Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

Кафедра «Компьютерные технологии»

С. В. Кессель

Разработка системы управления кофеваркой на основе автоматного подхода

Проектная документация

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

Санкт–Петербург
2003

Содержание

Введение

Данная работа выполнена в качестве курсовой по предмету «Теория автоматов и программирования», читаемому на кафедре «Компьютерные технологии» на факультете «Информационные технологии и программирование» СПбГУ ИТМО.

Для разработки систем логического управления, автором курса, А.А.Шалыто, была предложена SWITCH-технология, которую весьма целесообразно применять при разработке систем управления техническими объектами в реальном времени. Подробную информацию о SWITCH-технологии можно найти на сайте http://is.ifmo.ru.

Основными преимуществами указанной технологии являются высокая степень формализации процесса разработки, возможность отделения аппаратно-независимой части (как правило логики работы алгоритма) от аппаратно-зависимой и связанная с этим возможность централизация логики управления в программном коде. К достоинствам технологии относится также то, что код программы изоморфен графам переходов автоматов. Это дает возможность понимать логику работы программы (ее поведения) по графам переходов, не обращаясь к ее тексту. Кроме того, наличие графов переходов обеспечивает легкость выявления наиболее трудных в программах без явного выделения состояний ошибок – логических. Нахождение таких ошибок может быть еще больше упрощено за счет применения протоколирования работы программы в терминах автоматов.

В данной работе в качестве объекта управления выбрана кофеварка, а целью работы является разработка программного обеспечения ее системы управления.

Особенностью данного примера использования SWITCH-технологии является то, что при разработке системы применяются автоматы, взаимодействующие за счет обмена номерами состояний. Используются также вложенные автоматы. Другой особенностью проекта является то, что применяемые в нем классы являются «оболочками» соответствующих автоматов. Поэтому в одном классе реализовано не более одного автомата. Использование нескольких экземпляров одного и того же класса позволяет исключить дублирование кода для управления однотипными устройствами, такими, как клапана и нагреватели.

При разработке системы управления использован язык C++, а для создания модели кофеварки (ее визуализатора) - среда Borland C++ Builder 3.0.

1. Постановка задачи

Для реализации выбрана упрощенная модель кофеварки, позволяющая выбирать режим варки и необходимое количество чашек кофе (рис. 1).

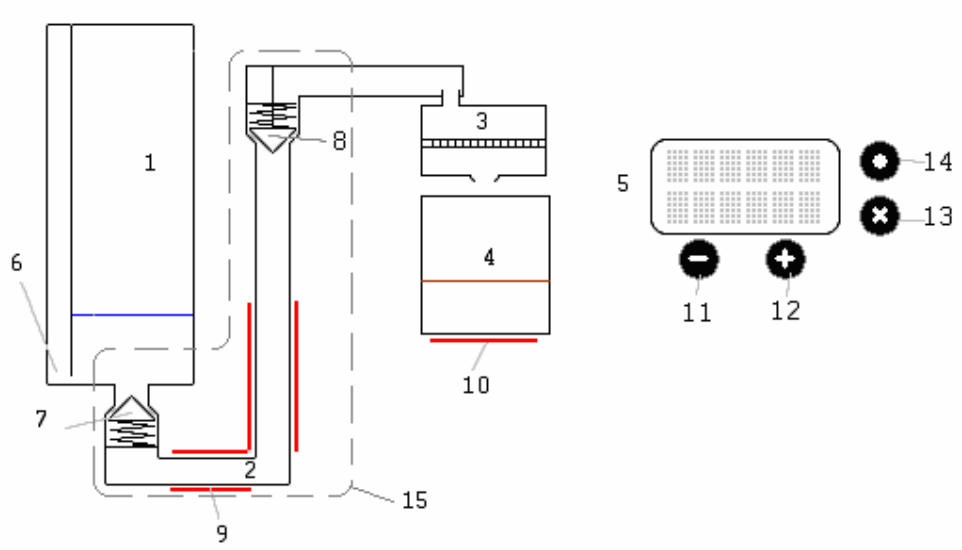

Рис. 1. Кофеварка

Перечислим основные элементы кофеварки:

- емкость для воды - 1;
- емкость для кипячения - 2;
- фильтр с кофе - 3;
- колба для готового напитка - 4;
- дисплей - 5;
- датчик воды - 6;
- входной клапан - 7 (клапан 1);
- выходной клапан - 8 (клапан 2);
- нагреватель кипячения воды - 9 (нагреватель 1);
- нагреватель подогрева готового кофе - 10 (нагреватель 2);
- кнопка «Увеличить параметр» (+) - 11;
- кнопка «Уменьшить параметр» (-) - 12;
- кнопка «Отмена» («Cancel», «C») - 13;
- кнопка «Подтвердить» («OK») – 14;
- клапаны 1 и 2, емкость для кипячения и нагреватель 1 образуют бойлер (15).

В кофеварке предусмотрена возможность индикации отсутствия воды и основных неисправностей (например, не работает один из нагревателей или неисправен клапан). Температура варки задается с точностью 5^{o}C в интервале от 50 до 150 ^{o}C. Количество чашек изменяется от 1 до 25.

На рис. 2 приведен визуализатор для моделирования работы кофеварки.

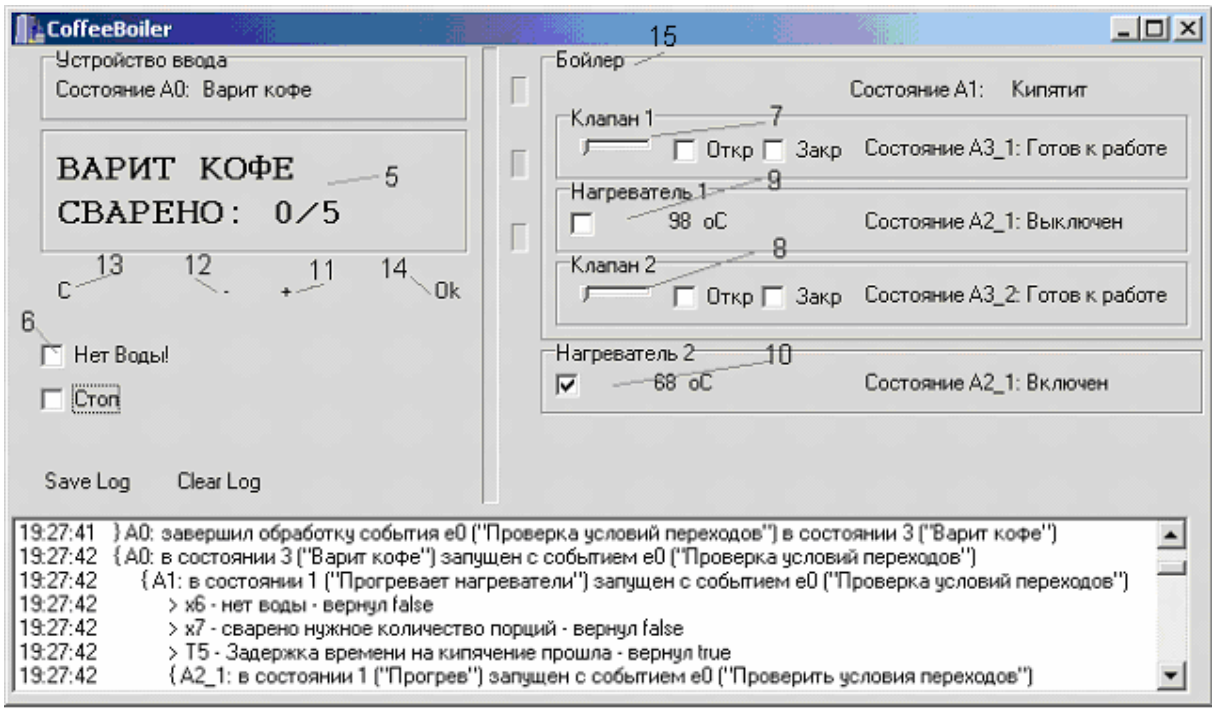

Рис. 2. Визуализатор кофеварки

В этом визуализаторе отсутствует ряд элементов из-за отсутствия динамики в них. К этим элементам относятся: емкость для воды (1), емкость для кипячения (2), фильтр с кофе (3) и колба для готового напитка (4).

2. Диаграмма классов

Для разработки визуализатора был использован автоматически сгенерированный средой программирования Borland C++ Builder 3.0 класс «Визуализатор кофеварки» (*TCoffeeBoiler*), порожденный от стандартного класса *TForm*. Основной функцией этого класса является предоставления интерфейса для визуализации. В класс *TCoffeeBoiler* включены в качестве членов экземпляры классов «Модель Кофеварки» (*TCoffeeModel*) и «Устройство управления кофеваркой» (*TCoffeeControl*). Первый из них является эмулятором кофеварки, а второй – собственно реализацией программы управления кофеваркой. В класс *TCoffeeControl* включен экземпляр класса «Бойлер» (*TBoiler*), осуществляющий управление бойлером. В класса *TBoiler* включены по два экземпляра классов «Клапан» (*TClapan*) и «Нагреватель» (*THeater*), осуществляющих соответственно управление клапанами и нагревателями.
Вызов автоматных процедур производится по таймеру *Timer1* из класса *TcoffeeBoiler*.

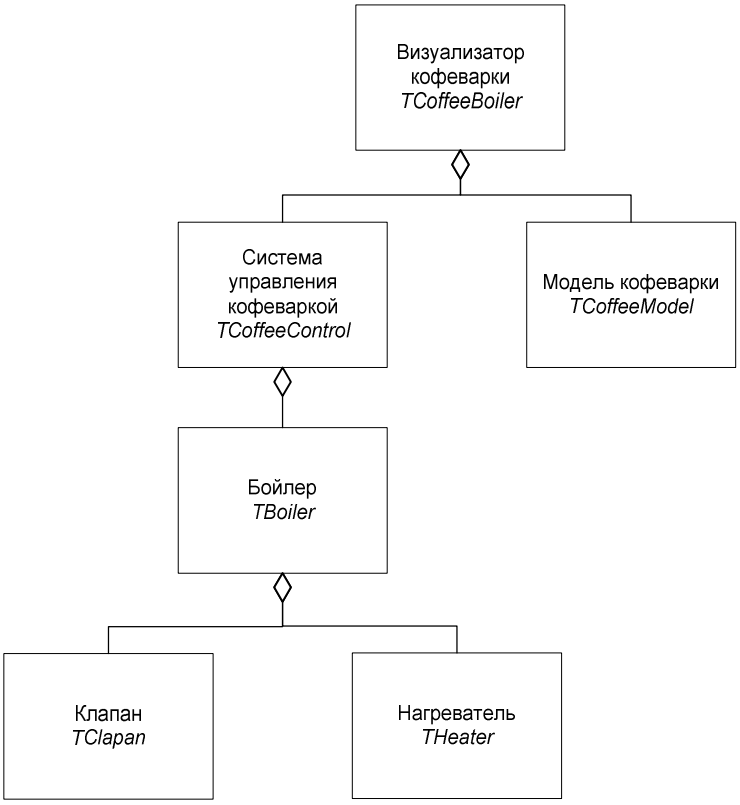

Рис. 3. Диаграмма классов

3. Класс «Устройство управления кофеваркой» (*TCoffeeControl*)

3.1. Словесное описание

Класс обеспечивает управление **кофеваркой**. Он позволяет выбрать температуру приготовления кофе и количество чашек, прервать и запустить процесс варки. Он также обеспечивает возможность индикации текущего состояния кофеварки. Класс является оболочкой автомата A0. Он содержит в себе экземпляр класса *TBoiler*, который реализует управление процессом варки кофе.

