки “Ввод”
7. Извлечь деньги
8. Вернуть карту
20. Нажатие кнопки “0”
21. Нажатие кнопки “1”
22. Нажатие кнопки “2”
23. Нажатие кнопки “3”
24. Нажатие кнопки “4”
25. Нажатие кнопки “5”
26. Нажатие кнопки “6”
27. Нажатие кнопки “7”
28. Нажатие кнопки “8”
29. Нажатие кнопки “9”
61. Нажатие первой обезличенной кнопки
62. Нажатие второй обезличенной кнопки
63. Нажатие третьей обезличенной кнопки
64. Нажатие четвертой обезличенной кнопки
65. Нажатие пятой обезличенной кнопки
66. Нажатие шестой обезличенной кнопки
67. Нажатие седьмой обезличенной кнопки
68. Нажатие восьмой обезличенной кнопки
71. Взятие денег клиентом
72. Сработал таймер извлечения денег
81. Взятие карты клиентом
82. Сработал таймер извлечения карты

4.3. Нумерация и перечень входных переменных (x)

1. Введено меньше четырех цифр PIN-кода
2. PIN-код введен неверно
3. Превышен лимит на карте
4. Требуется печать чека

4.4. Нумерация и перечень выходных воздействий (z)

4. Очистить поле ввода PIN-кода
5. Напечатать чек
7. Извлечь деньги из банкомата
30. Добавить цифру "0" к PIN-коду
31. Добавить цифру "1" к PIN-коду
32. Добавить цифру "2" к PIN-коду
33. Добавить цифру "3" к PIN-коду
34. Добавить цифру "4" к PIN-коду
35. Добавить цифру "5" к PIN-коду
36. Добавить цифру "6" к PIN-коду
37. Добавить цифру "7" к PIN-коду
38. Добавить цифру "8" к PIN-коду
39. Добавить цифру "9" к PIN-коду
40. Добавить цифру "0" к сумме
41. Добавить цифру "1" к сумме
42. Добавить цифру "2" к сумме
43. Добавить цифру "3" к сумме
44. Добавить цифру "4" к сумме
45. Добавить цифру "5" к сумме
46. Добавить цифру "6" к сумме
47. Добавить цифру "7" к сумме
48. Добавить цифру "8" к сумме
49. Добавить цифру "9" к сумме
61. Установить сумму равной 3000
62. Установить сумму равной 2000
63. Установить сумму равной 1500
64. Установить сумму равной 1000
65. Установить сумму равной 500
66. Установить сумму равной 200
67. Установить сумму равной 100
68. Очистить поле ввода суммы
71. Запустить таймер взятия денег
72. Забрать деньги внутрь банкомата
81. Запустить таймер взятия карты
82. Забрать карту внутрь банкомата
83. Извлечь карту из банкомата
100. Вывести на дисплей предложение вставить карту
101. Вывести на дисплей предложение ввести PIN-код
102. Вывести на дисплей сообщение о завершении транзакции
103. Вывести на дисплей меню
104. Вывести на дисплей предложение о выборе суммы
105. Вывести на дисплей предложение о печати чека
106. Вывести на дисплей предложение о вводе суммы
107. Вывести на дисплей сообщение о неверном PIN-коде
108. Вывести на дисплей сообщение о превышении лимита на карте
109. Вывести на дисплей предложение о взятии карты
110. Вывести на дисплей предложение о взятии денег

4.5. Автомат “Управление банкоматом” (A0)

Автомат является основным для управления банкоматом. Он отвечает за взаимодействие с клиентом и осуществляет:

- вывод справочной информации и меню на дисплей;
- ввод данных от клиента.

Он вызывает другие автоматы с определенными событиями.

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

Отметим, что событие $e0$ не участвует в условиях переходов, но автомат запускается по этому событию.

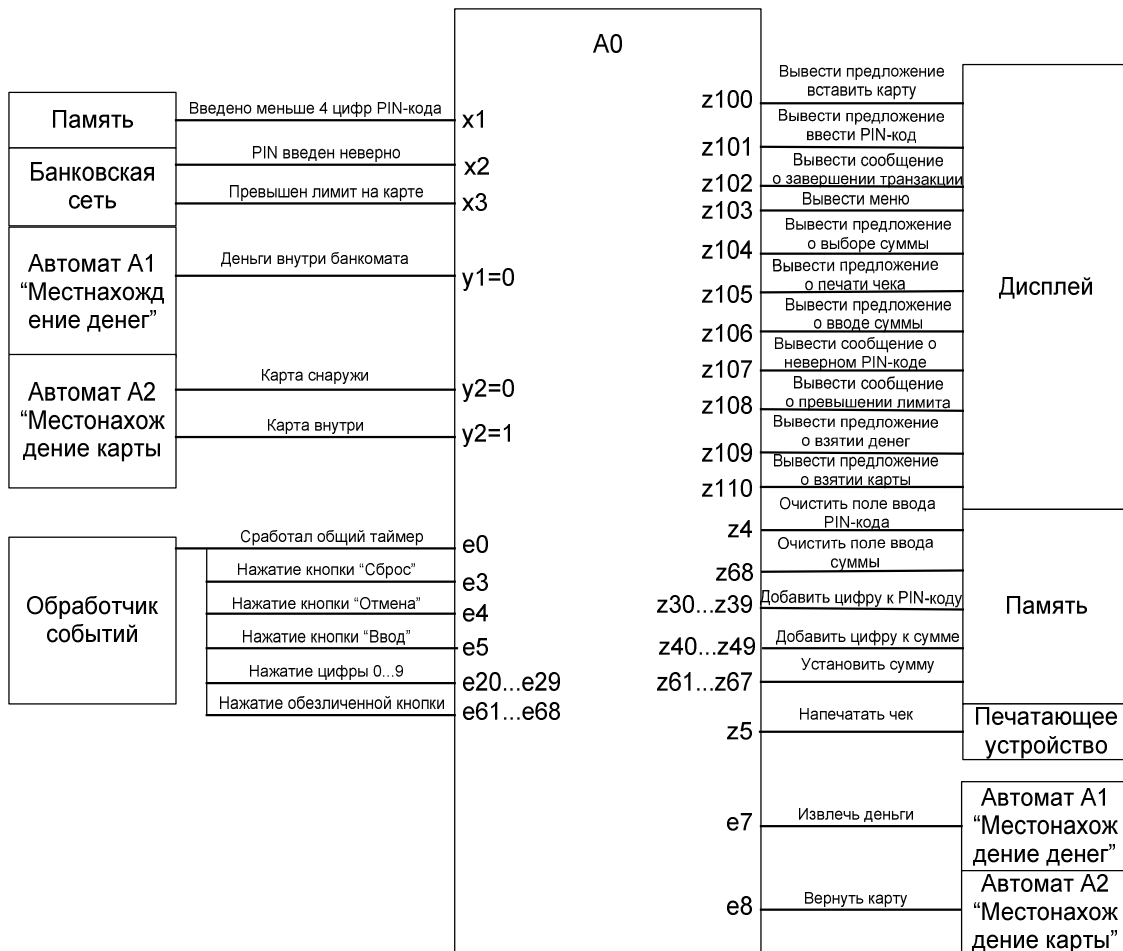


Рис. 4. Схема связей автомата “Управление банкоматом” (A0)