Управление кофеваркой осуществляется четырьмя кнопками («Увеличить параметр», «Уменьшить параметр», «Отмена» и «Подтвердить»). Действие кнопок зависит от состояния автомата A0. Например, при выборе температуры варки кнопки «Увеличить параметр» и «Уменьшить параметр» изменяют температуру варки, а при выборе количества чашек – количество чашек для варки.

В каждом состоянии на экране отображается информация о текущем состоянии (первая строка дисплея (5)) и сопутствующая этому состоянию вспомогательная информация (вторая строка).

Для начала варки необходимо в режиме ожидания (дисплей выключен) нажать кнопку «Подтвердить». Кофеварка перейдет в режим выбора температуры варки. В этом режиме кнопки «Увеличить параметр» и «Уменьшить параметр» соответственно увеличивают и уменьшают температуру, кнопка «Отмена» переводит кофеварку в режим ожидания, а кнопка «Подтвердить» - в режим выбора количества чашек кофе.

В режиме выбора количества чашек кофе кнопки «Увеличить параметр» и «Уменьшить параметр» соответственно увеличивают и уменьшают количество чашек, которые будут сварены. Кнопка «Отмена» переводит кофеварку в режим выбора температуры, а кнопка «Подтвердить» запускает процесс варки.

При варке на дисплее отображается текущее состояние («Прогрев» - кофеварка прогревает нагреватели, «Варка» - варится кофе и «Нет воды» - при отсутствии воды в емкости (1)). Во второй строке дисплея (5) отображается дополнительная информация (количество сваренных чашек кофе).

Нажатием кнопки «Отмена» варка кофе прерывается, и кофеварка переходит в режим индикации прерывания пользователем, из которого повторным нажатием кнопки «Отмена» она может быть переведена в режим ожидания.

3.2. Структурная схема класса

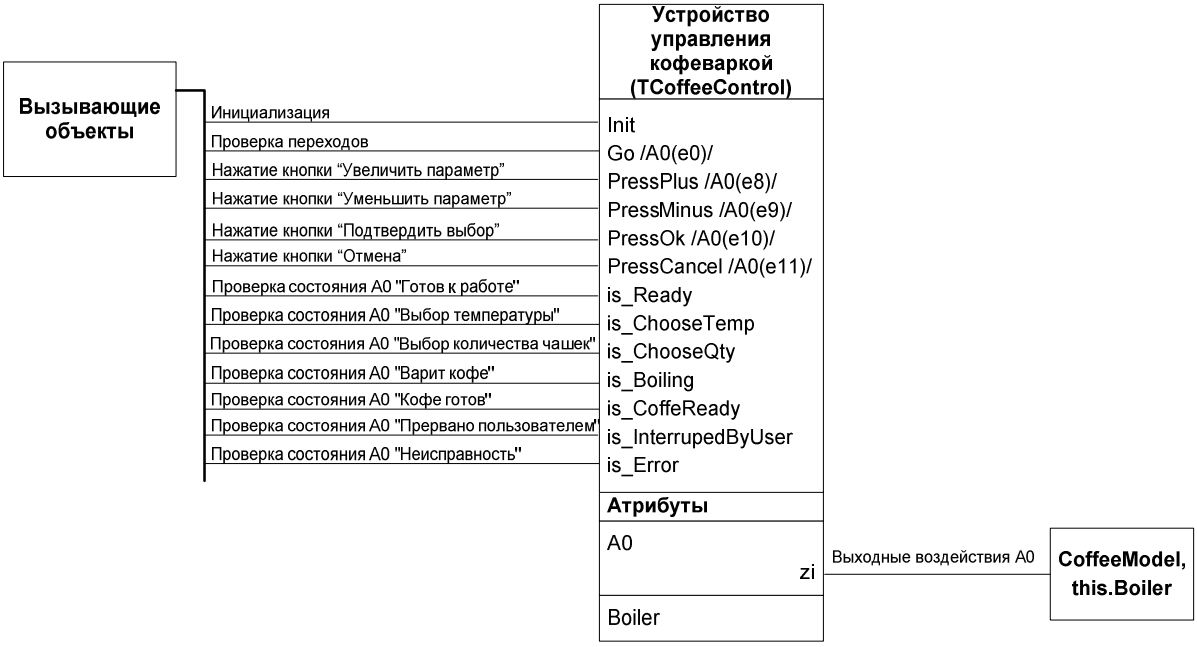

Рис. 4. Структурная схема класса «Устройство управления кофеваркой»

3.3. Автомат управления кофеваркой (A0)

3.3.1. Словесное описание

Автомат реализует управление кофеваркой, как указано в разд. 3.1. Он управляет устройством ввода-вывода, передавая управление процессом варки взаимодействующему с ним автомату управления бойлером (A1), который расположен в экземпляре класса *TBoiler*, входящего в класс *TCoffeeControl*. Передача управления производится вызовом автомата A1 с определенным событием из автомата A0. Кроме того, автомат A0 следит за состоянием автомата A1.

3.3.2. Схема связей

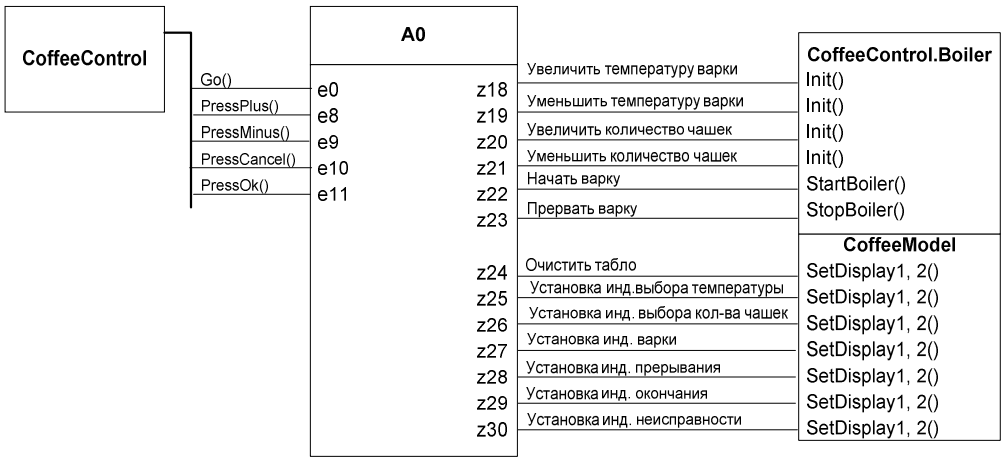

Рис. 5. Схема связей автомата управления кофеваркой

3.3.3. Граф переходов

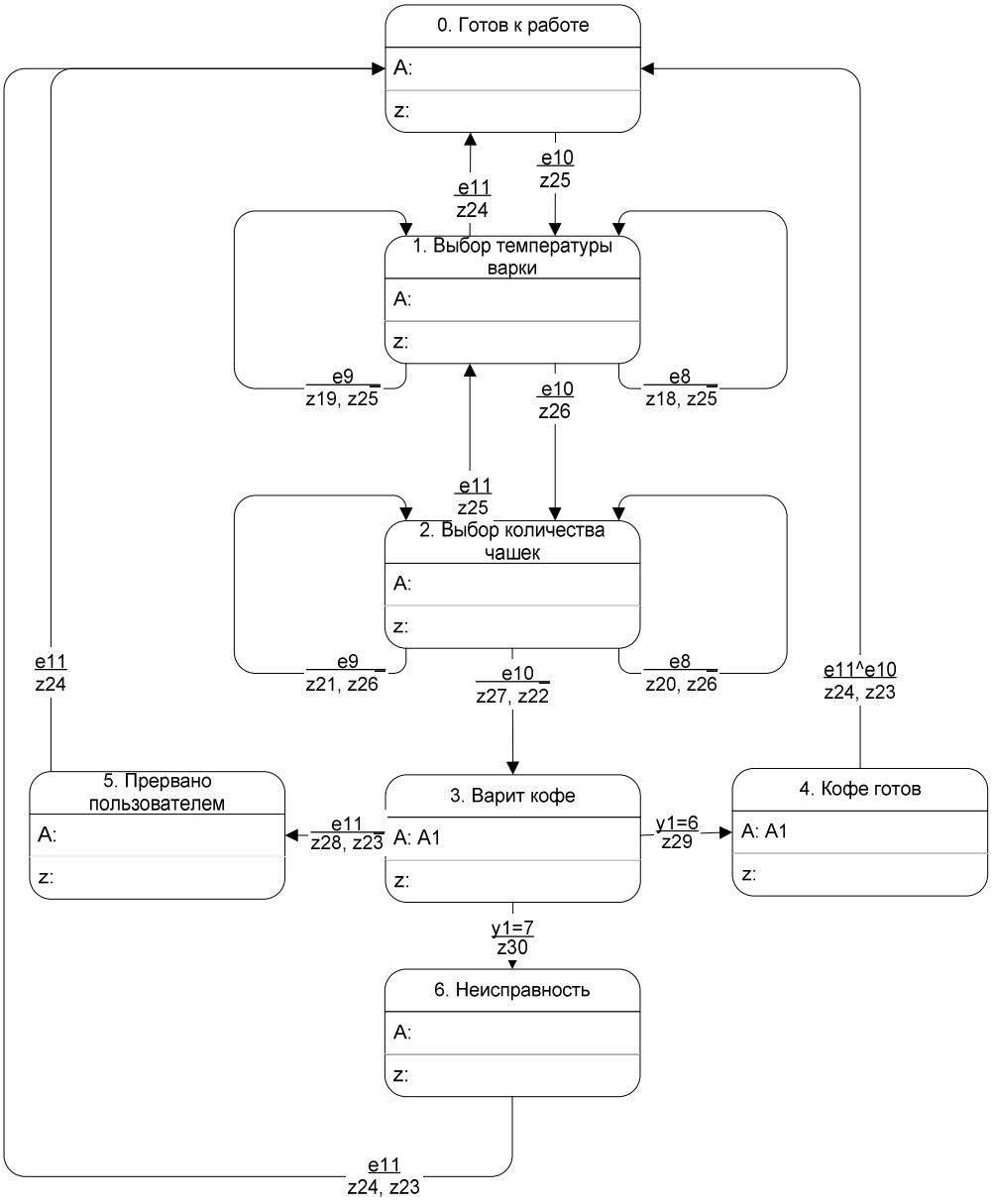

Рис. 6. Граф переходов автомата управления кофеваркой

4. Класс «Бойлер» (*TBoiler*)

4.1. Словесное описание

Класс является оболочкой автомата управления бойлером. Он управляет **процессом варки кофе**. Оно варится порциями. Считается, что в каждой чашке N порций (для большей наглядности N = 5). Таким образом, при варке M чашек кофе варится M*N порций.

В процессе варки контролируются работоспособность клапанов, нагревателей, а также проверяется наличие воды. В случае неисправности одного из клапанов или одного из нагревателей варка прерывается и высвечивается соответствующее предупреждение. В случае отсутствия воды в емкости варка приостанавливается, пока вода не будет долита либо пользователь не прервет варку.

Класс содержит два экземпляра класса *THeater* (соответственно нагреватель 1 и нагреватель 2) и два экземпляра класса *TClapan* **(**соответственно клапан 1 (входной) и клапан 2 (выходной)**)**.