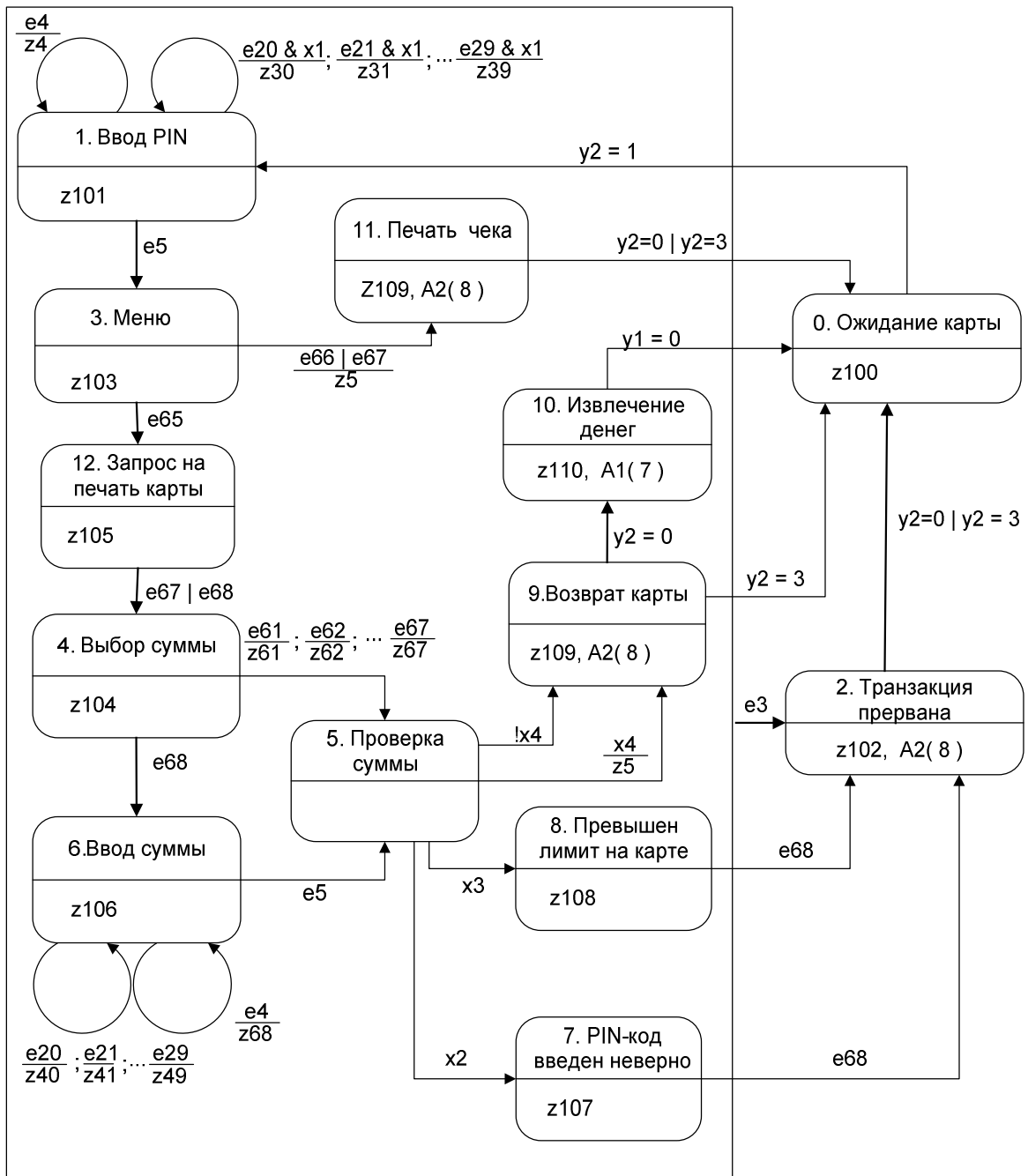


Рис. 5. Граф переходов автомата “Управление банкоматом” (A0)

Автомат реализован следующим образом (разд.7.2., метод A0 класса TAutomat).

В первом операторе *case* реализуются переходы и выполняются действия на них. Если состояние изменилось, то во втором операторе *case* выполняются действия в вершинах и вызываются автоматы с соответствующими событиями.

Следует отметить, что в других автоматах на вершинах не выполняется никаких действий и при их реализации второй оператор *case* не используется.

4.6. Автомат “Местонахождение денег” (A1)

Автомат предназначен для управления местонахождением денег. При поступлении события $e7$ “Извлечь деньги” банкомат должен обеспечить извлечение денег. Если деньги не будут взяты в течение времени, заданного таймером (двадцать секунд), то они будут втянуты внутрь банкомата.

Схема связей приведена на рис. 6, а граф переходов – на рис. 7.

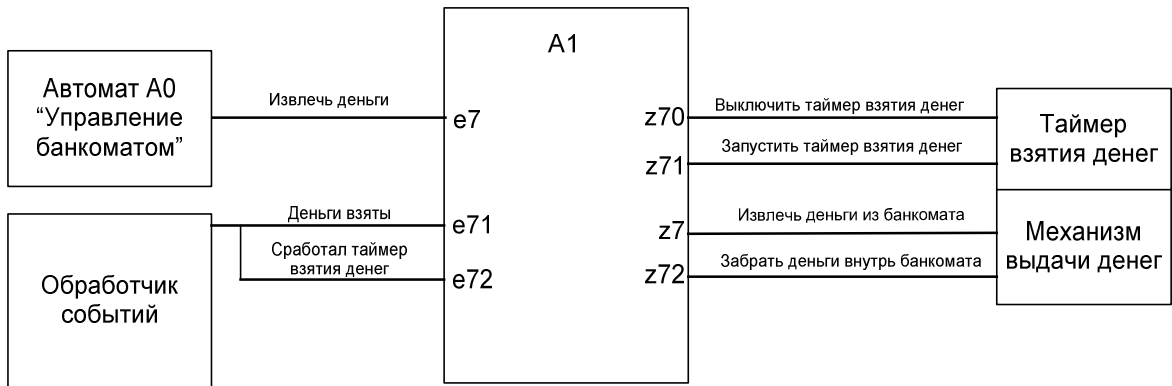


Рис. 6. Схема связей автомата “Местонахождение денег” (A1)

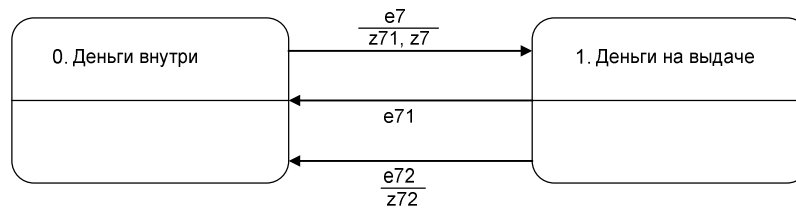


Рис. 7. Граф переходов автомата “Местонахождение денег” (A1)

4.7. Автомат “Местонахождение карты” (A2)

Автомат управляет местонахождением карты. При поступлении события $e8$ “Вернуть карту” банкомат должен выдвинуть карту наружу и, если она не будет взята клиентом по истечении времени, заданного таймером (двадцать секунд), она должна быть втянута в банкомат.