4.2. Структурная схема класса

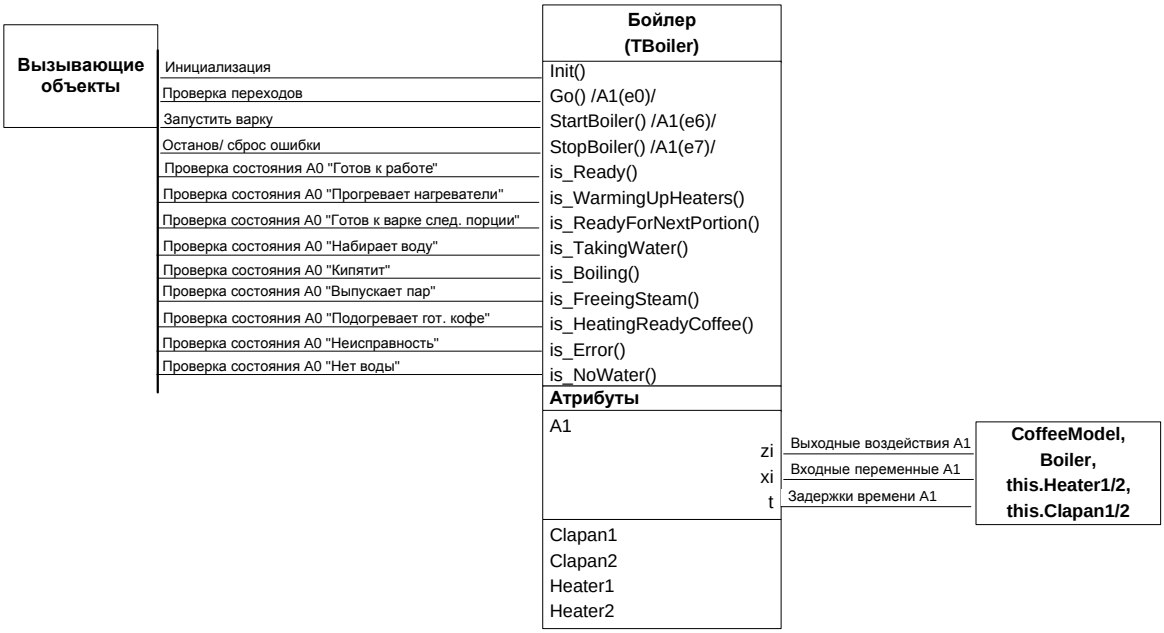

Рис. 7. Структурная схема класса «Бойлер»

4.3. Автомат управления бойлером (A1)

4.3.1. Словесное описание

Автомат реализует логику управления процессом варки кофе в соответствии с описанием, приведенным в разд. 4.1. Автомат взаимодействует с автоматом управления устройством ввода A0, передавая последнему текущее состояние и получая от него события.
Автомат также взаимодействует с автоматами A3_1, A3_2 (соответственно управление нагревателями 1 и 2), A2_1 и A2_2 (управление клапанами 1 и 2), получая их состояния и посылая им события.

4.3.2. Схема связей

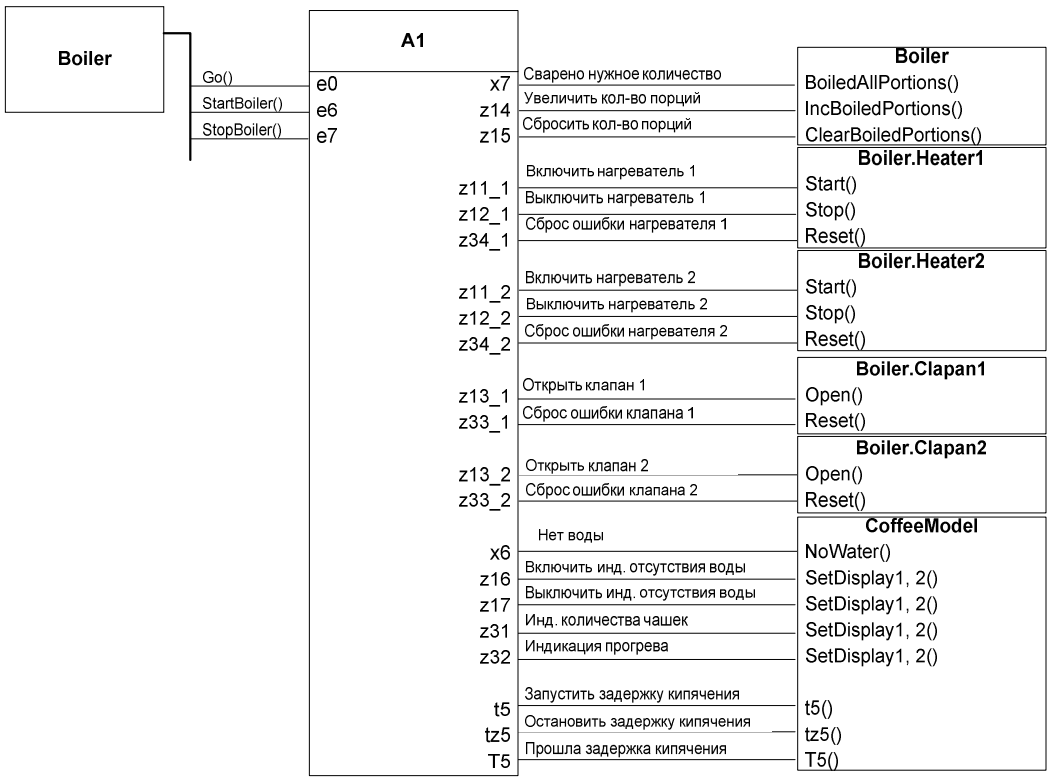

Рис. 8. Схема связей автомата управления бойлером

4.3.3. Граф переходов

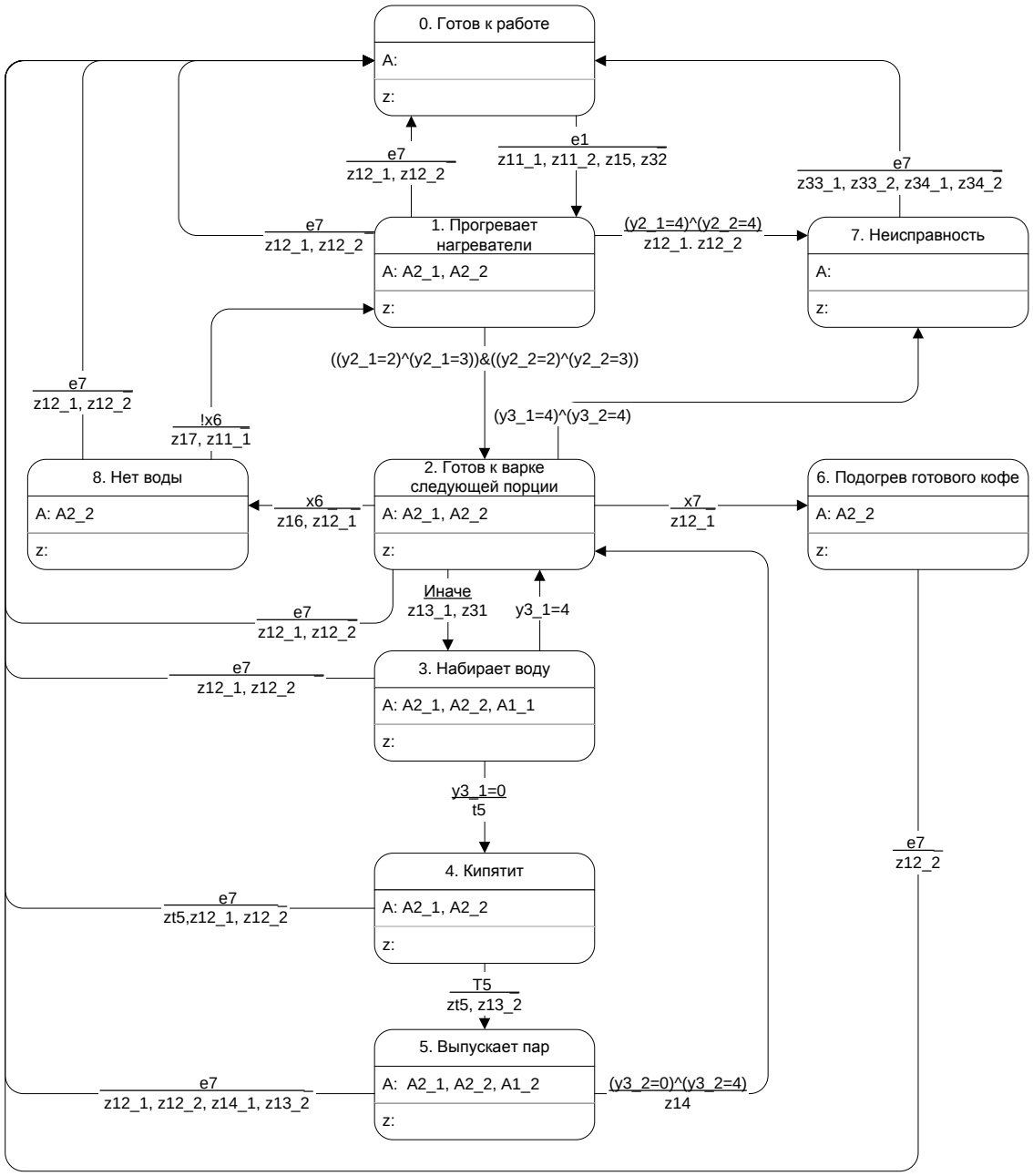

Рис. 9. Граф переходов автомата управления бойлером Вложенные АВТОМАТЫ

5. Класс «Нагреватель» (*THeater*)

5.1. Словесное описание

Класс является оболочкой автомата управления нагревателем. Температура нагревателя поддерживается в заранее заданном диапазоне. Предусмотрена возможность установления такой неисправности, как нагреватель не работает.

После включения нагревателя происходит его прогрев до нижней границы рабочего диапазона. Если он не прогрелся до данной температуры за заданное время, то нагреватель считается неисправным. Нагреватель включается при достижении температурой нижней границы рабочего интервала и выключается при достижении верхней границы.

5.2. Структурная схема класса

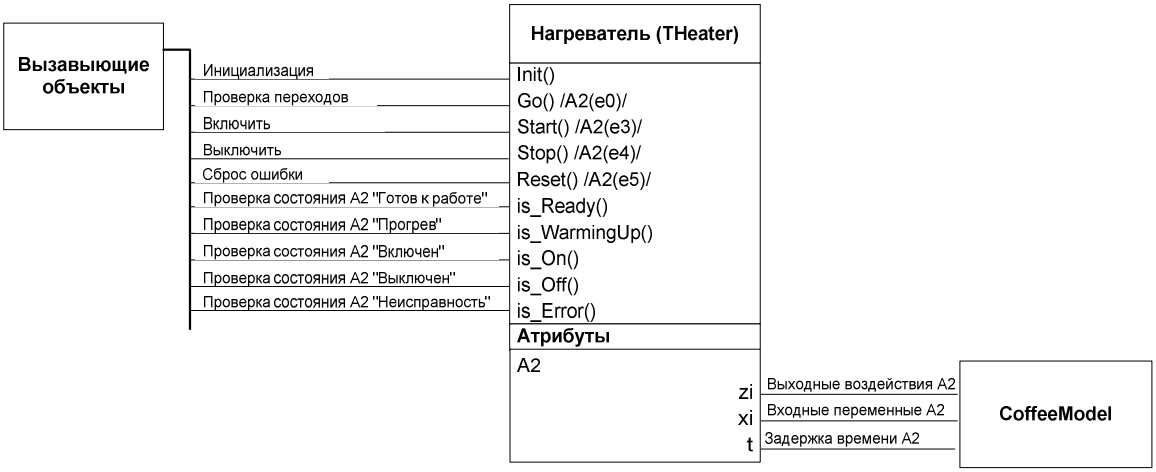

Рис. 10. Структурная схема класса «Нагреватель»

5.3. Автоматы управления нагревателями A2_1 и A2_2

5.3.1. Словесное описание

Автоматы A2_1 и A2_2 полностью идентичны и находятся в двух экземплярах одного класса *THeater*. В дальнейшем обозначение «автомат A2» относится как к автомату A2_1, так и к автомату A2_2. Этот автомат управляет работой нагревательного элемента. Он поддерживает температуру нагревателя в заданном диапазоне (задается минимальная и максимальная температуры). Предусмотрена возможность установления неисправности вида: нагреватель не работает. Автомат взаимодействует с автоматом управления бойлером A1, передавая последнему свое состояние и получая от него события.

5.3.2. Схема связей

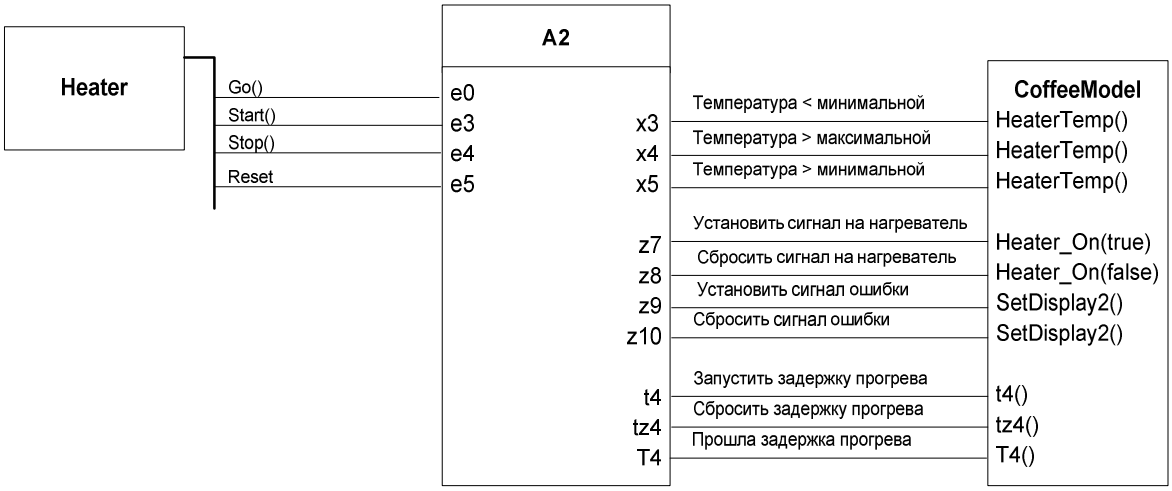

Рис. 11. Схема связей автомата управления нагревателем

5.3.3. Граф переходов

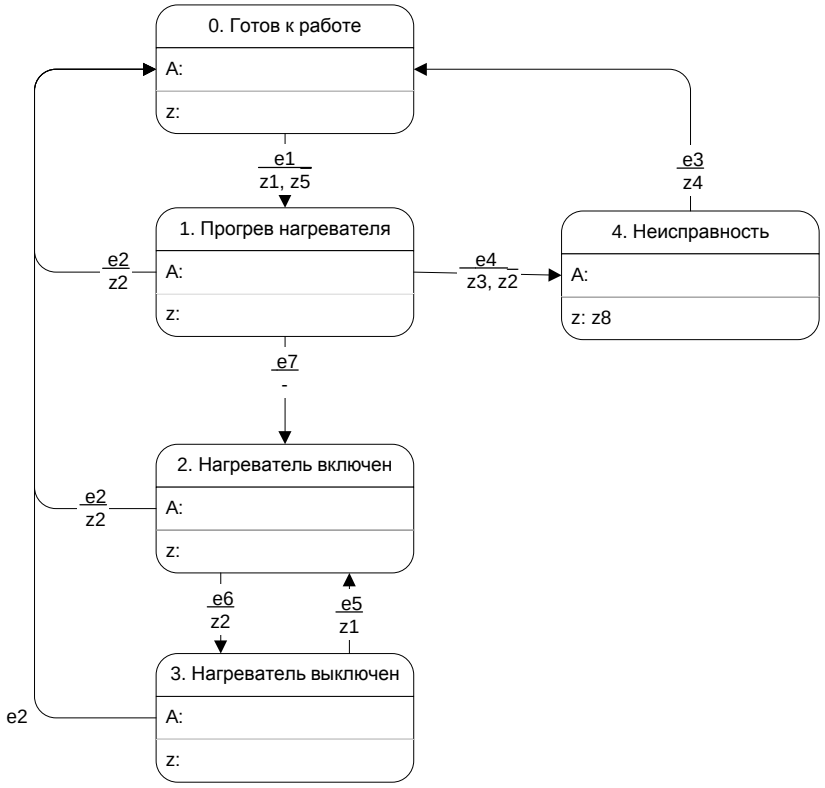

Рис. 12. Граф переходов автомата управления нагревателем

6. Класс «Клапан» (*TClapan*)

6.1. Словесное описание

Класс является оболочкой автомата управления клапаном и реализует функции управления работой клапана. В случает западания клапана (его неоткрытие либо незакрытие в течение заданного времени) клапан считается неработоспособным. Запуск клапана происходит по схеме «открыть-пауза-закрыть».

6.2. Структурная схема класса

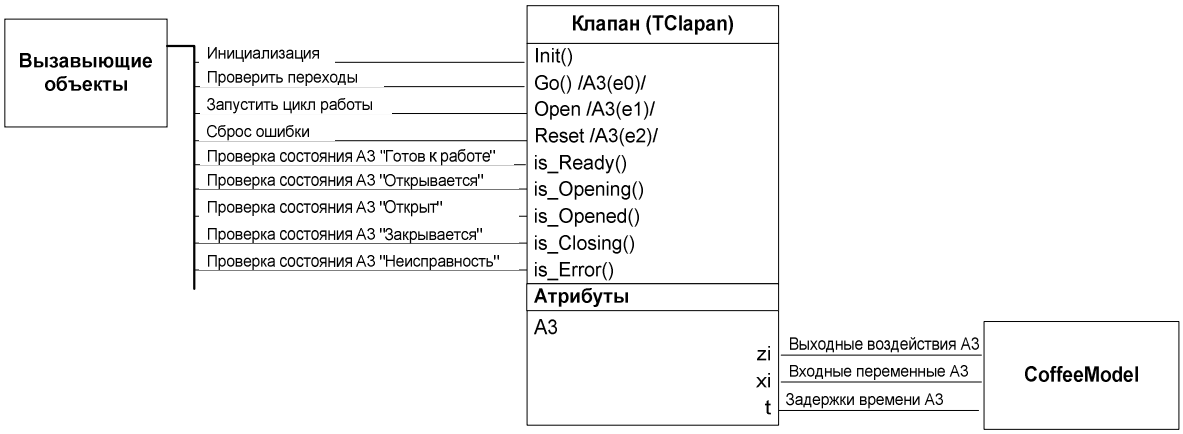

Рис. 13. Структурная схема класса «Клапан»

6.3. Автоматы управления клапанами A3_1 и A3_2:

6.3.1. Словесное описание

Автоматы A3_1 и A3_2 полностью идентичны и являются двумя экземплярами одного класса. В дальнейшем мы будем говорить «автомат A3», имея в виду, что сказанное относится и к автомату A3_1 и к автомату A3_2. Автомат A3 служит для реализации логики управления работой клапана в режиме «открыть-пауза-закрыть». Автомат имеет четыре состояния и взаимодействует с автоматом управления бойлером A1, передавая последнему свое состояние и получая от него события.

Автомат начинает цикл своей работы по событию e1 («Открыть клапан»). Он работает по схеме «открыть-пауза-закрыть». Сначала клапан открывается, затем делается пауза для набора воды или выпуска пара, а после этого клапан закрывается. В случае, если за определенное время клапан не откроется (либо не закроется), он считается неисправным.

6.3.2. Схема связей

Рис. 14. Схема связей автомата управления клапаном

6.3.3. Граф переходов

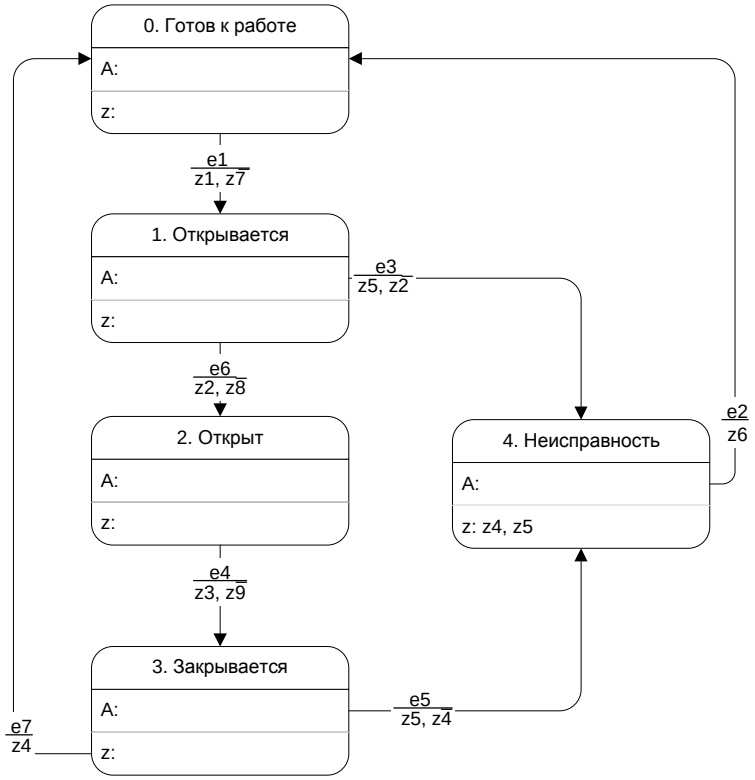

Рис. 15. Граф переходов автомата управления клапаном

8. Класс «Модель кофеварки» (*TCoffeeModel*)

8.1. Словесное описание

В этом классе реализована модель кофеварки, позволяющая тестировать работу программы, управляющей кофеваркой. В классе заданы все входные и выходные переменные других автоматов, а также задержки времени.

8.2. Структурная схема класса

Рис. 16. Структурная схема класса *TCoffeeModel*

9. Класс «Визуализатор кофеварки» (*TCoffeeBoiler*)

9.1. Словесное описание

Указанный класс автоматически сгенерирован средой Borland C++ Builder на основании визуально разработанной формы. В данный класс вручную добавлены экземпляры классов *TCoffeeControl* и *TCoffeeModel*. В свете того, что описываемый класс является частично автоматически сгенерированным и выполняет лишь служебную функцию, более подробно на нем останавливаться не будем.