Схема связи этого автомата изображена на рис. 8, а граф переходов – на рис. 9.

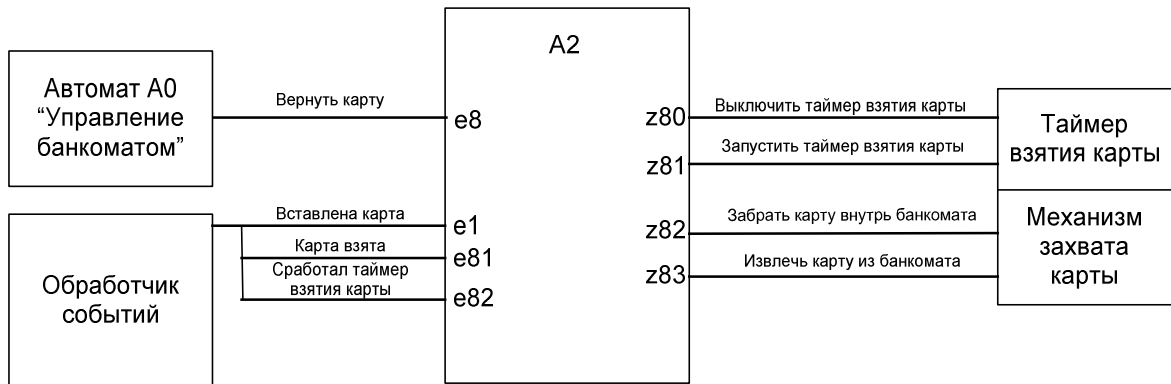


Рис. 8. Схема переходов автомата “Местонахождение карты” (A2)

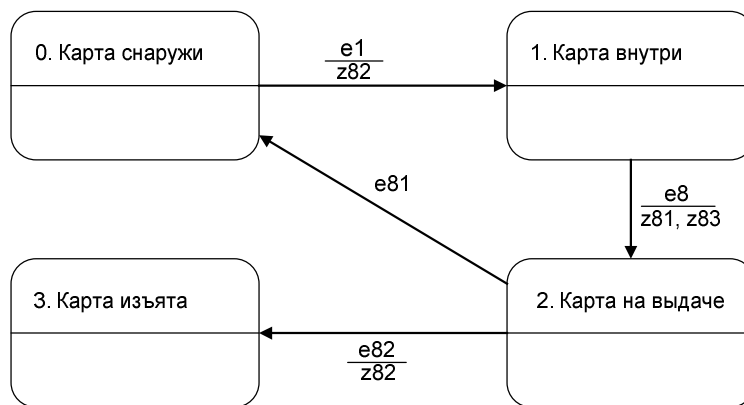


Рис. 9. Граф переходов автомата “Местонахождение карты” (A2)

5. Класс “TLog”

5.1. Словесное описание

Класс является наследником встроенного в *Delphi* класса *TForm*. Он реализует окно, отвечающее за вывод протокола, и содержит методы для управления протоколированием.

Протокол состоит из записей, которые автоматически добавляются в поле вывода (*TМето*). Подробность протокола регулируется набором флажков (*TCheckBox*). Поскольку автомат *AO* запускается по общему таймеру, то для отключения протоколирования по его срабатыванию предназначен специальный флажок.

5.2. Прототипы методов и их краткое описание

Приведем прототипы методов и их описание:

```
procedure add_begin( aut, state, event : integer );
```

Добавляет в протокол запись о запуске автомата *aut* в состоянии *state* с событием *event*

```
procedure add_trans( aut, old_state, new_state : integer );
```

Добавляет в протокол запись о переходе автомата *aut* из состояния *old_state* в состояние *new_state*

```
procedure add_end( aut, state, event : integer );
```

Добавляет в протокол запись о завершении автоматом *aut* обработки события *event* в состоянии *state*

```
procedure add_input( num : integer; str : string; ret :  
boolean);
```

Добавляет в протокол запись об опросе входной переменной *num* со словесным описанием *str*. Значение выходной переменной – *ret*.

```
procedure add_action( num : integer; str : string );
```

Добавляет в протокол запись о выполнении выходного воздействия *num* со словесным описанием *str*.

```
procedure add_error( aut : integer; err : string );
```

Добавляет в протокол запись об ошибке *err*

```
procedure add( str : String );
```

Добавляет в протокол строку *str*

6. Класс “TCard”

Класс хранит информацию о карте (персональный код, сумма) и содержит ряд методов для работы с ней:

```
procedure init;
```

Выполняет инициализацию карточки – присваивает ей случайный PIN-код и случайную сумму денег от 100 до 5100

```
function getPIN : string;
```

Возвращает персональный номер (PIN-код)

```
function getMoney : integer;
```

Возвращает остаток на счете

```
function withdraw( howmuch : integer ) : boolean;
```