10. Текст программы

10.1. Класс «Устройство управления кофеваркой»

```
//Объявление класса
class TCoffeeControl
{
    public:
        //Инициализация
        void Init(TCoffeeModel *NewCoffeeModel);
        //Вызовы автомата
        //Вызов автомата A0 с событием e0 (Проверка переходов)
        void Go();
        //Вызов автомата A0 с событием e8 (нажатие +)
        void PressPlus();
        //Вызов автомата A0 с событием e9 (нажатие -)
        void PressMinus();
        //Вызов автомата A0 с событием e10 (нажатие OK)
        void PressOk();
        //Вызов автомата A0 с событием e11 (нажатие CANCEL)
        void PressCancel();

        //Проверка состояний
        //Проверка состояния "Готов к работе"
        bool is_Ready();
        //Проверка состояния "Выбор температуры"
        bool is_ChooseTemp();
        //Проверка состояния "Выбор количества чашек"
        bool is_ChooseQty();
        //Проверка состояния "Варит кофе"
        bool is_Boiling();
        //Проверка состояния "Кофе готов"
        bool is_CoffeReady();
        //Проверка состояния "Прервано пользователем"
        bool is_InterrupedByUser();
        //Проверка состояния "Неисправность"
        bool is_Error();

        //Бойлер. public для доступа к состояниям
        TBoiler Boiler;
    private:
        //Заданная температура
        int Temperature;
        //Число порций (чашки)
        int NeededCups;
        //Ссылка на модель кофеварки
        TCoffeeModel *CoffeeModel;

        //Реализация автомата
        //Переменная, кодирующая состояние автомата A0: 0 - "Готов к работе",
        //1 - "Выбор температуры", 2 - "Выбор числа чашек", 3 - "Варит кофе",
        //4 - "Кофе готов", 5 - "Прервано пользователем", 6 - "Неисправность"

        int y0;

        //Выходные воздействия
        //Увеличить температуру варки
        void z18();
        //Уменьшить температуру варки
        void z19();
        //Увеличить количество чашек
```

```
        void z20();
        //Уменьшить количество чашек
        void z21();
        //Начать варку
        void z22();
        //Прервать варку
        void z23();
        //Очистить табло
        void z24();
        //Установка индикации выбора температуры
        void z25();
        //Установка индикации выбора количества чашек
        void z26();
        //Установка индикации варки
        void z27();
        //Установка индикации прерывания пользователя
        void z28();
        //Установка индикации окончания варки
        void z29();
        //Установка индикации неисправности
        void z30();

        //Процедура автомата
        void A0(int e);
};

//Инициализация
void TCoffeeControl::Init(TCoffeeModel *NewCoffeeModel)
{
    CoffeeModel = NewCoffeeModel;
    Temperature = 100;
    NeededCups = 5;
    Boiler.Init(Temperature - 5, Temperature + 5, NeededCups, CoffeeModel);
    y0 = 0;
}
//Вызовы автомата A0
//Вызов автомата A0 с событием e0 (Проверка переходов)
void TCoffeeControl::Go()
{
    if (y0 == 0) return;
    A0(0);
}
//Вызов автомата A0 с событием e8 (нажатие +)
void TCoffeeControl::PressPlus()
{
    A0(8);
}
//Вызов автомата A0 с событием e9 (нажатие -)
void TCoffeeControl::PressMinus()
{
    A0(9);
}
//Вызов автомата A0 с событием e10 (нажатие OK)
void TCoffeeControl::PressOk()
{
    A0(10);
}
//Вызов автомата A0 с событием e11 (нажатие CANCEL)
void TCoffeeControl::PressCancel()
{
    A0(11);
}
```

```
//Проверка состояний автомата A0
//Проверка состояния "Готов к работе"
bool TCoffeeControl::is_Ready()
{
    return y0 == 0;
}

//Проверка состояния "Выбор температуры"
bool TCoffeeControl::is_ChooseTemp()
{
    return y0 == 1;
}
//Проверка состояния "Выбор количества чашек"
bool TCoffeeControl::is_ChooseQty()
{
    return y0 == 2;
}
//Проверка состояния "Варит кофе"
bool TCoffeeControl::is_Boiling()
{
    return y0 == 3;
}
//Проверка состояния "Кофе готов"
bool TCoffeeControl::is_CoffeReady()
{
    return y0 == 4;
}
//Проверка состояния "Прервано пользователем"
bool TCoffeeControl::is_InterrupedByUser()
{
    return y0 == 5;
}
//Проверка состояния "Неисправность"
bool TCoffeeControl::is_Error()
{
    return y0 == 6;
}

//Выходные воздействия автомата A0
//Увеличить температуру варки
void TCoffeeControl::z18()
    {
    AppendToLog("*", "z18 - увеличить температуру варки");
    if (Temperature < 150 ) Temperature += 5;
    Boiler.Init(Temperature - 5, Temperature + 5, NeededCups, CoffeeModel);
    }
//Уменьшить температуру варки
void TCoffeeControl::z19()
    {
    AppendToLog("*", "z19 - уменьшить температуру варки");
    if (Temperature > 50 ) Temperature -= 5;
    Boiler.Init(Temperature - 5, Temperature + 5, NeededCups, CoffeeModel);
    }
//Увеличить количество чашек
void TCoffeeControl::z20()
    {
    AppendToLog("*", "z20 - увеличить количество чашек");
    if (NeededCups < 25  ) NeededCups++;
    Boiler.Init(Temperature - 5, Temperature + 5, NeededCups, CoffeeModel);
    }
//Уменьшить количество чашек
void TCoffeeControl::z21()
    {
    AppendToLog("*", "z21 - уменьшить количество чашек");
```

```
    if (NeededCups > 1  ) NeededCups--;
    Boiler.Init(Temperature - 5, Temperature + 5, NeededCups, CoffeeModel);
    }
//Начать варку
void TCoffeeControl::z22()
    {
    AppendToLog("*", "z22 - начать варку");
    Boiler.StartBoiler();
    }
//Прервать варку
void TCoffeeControl::z23()
    {
    AppendToLog("*", "z23 - прервать варку");
    Boiler.StopBoiler();
    }
//Очистить табло
void TCoffeeControl::z24()
    {
    AppendToLog("*", "z24 - очистить табло");
    CoffeeModel->SetDisplay1("");
    CoffeeModel->SetDisplay2("");
    }
//Установка индикации выбора температуры
void TCoffeeControl::z25()
    {
    AppendToLog("*", "z25 - установить индикацию выбора температуры");
    CoffeeModel->SetDisplay1("ТЕМПЕРАТУРА");
    CoffeeModel->SetDisplay2(IntToStr(Temperature));
    }
//Установка индикации выбора количества чашек
void TCoffeeControl::z26()
    {
    AppendToLog("*", "z26 - установить индикацию выбора количества чашек");
    CoffeeModel->SetDisplay1("ЧАШКИ");
    CoffeeModel->SetDisplay2(IntToStr(NeededCups));
    }
//Установка индикации варки
void TCoffeeControl::z27()
    {
    AppendToLog("*", "z26 - установить индикацию варки");
    CoffeeModel->SetDisplay1("ВАРИТ КОФЕ");
    CoffeeModel->SetDisplay2("");
    }
//Установка индикации прерывания пользователя
void TCoffeeControl::z28()
    {
    AppendToLog("*", "z26 - установить индикацию прерывания пользователем");
    CoffeeModel->SetDisplay1("ПРЕРВАНО");
    CoffeeModel->SetDisplay2("");
    }
//Установка индикации окончания варки
void TCoffeeControl::z29()
    {
    AppendToLog("*", "z29 - установить индикацию окончания варки");
    CoffeeModel->SetDisplay1("КОФЕ ГОТОВ!");
    CoffeeModel->SetDisplay2("");
    }
//Установка индикации неисправности
void TCoffeeControl::z30()
    {
    AppendToLog("*", "z26 - установить индикацию неисправности");
    CoffeeModel->SetDisplay1("НЕИСПРАВНОСТЬ");
    }
```

```
//Процедура автомата A0
void TCoffeeControl::A0(int e)
 {
    int old_y0;
    AnsiString StateDescr;
    AnsiString NewStateDescr;
    AnsiString EventDescr;
    old_y0 = y0;

    //Запись в лог

    switch (y0)
        {
        case 0:
            StateDescr = " (\"Готов к работе\")";
        break;

        case 1:
            StateDescr = " (\"Выбор температуры\")";
        break;

        case 2:
            StateDescr = " (\"Выбор количества чашек\")";
        break;

        case 3:
            StateDescr = " (\"Варит кофе\")";
        break;

        case 4:
            StateDescr = " (\"Кофе готов\")";
        break;

        case 5:
            StateDescr = " (\"Прервано пользователем\")";
        break;

        case 6:
            StateDescr = " (\"Неисправность\")";
        break;

        default:
            StateDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }

    switch (e)
        {
        case 0:
            EventDescr = " (\"Проверка условий переходов\")";
        break;

        case 8:
            EventDescr = " (\"Нажатие '+'\")";
        break;

        case 9:
            EventDescr = " (\"Нажатие '-'\")";
        break;

        case 10:
            EventDescr = " (\"Нажатие 'OK'\")";
        break;
```

```
        case 11:
            EventDescr = " (\"Нажатие 'CANCEL'\")";
        break;

        default:
            EventDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;

        }
    AppendToLog("{", "A0: в состоянии " + IntToStr(y0) + StateDescr + " запущен
с событием e" + IntToStr(e) + EventDescr);

    switch (y0)
    {
        case 0:
            if (e == 10)   {z25();                              y0 = 1; }
        break;

        case 1:
            if (e == 10)   {z26();                              y0 = 2; }
            else
            if (e == 11)   {z24();                              y0 = 0; }
            else
            if (e == 8)    {z18();z25();                        y0 = 1; }
            else
            if (e == 9)    {z19();z25();                        y0 = 1; }
        break;

        case 2:
            if (e == 10)   {z27();z22();                        y0 = 3; }
            else
            if (e == 11)   {z25();                              y0 = 1; }
            else
            if (e == 8)    {z20();z26();                        y0 = 2; }
            else
            if (e == 9)    {z21();z26();                        y0 = 2; }
        break;

        case 3:
            if (Boiler.is_HeatingReadyCoffee())   {z29();       y0 = 4; }
            else
            if (Boiler.is_Error()) {z30();                      y0 = 6; }
            else
            if (e == 11)    {z28();z23();                       y0 = 5; }
        break;


        case 4:
            if ((e == 11)||(e==10))  {z24();z23();              y0 = 0; }
        break;

        case 5:
            if (e == 11)  {z24();                               y0 = 0; }
        break;

        case 6:
            if (e == 11)  {z24(); z23();                        y0 = 0; }
        break;

    }
```

```
//Вложенные автоматы
switch (y0)
{
    case 0:
    break;

    case 1:
    break;

    case 2:
    break;

    case 3:
        Boiler.Go();
    break;

    case 4:
        Boiler.Go();
    break;

    case 5:
    break;

    case 6:
    break;
}
//Запись в лог
switch (y0)
    {
    case 0:
        NewStateDescr = " (\"Готов к работе\")";
    break;

    case 1:
        NewStateDescr = " (\"Выбор температуры\")";
    break;

    case 2:
        NewStateDescr = " (\"Выбор количества чашек\")";
    break;

    case 3:
        NewStateDescr = " (\"Варит кофе\")";
    break;

    case 4:
        NewStateDescr = " (\"Кофе готов\")";
    break;

    case 5:
        NewStateDescr = " (\"Прервано пользователем\")";
    break;

    case 6:
        NewStateDescr = " (\"Неисправность\")";
    break;

    default:
        StateDescr = " !!!  ОШИБКА ЛОГА  !!!";
    break;
    }
```

```
    if (y0 != old_y0)
        {
        AppendToLog("T", "A0: перешел из состояния " + IntToStr(old_y0) +
StateDescr + " в состояние " + IntToStr(y0) + NewStateDescr);
        }
    AppendToLog("}", "A0: завершил обработку события e" + IntToStr(e) +
EventDescr + " в состоянии " + IntToStr(y0) + NewStateDescr);

}
```

10.2. Класс «Бойлер»

```
//Объявление класса
class TBoiler
{
    public:
        //Инициализация
        void Init(double MaxTemp, double MinTemp, int CupsNeeded, TCoffeeModel
*NewCoffeeModel);
        //Вызовы автомата A1
        //Вызов автомата A1 с событием e0 (Проверка переходов)
        void Go();
        //Вызов автомата A1 с событием e6 (начать варку кофе)
        void StartBoiler();
        //Вызов автомата A1 с событием e7 (прервать варку кофе/ сброс ошибки)
        void StopBoiler();

        //Проверка состояний автомата A1
        //Проверка состояния "Готов к работе"
        bool is_Ready();
        //Проверка состояния "Прогревает нагреватели"
        bool is_WarmingUpHeaters();
        //Проверка состояния "Готов к варке следующей порции"
        bool is_ReadyForNextPortion();
        //Проверка состояния "Набирает воду"
        bool is_TakingWater();
        //Проверка состояния "Кипятит"
        bool is_Boiling();
        //Проверка состояния "Выпускает пар"
        bool is_FreeingSteam();
        //Проверка состояния "Подогревает готовый кофе"
        bool is_HeatingReadyCoffee();
        //Проверка состояния "Неисправность"
        bool is_Error();
        //Проверка состояния "Нет воды"
        bool is_NoWater();

        //2 клапана и 2 нагревателя ниже public для того, чтобы можно было их
состояния узнавать
        //Клапан 1 (входной)
        TClapan Clapan1;
        //Клапан 2 (выходной)
        TClapan Clapan2;
        //Нагреватель 1 (основной)
        THeater Heater1;
        //Нагреватель 2 (подогрева колбы с готовым кофе)
        THeater Heater2;
    private:
        //Ссылка на модель кофеварки
        TCoffeeModel *CoffeeModel;

        //Общее к-во порций
        int PortionsNeeded;
        //Кол-во сваренных порций
        int PortionsBoiled;
        //Проверка - сварено нужное количество порций
        bool BoiledAllPortions();
        //Увеличить количество сваренных порций
        void IncBoiledPortions();
        //Сбросить количество сваренных порций
        void ClearBoiledPortions();
```

```
        //Переменная, кодирующая состояние автомата A1: 0 - "Готов к работе",
        //1 - "Прогревает нагреватели", 2 - "Готов к варке следующей порции",
        //3 - "Набирает воду", 4 - "Кипятит", 5 - "Выпускает пар",
        //6 - "Подогревает готовый кофе", 7 - "Неисправность", 8 - "Нет воды"
        int y1;

        //Входные переменные автомата A1
        //x6 - нет воды
        bool x6();
        //x7 - сварено нужное количество порций
        bool x7();

        //Задержки времени автомата A1
        //Срабатывание задержки на кипячение
        bool T5();
        //Сигнал старта задержки времени на кипячение
        void t5();
        //Сигнал сброса задержки времени на кипячение
        void tz5();

        //Выходные воздействия автомата A1
        //Включить первый нагреватель
        void z11_1();
        //Включить второй нагреватель
        void z11_2();
        //Выключить первый нагреватель
        void z12_1();
        //Выключить второй нагреватель
        void z12_2();
        //Открыть первый клапан
        void z13_1();
        //Открыть второй клапан
        void z13_2();
        //Увеличить количество сваренных порций
        void z14();
        //Сбросить количество сваренных порций
        void z15();
        //Включить инидикацию отсутствия воды
        void z16();
        //Выключить инидикацию отсутствия воды
        void z17();
        //Индикация количества чашек
        void z31();
        //Индикация прогрева
        void z32();
        //Сброс ошибки клапана 1
        void z33_1();
        //Сброс ошибки клапана 2
        void z33_2();
        //Сброс ошибки нагревателя 1
        void z34_1();
        //Сброс ошибки нагревателя 2
        void z34_2();

        //Процедура автомата A1
        void A1(int e);
};

//Инициализация
void TBoiler::Init (double MinTemp, double MaxTemp, int CupsNeeded, TCoffeeModel
*NewCoffeeModel)
    {
    CoffeeModel = NewCoffeeModel;
    Clapan1.Init(0, CoffeeModel);
```

```
    Clapan2.Init(1, CoffeeModel);
    Heater1.Init(0, MinTemp, MaxTemp, CoffeeModel);
    Heater2.Init(1, 55, 75, CoffeeModel);
    PortionsNeeded = CupsNeeded * NoPortionsInCup;
    y1 = 0;
    }
//Проверка - сварено нужное количество порций
bool TBoiler::BoiledAllPortions()
{
    return (PortionsBoiled >= PortionsNeeded);
}
//Увеличить количество сваренных порций
void TBoiler::IncBoiledPortions()
{
        PortionsBoiled++;
}
//Сбросить количество сваренных порций
void TBoiler::ClearBoiledPortions()
{
    PortionsBoiled = 0;
}

//Вызовы автомата A1
//Вызов автомата A1 с событием e0 (Проверка переходов)
void TBoiler::Go()
    {
    if (y1 == 0) return;
    A1(0);
    }
//Вызов автомата A1 с событием e6 (начать варку кофе)
void TBoiler::StartBoiler() //Открыть
    {
    A1(6);
    }
//Вызов автомата A1 с событием e7 (прервать варку кофе/ сброс ошибки)
void TBoiler::StopBoiler()//Сброс ошибки
    {
    A1(7);
    }
//Проверка состояний автомата A1
//Проверка состояния "Готов к работе"
bool TBoiler::is_Ready()     //Состояние - готов к работе?
    {
    return (y1 == 0);
    }
//Проверка состояния "Прогревает нагреватели"
bool TBoiler::is_WarmingUpHeaters()    //Состояние - прогревает нагреватели?
    {
    return (y1 == 1);
    }
//Проверка состояния "Готов к варке следующей порции"
bool TBoiler::is_ReadyForNextPortion() //Состояние - готов к варке следующей
порции?
    {
    return (y1 == 2);
    }
//Проверка состояния "Набирает воду"
bool TBoiler::is_TakingWater()    //Состояние - набирает воду?
    {
    return (y1 == 3);
    }
//Проверка состояния "Кипятит"
bool TBoiler::is_Boiling()     //Состояние - кипятит?
    {
```

```
    return (y1 == 4);
    }
//Проверка состояния "Выпускает пар"
bool TBoiler::is_FreeingSteam()     //Состояние - выпускает пар?
    {
    return (y1 == 5);
    }
//Проверка состояния "Подогревает готовый кофе"
bool TBoiler::is_HeatingReadyCoffee()     //Состояние - кипятит?
    {
    return (y1 == 6);
    }
//Проверка состояния "Неисправность"
bool TBoiler::is_Error()     //Состояние - неисправность?
    {
    return (y1 == 7);
    }
//Проверка состояния "Нет воды"
bool TBoiler::is_NoWater()   //Состояние - нет воды?
    {
    return (y1 == 8);
    }
//Входные переменные автомата A1
//x6 - нет воды
bool TBoiler::x6()  //Нет воды
    {
    bool result;
    AnsiString sResult;
    result = CoffeeModel->NoWater();
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "x6 - нет воды - вернул " + sResult );
    return result;
    }
//x7 - сварено нужное количество порций
bool TBoiler::x7()  //Сварено нужное количество порций
    {
    bool result;
    AnsiString sResult;
    result = BoiledAllPortions();
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "x7 - сварено нужное количество порций - вернул " + sResult
);
    return result;
    }

//Задержки времени автомата A1
//Срабатывание задержки на кипячение
bool TBoiler::T5()   //Задержка времени на кипячение прошла
    {
    bool result;
    AnsiString sResult;
    result = CoffeeModel->T5();
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "T5 - Задержка времени на кипячение прошла - вернул " +
sResult );
    return result;
    }
//Сигнал старта задержки времени на кипячение
void TBoiler::t5()   //Сигнал старта задержки времени на кипячение
    {
    AppendToLog("*", "t4 - установка задержки времени на кипячение");
    CoffeeModel->t5();
    }
```

```
//Сигнал сброса задержки времени на кипячение
void TBoiler::tz5()   //Сигнал сброса задержки времени на кипячение
    {
    AppendToLog("*", "tz4 - сброс задержки времени на кипячение");
    CoffeeModel->tz5();
    }

//Выходные воздействия автомата A1
//Включить первый нагреватель
void TBoiler::z11_1()  //Включить первый нагреватель
    {
    AppendToLog("*", "z11_1 - включить первый нагреватель");
    Heater1.Start();
    }
//Включить второй нагреватель
void TBoiler::z11_2()
    {
    AppendToLog("*", "z11_2 - включить второй нагреватель");
    Heater2.Start();
    }
//Выключить первый нагреватель
void TBoiler::z12_1()
    {
    AppendToLog("*", "z12_1 - выключить первый нагреватель");
    Heater1.Stop();
    }
//Выключить второй нагреватель
void TBoiler::z12_2()
    {
    AppendToLog("*", "z12_2 - выключить второй нагреватель");
    Heater2.Stop();
    }
//Открыть первый клапан
void TBoiler::z13_1()  //Открыть первый клапан
    {
    AppendToLog("*", "z13_1 - открыть первый клапан");
    Clapan1.Open();
    }
//Открыть второй клапан
void TBoiler::z13_2()  //Открыть второй клапан
    {
    AppendToLog("*", "z13_2 - открыть второй клапан");
    Clapan2.Open();
    }
//Увеличить количество сваренных порций
void TBoiler::z14()    //Увеличить количество сваренных порций
    {
    AppendToLog("*", "z14 - увеличить количество сваренных порций");
    IncBoiledPortions();
    }
//Сбросить количество сваренных порций
void TBoiler::z15()    //Сбросить количество сваренных порций
    {
    AppendToLog("*", "z15 - сбросить количество сваренных порций");
    ClearBoiledPortions();
    }
//Включить инидикацию отсутствия воды
void TBoiler::z16()    //Включить инидикацию отсутствия воды
    {
    AppendToLog("*", "z16 - включить инидикацию отсутствия воды");
    CoffeeModel->SetDisplay2("НЕТ ВОДЫ");
    }
```

```
//Выключить инидикацию отсутствия воды
void TBoiler::z17()    //Выключить инидикацию отсутствия воды
    {
    AppendToLog("*", "z17 - выключить инидикацию отсутствия воды");
    CoffeeModel->SetDisplay2("");
    }
//Индикация количества чашек
void TBoiler::z31()    //Индикация количества чашек
    {
    AppendToLog("*", "z17 - включить индикацию количества чашек");
    CoffeeModel->SetDisplay2("СВАРЕНО: " + IntToStr(PortionsBoiled /
NoPortionsInCup) + "/" + IntToStr(PortionsNeeded/ NoPortionsInCup));
    }
//Индикация прогрева
void TBoiler::z32()    //Индикация прогрева
    {
    AppendToLog("*", "z17 - включить индикацию прогрева");
    CoffeeModel->SetDisplay2("ПРОГРЕВ НАГР.");
    }
//Сброс ошибки клапана 1
void TBoiler::z33_1()    //Сброс ошибки клапана
    {
    AppendToLog("*", "z33 - сброс ошибки клапана1");
    Clapan1.Reset();
    }
//Сброс ошибки клапана 2
void TBoiler::z33_2()    //Сброс ошибки клапана
    {
    AppendToLog("*", "z33 - сброс ошибки клапана1");
    Clapan2.Reset();
    }
//Сброс ошибки нагревателя 1
void TBoiler::z34_1()    //Сброс ошибки нагревателя1
    {
    AppendToLog("*", "z34 - сброс ошибки нагревателя1");
    Heater1.Reset();
    }
//Сброс ошибки нагревателя 2
void TBoiler::z34_2()    //Сброс ошибки нагревателя1
    {
    AppendToLog("*", "z34 - сброс ошибки нагревателя1");
    Heater2.Reset();
    }

//Процедура автомата A1
void TBoiler::A1(int e)
{
    int old_y1;
    AnsiString StateDescr;
    AnsiString NewStateDescr;
    AnsiString EventDescr;
    bool bx6;
    bool bx7;
    bool bT5;

    old_y1 = y1;
    //Запись в лог
    switch (y1)
        {
        case 0:
            StateDescr = " (\"Готов к работе\")";
        break;

        case 1:
```

```
            StateDescr = " (\"Прогревает нагреватели\")";
        break;

        case 2:
            StateDescr = " (\"Готов к варке следующей порции\")";
        break;

        case 3:
            StateDescr = " (\"Набирает воду\")";
        break;

        case 4:
            StateDescr = " (\"Кипятит\")";
        break;

        case 5:
            StateDescr = " (\"Выпускает пар\")";
        break;

        case 6:
            StateDescr = " (\"Подогревает готовый кофе\")";
        break;

        case 7:
            StateDescr = " (\"Неисправность\")";
        break;

        case 8:
            StateDescr = " (\"Нет воды\")";
        break;

        default:
            StateDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }
    switch (e)
        {
        case 0:
            EventDescr = " (\"Проверка условий переходов\")";
        break;

        case 6:
            EventDescr = " (\"Начать варку кофе\")";
        break;

        case 7:
            EventDescr = " (\"Прервать варку кофе/ сброс ошибки\")";
        break;

        default:
            EventDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }
    AppendToLog("{", "A1: в состоянии " + IntToStr(y1) + StateDescr + " запущен
с событием e" + IntToStr(e) + EventDescr);

    bx6 = x6();  //Нет воды
    bx7 = x7();  //Сварено нужное количество порций
    bT5 = T5();  //Задержка времени на кипячение прошла
```

```
switch (y1)
{
    case 0:
        if (e == 6)   {z11_1();z11_2();z15();z32();         y1 = 1; }
    break;

    case 1:
        if (e == 7)   {z12_1();z12_2();                     y1 = 0;}
        if ((Heater1.is_On()||Heater1.is_Off())&&
                (Heater2.is_On()||Heater2.is_Off()))        y1 = 2;

        if (Heater1.is_Error()||Heater2.is_Error())
                {z12_1();z12_2();                           y1 = 7;}
    break;

    case 2:
        if (e == 7)   {z12_1();z12_2();                     y1 = 0;}
        if (Clapan1.is_Error()||Clapan2.is_Error()){        y1 = 7;}
        else
        if (bx6)   {z16();z12_1();                          y1 = 8;}
        else
        if (bx7)   {z12_1();                                y1 = 6;}
        else
            {z13_1();z31();                                 y1 = 3;}
    break;

    case 3:
        if (e == 7)   {z12_1();z12_2();                     y1 = 0;}
        else
        if (Clapan1.is_Error()) {                           y1 = 2;}
        else
        if (Clapan1.is_Ready()) { t5();                     y1 = 4;}
    break;

    case 4:
        if (e == 7)   {tz5();z12_1();z12_2();               y1 = 0;}
        else
        if (bT5) { tz5();z13_2();                           y1 = 5;}
    break;

    case 5:
        if (e == 7)   {z12_1();z12_2();                     y1 = 0;}
        else
        if (Clapan2.is_Ready()||Clapan2.is_Error()){z14();  y1 = 2;}
    break;

    case 6:
        if (e == 7)   {z12_2();                             y1 = 0;}
    break;

    case 7:
        if (e == 7)   {z33_1();z33_2();z34_1();z34_2();     y1 = 0;}
    break;

    case 8:
        if (e == 7)   {z12_1();z12_2();                     y1 = 0;}
        if (!bx6)   {z17();z11_1();                         y1 = 1;}
    break;
}
```

```
//"Вложенные" автоматы
switch (y1)
{
    case 0:
    break;

    case 1:
        Heater1.Go(); Heater2.Go();
    break;

    case 2:
        Heater1.Go(); Heater2.Go();
    break;

    case 3:
        Heater1.Go(); Heater2.Go(); Clapan1.Go();
    break;

    case 4:
        Heater1.Go(); Heater2.Go();
    break;

    case 5:
        Heater1.Go(); Heater2.Go(); Clapan2.Go();
    break;

    case 6:
        Heater2.Go();
    break;

    case 7:
    break;

    case 8:
        Heater2.Go();
    break;
}


switch (y1)
    {
    case 0:
        NewStateDescr = " (\"Готов к работе\")";
    break;

    case 1:
        NewStateDescr = " (\"Прогревает нагреватели\")";
    break;

    case 2:
        NewStateDescr = " (\"Готов к варке следующей порции\")";
    break;

    case 3:
        NewStateDescr = " (\"Набирает воду\")";
    break;

    case 4:
        NewStateDescr = " (\"Кипятит\")";
    break;

    case 5:
        NewStateDescr = " (\"Выпускает пар\")";
    break;
```

```
        case 6:
            NewStateDescr = " (\"Подогревает готовый кофе\")";
        break;

        case 7:
            NewStateDescr = " (\"Неисправность\")";
        break;

        case 8:
            NewStateDescr = " (\"Нет воды\")";
        break;

        default:
            StateDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }

    if (y1 != old_y1)
        {
        AppendToLog("T", "A1: перешел из состояния " + IntToStr(old_y1) +
StateDescr + " в состояние " + IntToStr(y1) + NewStateDescr);
        }
    AppendToLog("}", "A1: завершил обработку события e" + IntToStr(e) +
EventDescr + " в состоянии " + IntToStr(y1) + NewStateDescr);
}
```