снимает со счета деньги

7. Листинги программы

7.1. Bnk.pas

```
// Главная форма визуализатора
//
unit bnk;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, automats;

type
  TBankomatForm = class(TForm)
    btn_1: TButton; // Кнопки у дисплея: 8 штук
    btn_2: TButton;
    btn_3: TButton;
    btn_4: TButton;
    btn_5: TButton;
    btn_6: TButton;
    btn_7: TButton;
    btn_8: TButton;
    digit_1: TButton; // Кнопки для цифр: 10 штук
    digit_2: TButton;
    digit_3: TButton;
    digit_4: TButton;
    digit_5: TButton;
    digit_6: TButton;
    digit_7: TButton;
    digit_8: TButton;
    digit_9: TButton;
    digit_0: TButton;
    eject: TButton; // Сброс
    cancel: TButton; // Отмена
    enter: TButton; // Ввод
    Memol: TMemo; // Дисплей
    card_hole: TImage; // Изображение отверстия для
    // карты
    card: TImage; // Изображение карты
    t0: TTimer; // Общий таймер
    CardTimer: TTimer; // Таймер для карты
    money_hole: TImage; // Изображение для выдачи денег
    money: TImage; // Изображение денег
    Labell: TLabel; // Надпись для PIN-кода и
    // остатка на счете
    MoneyTimer: TTimer; // Таймер для денег
    receipt_hole: TImage; // Изображение щели для чека
    receipt: TImage; // Изображение чека
    reset: TButton; // Начать сначала
    sh_log: TButton; // Показать окно
    // протоколирования

    procedure digit_1Click(Sender: TObject);
    procedure digit_2Click(Sender: TObject);
```

```

    procedure digit_3Click(Sender: TObject);
    procedure digit_4Click(Sender: TObject);
    procedure digit_5Click(Sender: TObject);
    procedure digit_6Click(Sender: TObject);
    procedure digit_7Click(Sender: TObject);
    procedure digit_8Click(Sender: TObject);
    procedure digit_9Click(Sender: TObject);
    procedure digit_0Click(Sender: TObject);
    procedure btn_1Click(Sender: TObject);
    procedure btn_2Click(Sender: TObject);
    procedure btn_3Click(Sender: TObject);
    procedure btn_4Click(Sender: TObject);
    procedure btn_5Click(Sender: TObject);
    procedure btn_6Click(Sender: TObject);
    procedure btn_7Click(Sender: TObject);
    procedure btn_8Click(Sender: TObject);
    procedure ejectClick(Sender: TObject);
    procedure cancelClick(Sender: TObject);
    procedure enterClick(Sender: TObject);
    procedure t0Timer(Sender: TObject);
    procedure cardClick(Sender: TObject);
    procedure CardTimerTimer(Sender: TObject);
    procedure moneyClick(Sender: TObject);
    procedure MoneyTimerTimer(Sender: TObject);
    procedure receiptClick(Sender: TObject);
    procedure resetClick(Sender: TObject);
    procedure sh_logClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
private
    { Private declarations }
    aut : TAutomats;
public
    { Public declarations }
end;

var
    BankomatForm: TBankomatForm;

implementation

uses log_unit;

{$R *.DFM}

// Обработчики событий

procedure TBankomatForm.digit_1Click(Sender: TObject);
begin
    aut.A0( 21 );
end;

procedure TBankomatForm.digit_2Click(Sender: TObject);
begin
    aut.A0( 22 );
end;

procedure TBankomatForm.digit_3Click(Sender: TObject);
begin

```



```

        aut.A0( 23 );
    end;

    procedure TBankomatForm.digit_4Click(Sender: TObject);
    begin
        aut.A0( 24 );
    end;

    procedure TBankomatForm.digit_5Click(Sender: TObject);
    begin
        aut.A0( 25 );
    end;

    procedure TBankomatForm.digit_6Click(Sender: TObject);
    begin
        aut.A0( 26 );
    end;

    procedure TBankomatForm.digit_7Click(Sender: TObject);
    begin
        aut.A0( 27 );
    end;

    procedure TBankomatForm.digit_8Click(Sender: TObject);
    begin
        aut.A0( 28 );
    end;

    procedure TBankomatForm.digit_9Click(Sender: TObject);
    begin
        aut.A0( 29 );
    end;

    procedure TBankomatForm.digit_0Click(Sender: TObject);
    begin
        aut.A0( 20 );
    end;

    procedure TBankomatForm.btn_1Click(Sender: TObject);
    begin
        aut.A0( 61 );
    end;

    procedure TBankomatForm.btn_2Click(Sender: TObject);
    begin
        aut.A0( 62 );
    end;

    procedure TBankomatForm.btn_3Click(Sender: TObject);
    begin
        aut.A0( 63 );
    end;

    procedure TBankomatForm.btn_4Click(Sender: TObject);
    begin
        aut.A0( 64 );
    end;

```

```

procedure TBankomatForm.btn_5Click(Sender: TObject);
begin
    aut.A0( 65 );
end;

procedure TBankomatForm.btn_6Click(Sender: TObject);
begin
    aut.A0( 66 );
end;

procedure TBankomatForm.btn_7Click(Sender: TObject);
begin
    aut.A0( 67 );
end;

procedure TBankomatForm.btn_8Click(Sender: TObject);
begin
    aut.A0( 68 );
end;

procedure TBankomatForm.ejectClick(Sender: TObject);
begin
    aut.A0( 3 );
end;

procedure TBankomatForm.cancelClick(Sender: TObject);
begin
    aut.A0( 4 );
end;

procedure TBankomatForm.enterClick(Sender: TObject);
begin
    aut.A0( 5 );
end;

// Сработал общий таймер
procedure TBankomatForm.t0Timer(Sender: TObject);
begin
    // Визуализация: Взятие карты банкоматом
    if insert_card and not extract_card then begin
        if card.Height > 20 then
            card.Height := card.Height - 10
        else begin
            card.Visible := false;
            insert_card := false;
        end;
    end;

    // Визуализация: Извлечение карты
    if extract_card then begin
        if card.Height < 95 then begin
            card.Visible := true;
            card.Height := card.Height + 10;
        end else begin
            card.Height := 109;
            extract_card := false;
        end;
    end;
end;

```

```

// Визуализация: Извлечение денег
if drawout_money and not drawin_money then begin
  if money.Height < 65 then begin
    money.Height := money.Height + 8;
  end else begin
    money.Height := 73;
    drawout_money := false;
  end;
end;

// Визуализация: Взятие денег
if drawin_money then begin
  if money.Height > 10 then begin
    money.Height := money.Height - 8;
  end else begin
    money.Height := 0;
    drawin_money := false;
  end;
end;

// Визуализация: Печать чека
if print_receipt then begin
  if receipt.Height = 105 then receipt.Height := 5;
  if receipt.Height < 95 then begin
    receipt.Height := receipt.Height + 10;
  end else begin
    receipt.Height := 105;
    print_receipt := false;
  end;
end;
aut.A0( 0 );
end;

// Клик по изображению карты
procedure TBankomatForm.cardClick(Sender: TObject);
begin
  // Необходимо втянуть карту внутрь банкомата
  if ( aut.y2 = 0 ) and ( aut.y0 = 0 ) then begin
    aut.A2( 1 );
    card.top := 88;
  end;

  // Клиент забрал карту
  if aut.y2 = 2 then begin
    card.top := 136;
    aut.A2( 81 );
  end;
end;

procedure TBankomatForm.CardTimerTimer(Sender: TObject);
begin
  CardTimer.Enabled := false;
  aut.A2( 82 );
end;

procedure TBankomatForm.moneyClick(Sender: TObject);
begin
  aut.A1( 71 );
end;

```

```

    if money.height > 50 then money.Height := 0;
end;

procedure TBankomatForm.MoneyTimerTimer(Sender: TObject);
begin
    MoneyTimer.Enabled := false;
    aut.A1( 72 );
end;

procedure TBankomatForm.receiptClick(Sender: TObject);
begin
    receipt.Height := 0;
    print_receipt := false;
end;

// Клик по кнопке "Начало"
procedure TBankomatForm.resetClick(Sender: TObject);
begin
    aut.card.init;
    aut.card_info.Caption := 'PIN-код : ' + aut.card.getPIN + #13 +
        'Сумма: ' + IntToStr(aut.card.getMoney);

    receipt.Height := 0;
    card.Height := 109;
    card.Visible := true;
    card.Top := 136;
    money.Height := 0;
    aut.y0 := 0;
    aut.y1 := 0;
    aut.y2 := 0;
    insert_card := false;
    extract_card := false;
    print_receipt := false;
    drawout_money := false;
    drawin_money := false;
    receipt_requested := false;
    requested_money := 0;

    aut.z100;
end;

procedure TBankomatForm.sh_logClick(Sender: TObject);
begin
    Log.Show;
end;

procedure TBankomatForm.FormShow(Sender: TObject);
begin
    aut := TAutomats.Create(Memol.Lines);
    aut.card_timer := CardTimer;
    aut.money_timer := MoneyTimer;
    aut.card_info := Label1;
    aut.card.init;
    aut.card_info.Caption := 'PIN-код : ' + aut.card.getPIN + #13 +
        'Сумма: ' + IntToStr(aut.card.getMoney);
    receipt.Height := 0;
    aut.z100;
end;
end.