10.3. Класс «Нагреватель»

```
//Объявление класса
class THeater
{
    public:
        //Инициализация
        void Init(int HeaterNo, double MinTemp, double MaxTemp, TCoffeeModel
*NewCoffeeModel);
        //Вызовы автомата A2
        //Вызов автомата A2 с событием e0 (Проверка условий переходов)
        void Go();
        //Вызов автомата A2 с событием e3 (Включение нагревателя)
        void Start();
        //Вызов автомата A2 с событием e4 (Выключение нагревателя)
        void Stop();
        //Вызов автомата A2 с событием e5 (Сброс ошибки нагревателя)
        void Reset();

        //Проверка состояния автомата A2
        //Проверка состояния "Готов к работе"
        bool is_Ready();
        //Проверка состояния "Прогрев"
        bool is_WarmingUp();
        //Проверка состояния "Включен"
        bool is_On();
        //Проверка состояния "Выключен"
        bool is_Off();
        //Проверка состояния "Неисправность"
        bool is_Error();
    private:
        //ССылка на модель кофеварки
        TCoffeeModel *CoffeeModel;
        //Идентификатор нагревателя 0 - нагреватель 1, 1 - нагреватель 2
        int HeaterID;
        //Минимальная и максимальная температуры
        double MinTemperature;
        double MaxTemperature;

        //Входные переменные автомата A2
        //Температура меньше заданной минимальной
        bool x3();
        //Температура больше заданной максимальной
        bool x4();
        //Температура больше заданной минимальной (нагреватель прогрелся)
        bool x5();

        //Выходные воздействия автомата A2
        //Установить сигнал "Нагреватель выключен"
        void z7();
        //сбросить сигнал "Нагреватель включен"
        void z8();
        //Установить сигнал "Ошибка нагревателя"
        void z9();
        //Сбросить сигнал "Ошибка нагревателя"
        void z10();

        //Задержки времени автомата A2
        //Срабатывание задержки на прогрев
        bool T4();
        //Сигнал старта задержки времени на открытие
        void t4();
        //Сигнал сброса задержки времени на закрытие
```

```
        void tz4();
        //Переменная, кодируещая состояние автомата A3
        //Состояния: 0 - "Готов к работе"; 1 - "Прогрев"; 2 - "Включен"; 3 -
        //"Выключен"; 4 - "Неисправность"
        int y2;

        //Процедура автомата A2
        void A2(int e);   //Процедура автомата: e1 - сигнал старта, e2 - сигнал
останова, e3 - Сброс ошибки
};

//Инициализация
void THeater::Init(int HeaterNo, double MinTemp, double MaxTemp, TCoffeeModel
*NewCoffeeModel)
{
    CoffeeModel = NewCoffeeModel;
    y2 = 0;
    HeaterID = HeaterNo;
    MinTemperature = MinTemp;
    MaxTemperature = MaxTemp;
}
//Вызовы автомата A2
//Вызов автомата A2 с событием e0 (Проверка условий переходов)
void THeater::Go()
{
    if (y2 == 0) return;
    A2(0);
}
//Вызов автомата A2 с событием e3 (Включение нагревателя)
void THeater::Start() //Включить
{
    A2(3);
}
//Вызов автомата A2 с событием e4 (Выключение нагревателя)
void THeater::Stop()//Выключить
{
    A2(4);
}
//Вызов автомата A2 с событием e5 (Сброс ошибки нагревателя)
void THeater::Reset()
{
    A2(5);
}
//Проверка состояния автомата A2
//Проверка состояния "Готов к работе"
bool THeater::is_Ready()
{
    return (y2 == 0);
}
//Проверка состояния "Прогрев"
bool THeater::is_WarmingUp()
{
    return (y2 == 1);
}
//Проверка состояния "Включен"
bool THeater::is_On()
{
    return (y2 == 2);
}
//Проверка состояния "Выключен"
bool THeater::is_Off()
{
    return (y2 == 3);
}
```

```
//Проверка состояния "Неисправность"
bool THeater::is_Error()
{
    return (y2 == 4);
}

//Входные переменные автомата A2
//Температура меньше заданной минимальной
bool THeater::x3()
{
    bool result;
    AnsiString sResult;
    result = (CoffeeModel->HeaterTemp(HeaterID) <= MinTemperature);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "x3 - температура нагревателя" + IntToStr(HeaterID+1) + "
меньше заданной минимальной - вернул " + sResult );
    return result;
}
//Температура больше заданной максимальной
bool THeater::x4()
{
    bool result;
    AnsiString sResult;
    result = (CoffeeModel->HeaterTemp(HeaterID) >= MaxTemperature);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "x4 - температура нагревателя" + IntToStr(HeaterID+1) + "
больше заданной максимальной - вернул " + sResult );
    return result;
}
//Температура больше заданной минимальной (нагреватель прогрелся)
bool THeater::x5()
{
    bool result;
    AnsiString sResult;
    result = (CoffeeModel->HeaterTemp(HeaterID) > MinTemperature);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "x5 - температура нагревателя" + IntToStr(HeaterID+1) + "
больше заданной минимальной - вернул " + sResult );
    return result;
}
//Выходные воздействия автомата A2
//Установить сигнал "Нагреватель выключен"
void THeater::z7()
{
    AppendToLog("*", "z1 - установить сигнал нагревателя" +
IntToStr(HeaterID+1));
    CoffeeModel->Heater_On(HeaterID, true);
}
//сбросить сигнал "Нагреватель включен"
void THeater::z8()
{
    AppendToLog("*", "z1 - сбросить сигнал нагревателя" + IntToStr(HeaterID+1));
    CoffeeModel->Heater_On(HeaterID, false);
    return;
}
//Установить сигнал "Ошибка нагревателя"
void THeater::z9()
{
    AppendToLog("*", "z1 - установить сигнал ошибки нагревателя" +
IntToStr(HeaterID+1));
    CoffeeModel->SetDisplay2("НЕИСПР. НАГРЕВАТЕЛЬ " + IntToStr(HeaterID+1));
}
```

```
//Сбросить сигнал "Ошибка нагревателя"
void THeater::z10()
{
    AppendToLog("*", "z1 - сбросить сигнал ошибки нагревателя" +
IntToStr(HeaterID+1));
    CoffeeModel->SetDisplay2("");
}
//Задержки времени
//Сигнал старта задержки времени на открытие
void THeater::t4()
{
    AppendToLog("*", "t4 - запуск задержки времени на прогрев нагревателя" +
IntToStr(HeaterID+1));
    CoffeeModel->t4(HeaterID);
}
//Сигнал сброса задержки времени на закрытие
void THeater::tz4()
{
    AppendToLog("*", "tz4 - сброс задержки времени на прогрев нагревателя" +
IntToStr(HeaterID+1));
    CoffeeModel->tz4(HeaterID);
}
//Срабатывание задержки на прогрев
bool THeater::T4()
{
    bool result;
    AnsiString sResult;
    result = CoffeeModel->T4(HeaterID);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "T4 - Задержка времени на прогрев нагревателя" +
IntToStr(HeaterID+1) + " прошла - вернул " + sResult );
    return result;
}
//Процедура автомата A2
void THeater::A2(int e)
{
    int old_y2;
    AnsiString StateDescr;
    AnsiString NewStateDescr;
    AnsiString EventDescr;
    bool bx3;
    bool bx4;
    bool bx5;
    bool bT4;

    old_y2 = y2;
    //Записываем в лог
    switch (y2)
        {
        case 0:
            StateDescr = " (\"Готов к работе\")";
        break;

        case 1:
            StateDescr = " (\"Прогрев\")";
        break;

        case 2:
            StateDescr = " (\"Включен\")";
        break;

        case 3:
            StateDescr = " (\"Выключен\")";
        break;
```

```
        case 4:
            StateDescr = " (\"Неисправность\")";
        break;

        default:
            StateDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }
    switch (e)
        {
        case 0:
            EventDescr = " (\"Проверить условия переходов\")";
        break;

        case 3:
            EventDescr = " (\"Включение нагревателя\")";
        break;

        case 4:
            EventDescr = " (\"Выключение нагревателя\")";
        break;

        case 5:
            EventDescr = " (\"Сброс ошибки нагревателя\")";
        break;

        default:
            EventDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }
    AppendToLog("{", "A2_"+ IntToStr(HeaterID+1)+": в состоянии " + IntToStr(y2)
+ StateDescr + " запущен с событием e" + IntToStr(e) + EventDescr);

    bx3 = x3();
    bx4 = x4();
    bx5 = x5();
    bT4 = T4();
    switch (y2)
    {
        case 0:
            if (e == 3)         {z7();t4();         y2 = 1; };
        break;

        case 1:
            if (e == 4)      {tz4();z8();           y2 = 0;}
            else
            if (bx5)         {tz4();                y2 = 2;}
            else
            if (!bx5 & bT4)  {tz4();z9();z8();      y2 = 4;};
        break;

        case 2:
            if (e == 4)         {z8();              y2 = 0;}
            else
            if (bx4)           {z8();               y2 = 3;};
        break;

        case 3:
            if (e == 4)         {                   y2 = 0;}
            else
            if (bx3)           {z7();               y2 = 2;};
        break;
```

```
        case 4:
            if (e == 5)         {z10();             y2 = 0;};
        break;
    }
    switch (y2)
        {
        case 4:
            z8();
        break;
        }

    //Запись в лог
    switch (y2)
        {
        case 0:
            NewStateDescr = " (\"Готов к работе\")";
        break;

        case 1:
            NewStateDescr = " (\"Прогрев\")";
        break;

        case 2:
            NewStateDescr = " (\"Включен\")";
        break;

        case 3:
            NewStateDescr = " (\"Выключен\")";
        break;

        case 4:
            NewStateDescr = " (\"Неисправность\")";
        break;

        default:
            StateDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }
    if (y2 != old_y2)
        {
        AppendToLog("T", "A2_"+ IntToStr(HeaterID+1)+": перешел из состояния " +
IntToStr(old_y2) + StateDescr + " в состояние " + IntToStr(y2) + NewStateDescr);
        }
    AppendToLog("}", "A2_"+ IntToStr(HeaterID+1)+": завершил обработку события
e" + IntToStr(e) + EventDescr + " в состоянии " + IntToStr(y2) + NewStateDescr);
}
```