```

7.2. Automats.pas

```
// Автоматы - управление логикой банкомата
//
unit automats;

interface
uses stdctrls, classes, extctrls, sysutils, cards, log_unit;

const
    insert_card      : boolean = false;
    extract_card     : boolean = false;
    print_receipt    : boolean = false;
    drawout_money    : boolean = false;
    drawin_money     : boolean = false;
    receipt_requested : boolean = false;
    requested_money   : integer = 0;

type
    TAutomats = class
        screen      : TStrings;
        y0,
        y1,
        y2          : integer;
        pin,
        pin_mask    : String;
        card_timer,
        money_timer : TTimer;
        card        : TCard;
        card_info   : TLabel;
        constructor Create (lines : TStrings);

        procedure A0( e : integer );
        procedure A1( e : integer );
        procedure A2( e : integer );

        function x1 : boolean;           // Введено меньше четырех цифр
                                         // PIN-кода
        function x2 : boolean;           // PIN-код введен неверно
        function x3 : boolean;           // Превышен лимит на карте
        function x4 : boolean;           // Требуется печать чека

        procedure z100;                  // Вывести на дисплей
                                         // предложение вставить карту
        procedure z101;                  // Предложение ввести PIN-код
        procedure z102;                  // Вывести на дисплей сообщение
                                         // о завершении транзакции
        procedure z103;                  // Вывести на дисплей меню
        procedure z104;                  // Вывести на дисплей
                                         // предложение выбрать сумму
        procedure z105;                  // Вывести на дисплей
                                         // предложение печатать чек
        procedure z106;                  // Вывести на дисплей
                                         // предложение ввести сумму
        procedure z107;                  // Вывести на дисплей сообщение
    end;
end;
```

```

procedure z108; // о неверном вводе PIN
                // Вывести на дисплей сообщение
procedure z109; // о превышении лимита на карте
                // Вывести на дисплей
                // предложение забрать карту
procedure z110; // Вывести на дисплей
                // предложение взять деньги
procedure z30; // Добавить цифру "0" к PIN
procedure z31; // Добавить цифру "1" к PIN
procedure z32; // Добавить цифру "2" к PIN
procedure z33; // Добавить цифру "3" к PIN
procedure z34; // Добавить цифру "4" к PIN
procedure z35; // Добавить цифру "5" к PIN
procedure z36; // Добавить цифру "6" к PIN
procedure z37; // Добавить цифру "7" к PIN
procedure z38; // Добавить цифру "8" к PIN
procedure z39; // Добавить цифру "9" к PIN
procedure z5;  // Напечатать чек
procedure z4;  // Очистить поле ввода PIN-кода
procedure z40; // Добавить цифру "0" к сумме
procedure z41; // Добавить цифру "1" к сумме
procedure z42; // Добавить цифру "2" к сумме
procedure z43; // Добавить цифру "3" к сумме
procedure z44; // Добавить цифру "4" к сумме
procedure z45; // Добавить цифру "5" к сумме
procedure z46; // Добавить цифру "6" к сумме
procedure z47; // Добавить цифру "7" к сумме
procedure z48; // Добавить цифру "8" к сумме
procedure z49; // Добавить цифру "9" к сумме
procedure z61; // Установить сумму равной 3000
procedure z62; // Установить сумму равной 2000
procedure z63; // Установить сумму равной 1500
procedure z64; // Установить сумму равной 1000
procedure z65; // Установить сумму равной 500
procedure z66; // Установить сумму равной 200
procedure z67; // Установить сумму равной 100
procedure z68; // Очистить поле ввода суммы
procedure z69; // Установить необходимость
                // для печати чека
procedure z7;  // Извлечь деньги
procedure z71; // Запустить таймер взятия денег
procedure z72; // Забрать деньги внутрь
                // банкомата
procedure z81; // Запустить таймер взятия карты
procedure z82; // Забрать карту внутрь
                // банкомата
procedure z83; // Выдвинуть карту из банкомата
end;

```

implementation

```

constructor TAutomats.Create;
begin
    card    := TCard.Create;
    screen  := lines;
    y0      := 0;
    y1      := 0;
    y2      := 0;

```

```

end;

// Входные переменные

// Введено меньше четырех цифр PIN
function TAutomats.x1 : boolean;
var ret : boolean;
begin
    ret := length(pin) < 4;
    log.add_input( 1, 'введено меньше четырех цифр PIN', ret );
    x1 := ret;
end;

// PIN введен неверно
function TAutomats.x2 : boolean;
var ret : boolean;
begin
    ret := pin <> card.getPIN;
    log.add_input( 2, 'PIN введен неверно', ret );
    x2 := ret;
end;

// Превышен лимит на карте
function TAutomats.x3 : boolean;
var ret : boolean;
begin
    ret := requested_money > card.getMoney;
    log.add_input( 3, 'Превышен лимит на карте', ret );
    x3 := ret;
end;

// Требуется печать чека
function TAutomats.x4 : boolean;
var ret : boolean;
begin
    ret := receipt_requested;
    log.add_input( 3, 'Превышен лимит на карте', ret );
    x4 := ret;
end;

// Выходные воздействия

// Вывести на дисплей предложение вставить карту
procedure TAutomats.z100;
begin
    log.add_action( 100, 'Вывести на дисплей предложение вставить
    карту' );
    screen.Clear;
    screen.Add( 'Вставьте карту' );
    card_timer.Enabled := false;
    money_timer.Enabled := false;
end;

// Вывести на дисплей предложение ввести PIN
procedure TAutomats.z101;
begin

```

```

    log.add_action( 101, 'Вывести на дисплей предложение ввести PIN-
код' );
    screen.Clear;
    pin := ''; pin_mask := '';
    screen.Add( '    Введите ваш' );
    screen.Add( ' персональный код' );
end;

// Вывести на дисплей сообщение о завершении транзакции
procedure TAutomats.z102;
begin
    log.add_action( 102, 'Вывести на дисплей сообщение о завершении
транзакции' );
    screen.Clear;
    screen.Add( 'Транзакция прервана' );
    screen.Add( 'Возьмите карту' );
end;

// Вывести на дисплей меню
procedure TAutomats.z103;
begin
    log.add_action( 103, 'Вывести на дисплей меню' );
    screen.Clear;
    screen.Add( 'Выберите действие' );
    screen.Add( ' ' );
    screen.Add( '                Снятие наличных' );
    screen.Add( '                Остаток на счете' );
    screen.Add( '                Выписка по счету' );
end;

// Вывести на дисплей предложение выбрать сумму
procedure TAutomats.z104;
begin
    log.add_action( 104, 'Вывести на дисплей предложение выбрать
сумму' );
    requested_money := 0;
    screen.Clear;
    screen.Add( 'Выберите сумму' );
    screen.Add( ' ' );
    screen.Add( '3000                500' );
    screen.Add( '2000                200' );
    screen.Add( '1500                100' );
    screen.Add( '1000                Другая' );
end;

// Вывести на дисплей предложение печатать чек
procedure TAutomats.z105;
begin
    log.add_action( 105, 'Вывести на дисплей предложение о печати
чека' );
    screen.Clear;
    screen.Add( 'Печатать чек?' );
    screen.Add( ' ' );
    screen.Add( ' ' );
    screen.Add( ' ' );
    screen.Add( '                Да' );
    screen.Add( '                Нет' );
end;

```



```

// Вывести на дисплей предложение ввести сумму
procedure TAutomats.z106;
begin
    log.add_action(106, 'Вывести на дисплей предложение ввести сумму'
);
    screen.Clear;
    screen.Add( 'Введите сумму:' );
end;

// Вывести на дисплей сообщение о неверном вводе PIN
procedure TAutomats.z107;
begin
    log.add_action(107, 'Вывести на дисплей сообщение о неверном вводе
PIN-кода' );
    screen.Clear;
    screen.Add( '      Неправильный' );
    screen.Add( ' персональный код' );
    screen.Add( ' повторить ввод?' );
    screen.Add( ' ' );
    screen.Add( '                                Да' );
    screen.Add( '                                Нет' );
end;

// Вывести на дисплей сообщение о превышении лимита на карте
procedure TAutomats.z108;
begin
    log.add_action( 108, 'Вывести на дисплей сообщение о превышении
лимита на карте' );
    screen.Clear;
    screen.Add( 'Превышен лимит на карте' );
    screen.Add( 'повторить ввод суммы?' );
    screen.Add( ' ' );
    screen.Add( ' ' );
    screen.Add( '                                Да' );
    screen.Add( '                                Нет' );
end;

// Вывести на дисплей предложение забрать карту
procedure TAutomats.z109;
begin
    log.add_action( 109, 'вывести на дисплей предложение забрать
карту' );
    screen.Clear;
    screen.Add( 'Возьмите, пожалуйста,' );
    screen.Add( ' карту' );
end;

// Вывести на дисплей предложение взять деньги
procedure TAutomats.z110;
begin
    log.add_action(110, 'Вывести на дисплей предложение взять деньги'
);
    screen.Clear;
    screen.Add( 'Возьмите деньги' );
end;

```

```

// Добавить цифру "0" к PIN
procedure TAutomats.z30;
begin
    log.add_action( 30, 'Добавить цифру "0" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '0';
    screen[2] := '      ' + pin_mask;
end;

// Добавить цифру "1" к PIN
procedure TAutomats.z31;
begin
    log.add_action( 31, 'Добавить цифру "1" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '1';
    screen[2] := '      ' + pin_mask;
end;

// Добавить цифру "2" к PIN
procedure TAutomats.z32;
begin
    log.add_action( 32, 'Добавить цифру "2" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '2';
    screen[2] := '      ' + pin_mask;
end;

// Добавить цифру "3" к PIN
procedure TAutomats.z33;
begin
    log.add_action( 33, 'Добавить цифру "3" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '3';
    screen[2] := '      ' + pin_mask;
end;

// Добавить цифру "4" к PIN
procedure TAutomats.z34;
begin
    log.add_action( 34, 'Добавить цифру "4" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '4';
    screen[2] := '      ' + pin_mask;
end;

// Добавить цифру "5" к PIN
procedure TAutomats.z35;
begin
    log.add_action( 35, 'Добавить цифру "5" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '5';
    screen[2] := '      ' + pin_mask;
end;

```

```

// Добавить цифру "6" к PIN
procedure TAutomats.z36;
begin
    log.add_action( 36, 'Добавить цифру "6" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '6';
    screen[2] := '      ' + pin_mask;
end;

// Добавить цифру "7" к PIN
procedure TAutomats.z37;
begin
    log.add_action( 37, 'Добавить цифру "7" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '7';
    screen[2] := '      ' + pin_mask;
end;

// Добавить цифру "8" к PIN
procedure TAutomats.z38;
begin
    log.add_action( 38, 'Добавить цифру "8" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '8';
    screen[2] := '      ' + pin_mask;
end;

// Добавить цифру "9" к PIN
procedure TAutomats.z39;
begin
    log.add_action( 39, 'Добавить цифру "9" к PIN' );
    pin_mask := pin_mask + 'X';
    pin      := pin + '9';
    screen[2] := '      ' + pin_mask;
end;

// Напечатать чек
procedure TAutomats.z5;
begin
    log.add_action( 5, 'Напечатать чек' );
    print_receipt := true;
end;

// Очистить поле ввода PIN
procedure TAutomats.z4;
begin
    log.add_action( 4, 'Очистить поле ввода PIN' );
    pin := '';
    pin_mask := '';
    screen[2] := '';
end;

// Добавить цифру "0" к сумме
procedure TAutomats.z40;
begin
    log.add_action( 40, 'Добавить цифру "0" к сумме' );
    requested_money := requested_money * 10 + 0;
    screen[1] := '      ' + IntToStr( requested_money );
end;

```

```

end;

// Добавить цифру "1" к сумме
procedure TAutomats.z41;
begin
    log.add_action( 41, 'Добавить цифру "1" к сумме' );
    requested_money := requested_money * 10 + 1;
    screen[1] := '    ' + IntToStr( requested_money );
end;

// Добавить цифру "2" к сумме
procedure TAutomats.z42;
begin
    log.add_action( 42, 'Добавить цифру "2" к сумме' );
    requested_money := requested_money * 10 + 2;
    screen[1] := '    ' + IntToStr( requested_money );
end;

// Добавить цифру "3" к сумме
procedure TAutomats.z43;
begin
    log.add_action( 43, 'Добавить цифру "3" к сумме' );
    requested_money := requested_money * 10 + 3;
    screen[1] := '    ' + IntToStr( requested_money );
end;

// Добавить цифру "4" к сумме
procedure TAutomats.z44;
begin
    log.add_action( 44, 'Добавить цифру "4" к сумме' );
    requested_money := requested_money * 10 + 4;
    screen[1] := '    ' + IntToStr( requested_money );
end;

// Добавить цифру "5" к сумме
procedure TAutomats.z45;
begin
    log.add_action( 45, 'Добавить цифру "5" к сумме' );
    requested_money := requested_money * 10 + 5;
    screen[1] := '    ' + IntToStr( requested_money );
end;

// Добавить цифру "6" к сумме
procedure TAutomats.z46;
begin
    log.add_action( 46, 'Добавить цифру "6" к сумме' );
    requested_money := requested_money * 10 + 6;
    screen[1] := '    ' + IntToStr( requested_money );
end;

// Добавить цифру "7" к сумме
procedure TAutomats.z47;
begin
    log.add_action( 47, 'Добавить цифру "7" к сумме' );
    requested_money := requested_money * 10 + 7;
    screen[1] := '    ' + IntToStr( requested_money );
end;

```

```

// Добавить цифру "8" к сумме
procedure TAutomats.z48;
begin
    log.add_action( 48, 'Добавить цифру "8" к сумме' );
    requested_money := requested_money * 10 + 8;
    screen[1] := '    ' + IntToStr( requested_money );
end;

// Добавить цифру "9" к сумме
procedure TAutomats.z49;
begin
    log.add_action( 49, 'Добавить цифру "9" к сумме' );
    requested_money := requested_money * 10 + 9;
    screen[1] := '    ' + IntToStr( requested_money );
end;

// Установить сумму равной 3000
procedure TAutomats.z61;
begin
    log.add_action( 61, 'Установить сумму равной 3000' );
    requested_money := 3000;
end;

// Установить сумму равной 2000
procedure TAutomats.z62;
begin
    log.add_action( 62, 'Установить сумму равной 2000' );
    requested_money := 2000;
end;

// Установить сумму равной 1500
procedure TAutomats.z63;
begin
    log.add_action( 63, 'Установить сумму равной 1500' );
    requested_money := 1500;
end;

// Установить сумму равной 1000
procedure TAutomats.z64;
begin
    log.add_action( 64, 'Установить сумму равной 1000' );
    requested_money := 1000;
end;

// Установить сумму равной 500
procedure TAutomats.z65;
begin
    log.add_action( 65, 'Установить сумму равной 500' );
    requested_money := 500;
end;

// Установить сумму равной 200
procedure TAutomats.z66;
begin
    log.add_action( 66, 'Установить сумму равной 200' );
    requested_money := 200;
end;

```

```

// Установить сумму равной 100
procedure TAutomats.z67;
begin
    log.add_action( 67, 'Установить сумму равной 100' );
    requested_money := 100;
end;

// Очистить поле ввода суммы
procedure TAutomats.z68;
begin
    log.add_action( 68, 'Очистить поле ввода суммы' );
    requested_money := 0;
    screen[1] := ' ';
end;

// Установить необходимость печати чека
procedure TAutomats.z69;
begin
    log.add_action( 69, 'Установить необходимость печати чека' );
    receipt_requested := true;
end;

// Выдвинуть деньги
procedure TAutomats.z7;
begin
    log.add_action( 7, 'Выдвинуть деньги' );
    drawout_money := true;
    card.withdraw( requested_money );
    card_info.Caption := 'PIN-код : ' + card.getPIN + #13 +
        'Сумма: ' + IntToStr(card.getMoney);
end;

// Запустить таймер взятия денег
procedure TAutomats.z71;
begin
    log.add_action(71, 'Запустить таймер взятия денег' );
    money_timer.Enabled := true;
end;

// Забрать деньги внутрь банкомата
procedure TAutomats.z72;
begin
    log.add_action( 72, 'Забрать деньги внутрь банкомата' );
    drawin_money := true;
end;

// Запустить таймер взятия карты
procedure TAutomats.z81;
begin
    log.add_action( 81, 'Запустить таймер взятия карты' );
    card_timer.Enabled := true;
end;

// Забрать карту внутрь банкомата
procedure TAutomats.z82;
begin
    log.add_action( 82, 'Забрать карту внутрь банкомата' );
    insert_card := true;
end;

```

```

end;

// Выдвинуть карту из банкомата
procedure TAutomats.z83;
begin
    log.add_action( 83, 'Выдвинуть карту из банкомата' );
    extract_card := true;
end;

//
// Автомат A0
//
procedure TAutomats.A0( e : integer );
var
    y_old : integer;
begin
    y_old := y0;
    log.add_begin( 0, y0, e );
    case y0 of
        0:
            if y2 = 1 then begin
                y0 := 1;
            end;
        1:
            if x1 and ( e = 20 ) then begin
                z30; y0 := 1;
            end else if x1 and ( e = 21 ) then begin
                z31; y0 := 1;
            end else if x1 and ( e = 22 ) then begin
                z32; y0 := 1;
            end else if x1 and ( e = 23 ) then begin
                z33; y0 := 1;
            end else if x1 and ( e = 24 ) then begin
                z34; y0 := 1;
            end else if x1 and ( e = 25 ) then begin
                z35; y0 := 1;
            end else if x1 and ( e = 26 ) then begin
                z36; y0 := 1;
            end else if x1 and ( e = 27 ) then begin
                z37; y0 := 1;
            end else if x1 and ( e = 28 ) then begin
                z38; y0 := 1;
            end else if x1 and ( e = 29 ) then begin
                z39; y0 := 1;
            end else if e = 4 then begin
                z4; y0 := 1;
            end else if e = 5 then begin
                y0 := 3;
            end else if e = 3 then begin
                y0 := 2;
            end;
        2:
            if ( y2 = 0 ) or ( y2 = 3 ) then begin
                y0 := 0;
            end;
    end;
end;

```

```

3:
  if e = 65 then begin
    end else if ( e = 66 ) or ( e = 67 ) then begin
      z5;
    end else if e = 3 then begin
      end;
4:
  if e = 61 then begin
    z61;
  end else if e = 62 then begin
    z62;
  end else if e = 63 then begin
    z63;
  end else if e = 64 then begin
    z64;
  end else if e = 65 then begin
    z65;
  end else if e = 66 then begin
    z66;
  end else if e = 67 then begin
    z67;
  end else if e = 68 then begin
    end else if e = 3 then begin
      end;
5:
  if x2 then begin
    end else if x3 then begin
    end else if not x4 then begin
    end else if x4 then begin
      z5;
    end;
6:
  if e = 20 then begin
    z40;
  end else if e = 21 then begin
    z41;
  end else if e = 22 then begin
    z42;
  end else if e = 23 then begin
    z43;
  end else if e = 24 then begin
    z44;
  end else if e = 25 then begin
    z45;
  end else if e = 26 then begin
    z46;
  end else if e = 27 then begin
    z47;
  end else if e = 28 then begin
    z48;

```



```

end else if e = 29 then begin
    z49;
    y0 := 6;
end else if e = 4 then begin
    z68;
    y0 := 6;
end else if e = 5 then begin
    y0 := 5;
end else if e = 3 then begin
    y0 := 2;
end;
7:
if e = 67 then begin
    y0 := 1;
end else if ( e = 68 ) or ( e = 3 ) then begin
    y0 := 2;
end;
8:
if e = 67 then begin
    y0 := 4;
end else if ( e = 68 ) or ( e = 3 ) then begin
    y0 := 2;
end;
9:
if y2 = 0 then begin
    y0 := 10;
end else if y2 = 3 then begin
    y0 := 0;
end else if e = 3 then begin
    y0 := 2;
end;
10:
if y1 = 0 then begin
    y0 := 0;
end else if e = 3 then begin
    y0 := 2;
end;
11:
if ( y2 = 0 ) or ( y2 = 3 ) or ( e = 3 ) then begin
    y0 := 0;
end;
12:
if e = 67 then begin
    z69;
    y0 := 4;
end else if e = 68 then begin
    y0 := 4;
end else if e = 3 then begin
    y0 := 2;
end;
else
log.add_error( 0, 'Неизвестное состояние' );
end;
if y0 <> y_old then begin
log.add_trans( 0, y_old, y0 );
case y0 of
0:
z100;
1:
z101;

```

```

2:
  begin
    z102; A2( 8 );
  end;
3:
  z103;
4:
  z104;
6:
  z106;
7:
  z107;
8:
  z108;
9:
  begin
    z109; A2( 8 );
  end;
10:
  begin
    z110; A1( 7 );
  end;
11:
  begin
    z109; A2( 8 );
  end;
12:
  z105;
end;
log.add_end( 0, y0, e );
end;

//
// Автомат A1
//
procedure TAutomats.A1( e : integer );
var y_old : integer;
begin
  y_old := y1;
  log.add_begin( 1, y1, e );
  case y1 of
    0:
      if e = 7 then begin
        z71; z7;
        y1 := 1;
      end;
    1:
      if e = 71 then begin
        y1 := 0;
      end else if e = 72 then begin
        z72;
        y1 := 0;
      end;
  else
    log.add_error( 1, 'Неизвестное состояние' );
  end;
  if y_old <> y1 then
    log.add_trans( 1, y_old, y1 );
end;

```

```

    log.add_end( 1, y1, e );
end;

//
// Автомат А2
//
procedure TAutomats.A2( e : integer );
var y_old : integer;
begin
    y_old := y2;
    log.add_begin( 2, y2, e );
    case y2 of
        0:
            if e = 1 then begin
                z82;                                y2 := 1;
            end;
        1:
            if e = 8 then begin
                z81; z83;                            y2 := 2;
            end;
        2:
            if e = 81 then begin
                y2 := 0;
            end else if e = 82 then begin
                z82;                                y2 := 3;
            end;
    else
        log.add_error( 2, 'Неизвестное состояние' );
    end;
    if y_old <> y2 then
        log.add_trans( 2, y_old, y2 );
    log.add_end( 2, y2, e );
end;

end.

```

7.3. Log_unit.pas

```

// Ведение протокола
//
unit log_unit;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs,
    StdCtrls;

type
    TLog = class(TForm)
        log_memo: TMemo;
        Button1: TButton;
        Button2: TButton;
        logBegin: TCheckBox;
    end;

```

```

    logTrans: TCheckBox;
    logEnd: TCheckBox;
    logInput: TCheckBox;
    logAction: TCheckBox;
    logt0: TCheckBox;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
    procedure add_log( str : String );

// Начало работы автомата
    procedure add_begin( aut, state, event : integer );

// Переход между состояниями
    procedure add_trans( aut, old_state, new_state : integer );

// Окончание работы автомата
    procedure add_end( aut, state, event : integer );

// Опрос входной переменной
    procedure add_input( num : integer; str : string; ret : boolean);

// Выполнение действия
    procedure add_action( num : integer; str : string );

// Сообщение об ошибке
    procedure add_error( aut : integer; err : string );

// Вывод строки в протокол
    procedure add( str : String );
end;

var
    Log: TLog;

implementation

{$R *.DFM}

procedure TLog.Button1Click(Sender: TObject);
begin
    Hide;
end;

procedure TLog.add( str : String );
begin
    log_memo.Lines.Add(TimeToStr(Time) + ' ' + str);
end;

procedure TLog.add_log( str : String );
begin
    add( '# ' + str );
end;

procedure TLog.add_begin( aut, state, event : integer );

```

```

begin
  if ( event = 0 ) and ( not logt0.Checked ) then exit;
  if logBegin.Checked then
    add( '{ A' + IntToStr( aut ) + ': в состоянии ' + IntToStr(
state ) +
          ' запущен с событием е' + IntToStr( event ) );
end;

procedure TLog.add_trans( aut, old_state, new_state : integer );
begin
  if logTrans.Checked then
    add( 'T A' + IntToStr( aut ) + ': перешел из состояния ' +
          IntToStr( old_state ) + ' в состояние ' +
IntToStr(new_state) );
end;

procedure TLog.add_end( aut, state, event : integer );
begin
  if ( event = 0 ) and ( not logt0.Checked ) then exit;
  if logEnd.Checked then
    add( '} A' + IntToStr( aut ) + ': завершил обработку события е'
+
          IntToStr( event ) + ' в состоянии ' + IntToStr( state ));
end;

procedure TLog.add_input( num : integer; str : string; ret : boolean);
var s : String;
begin
  if ret then s := 'true' else s := 'false';
  if logInput.Checked then
    add( '> x' + IntToStr( num ) + ' - ' + str + ' - вернул ' + s
);
end;

procedure TLog.add_action( num : integer; str : string );
begin
  if logAction.Checked then
    add( '* z' + IntToStr( num ) + ' - ' + str );
end;

procedure TLog.add_error( aut : integer; err : string );
begin
  add( '! A' + IntToStr( aut ) + ': ОШИБКА: ' + err );
end;

procedure TLog.Button2Click(Sender: TObject);
begin
  log_memo.Clear;
end;

end.

```

7.4. Cards.pas

```
// Класс TCard - работа с картой
//
unit cards;

interface

type
  TCard = class
    procedure init;
    function getPIN : string;
    function getMoney : integer;
    function withdraw( howmuch : integer ) : boolean;
    constructor Create;
  private
    pin : string;           // Персональный код
    money : integer;       // Остаток на счете
  end;

implementation

// Установить PIN и сумму денег случайными
procedure TCard.init;
var i : integer;
begin
  pin := '';
  for i := 1 to 4 do begin
    pin := pin + char(ord( '0' )+random( 10 ));
  end;
  money := 100 + random( 5000 );
end;

// Получить сумму денег
function TCard.getMoney : integer;
begin
  getMoney := money;
end;

// Получить персональный код
function TCard.getPIN : string;
begin
  getPIN := pin;
end;

// Снять деньги
function TCard.withdraw( howmuch : integer ) : boolean;
begin
  if howmuch > money then
    withdraw := false
  else begin
    money := money - howmuch;
    withdraw := true;
  end;
end;
end;
```

```
// Конструктор
constructor TCard.Create;
begin
    init;
end;

end.
```

Заключение

Результатом проекта можно считать подтверждение целесообразности применения SWITCH-технологии для задачи построения системы управления техническими объектами, такими как банкомат, описанный в данной работе.

Для понимания логики программы можно рассматривать лишь графы переходов, поскольку текст программы при таком подходе является изоморфным им.

Также удалось повысить централизацию логики управления и отделить систему управления от ее визуализации.