10.4. Класс «Клапан»

```
//Объявление класса
class TClapan
{
    public:
        //Инициализация
        void Init(int ClapanNo, TCoffeeModel *NewCoffeeModel);
        //Вызовы автомата A3
        //Вызов автомата A3 с событием e0 (Проверка переходов)
        void Go();
        //Вызов автомата A3 с событием e1 (основной цикл работы клапана)
        void Open();
        //Вызов автомата A3 с событием e2 (сброс ошибки клапана)
        void Reset();
        //Проверка состояния автомата A3
        //Проверка состояния "Готов к работе"
        bool is_Ready();
        //Проверка состояния "Открывается"
        bool is_Opening();
        //Проверка состояния "Открыт"
        bool is_Opened();
        //Проверка состояния "Закрывается"
        bool is_Closing();
        //Проверка состояния "Неисправность"
        bool is_Error();
    private:
        //Указатель на модель кофеварки
        TCoffeeModel *CoffeeModel;
        //Номер клапана: 0 - клапан 1 (входной), 1 - клапан 2 (выходной)
        int ClapanID;

        //Входные переменные автомата A3
        //x1 - клапан открыт
        bool x1();
        //x2 - клапан закрыт
        bool x2();

        //Выходные воздействия автомата A3
        //Установить сигнал "Открыть"
        void z1();
        //Сбросить сигнал "Открыть"
        void z2();
        //Установить сигнал "Закрыть"
        void z3();
        //Сбросить сигнал "Закрыть"
        void z4();
        //Установить Сигнал "Ошибка клапана"
        void z5();
        //Сбросить Сигнал "Ошибка клапана"
        void z6();
        //Задержки времени автомата A3
        //Сигнал старта задержки времени на открытие
        void t1();
        //Сигнал старта задержки времени на паузу
        void t2();
        //Сигнал старта задержки времени на закрытие
        void t3();
        //Сигнал сброса задержки времени на открытие
        void tz1();
        //Сигнал сброса задержки времени на паузу в открытом состоянии
        void tz2();
        //Сигнал сброса задержки времени на закрытие
```

```
        void tz3();
        //Срабатывание задержки на открытие
        bool T1();
        //Срабатывание задержки на паузу в открытом состоянии
        bool T2();
        //Срабатывание задержки на закрытие
        bool T3();

        //Переменная, кодируещая состояние автомата A3
        //Состояния: 0 - "Готов к работе"; 1 - "Открывается"; 2 - "Открыт";
        //3 - "Закрывается"; 4 - "Неисправность"
        int y3;
        //Процедура автомата A3
        void A3(int e);
};

//Инициализация
void TClapan::Init(int ClapanNo, TCoffeeModel *NewCoffeeModel)
{
    CoffeeModel = NewCoffeeModel;
    ClapanID = ClapanNo;
    y3 = 0;
}
//Вызовы автомата A3
//Вызов автомата A3 с событием e0 (Проверка условий переходов)
void TClapan::Go()
{
    if (y3 == 0) return;
    A3(0);
}
//Вызов автомата A3 с событием e1 (запуск цикла работы клапана)
void TClapan::Open()
{
    A3(1);
}
//Вызов автомата A3 с событием e2 (сброс ошибки клапана)
void TClapan::Reset()
{
    A3(2);
}
//Проверка состояния автомата A3
//Проверка состояния "Готов к работе"
bool TClapan::is_Ready()
{
    return (y3 == 0);
}
//Проверка состояния "Открывается"
bool TClapan::is_Opening()
{
    return (y3 == 1);
}
//Проверка состояния "Открыт"
bool TClapan::is_Opened()
{
    return (y3 == 2);
}
//Проверка состояния "Закрывается"
bool TClapan::is_Closing()
{
    return (y3 == 3);
}
```

```
//Проверка состояния "Неисправность"
bool TClapan::is_Error()
{
    return (y3 == 4);
}
//Входные переменные автомата A3
//x1 - клапан открыт
bool TClapan::x1()
{
    bool result;
    AnsiString sResult;
    result = CoffeeModel->Clapan_isOpen(ClapanID);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "x1 - клапан" + IntToStr(ClapanID+1) + " открыт - вернул "
+ sResult );
    return result;
}
//x2 - клапан закрыт
bool TClapan::x2()
{
    bool result;
    AnsiString sResult;
    result = CoffeeModel->Clapan_isClosed(ClapanID);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "x2 - клапан" + IntToStr(ClapanID+1) + " открыт - вернул "
+ sResult );
    return result;
}
//Срабатывание задержки на открытие
bool TClapan::T1()
{
    bool result;
    AnsiString sResult;
    result = CoffeeModel->T1(ClapanID);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "T1 - Задержка времени на открытие клапана" +
IntToStr(ClapanID+1) + " прошла - вернул " + sResult );
    return result;
}
//Срабатывание задержки на паузу в открытом состоянии
bool TClapan::T2()
{
    bool result;
    AnsiString sResult;
    result = CoffeeModel->T2(ClapanID);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "T2 - Задержка времени на открытое состояние клапана" +
IntToStr(ClapanID+1) + " прошла - вернул " + sResult );
    return result;
}
//Срабатывание задержки на закрытие
bool TClapan::T3()
{
    bool result;
    AnsiString sResult;
    result = CoffeeModel->T3(ClapanID);
    sResult = (result) ? "true" : "false";
    AppendToLog(">", "T3 - Задержка времени на закрытие клапана" +
IntToStr(ClapanID+1) + " прошла - вернул " + sResult );
    return result;
}
```

```
//Выходные воздействия автомата A3
//Установить сигнал "Открыть"
void TClapan::z1()
{
    AppendToLog("*", "z1 - установить сигнал открыть клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->Clapan_SetOpen(ClapanID, true);
}
//Сбросить сигнал "Открыть"
void TClapan::z2()
{
    AppendToLog("*", "z2 - сбросить сигнал открыть клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->Clapan_SetOpen(ClapanID, false);
}
//Установить сигнал "Закрыть"
void TClapan::z3()
{
    AppendToLog("*", "z3 - установить сигнал закрыть клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->Clapan_SetClose(ClapanID, true);//    return true;
}
//Сбросить сигнал "Закрыть"
void TClapan::z4()
{
    AppendToLog("*", "z4 - сбросить сигнал закрыть клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->Clapan_SetClose(ClapanID, false);//    return true;
}
//Установить Сигнал "Ошибка клапана"
void TClapan::z5()
{
    AppendToLog("*", "z5 - установить сигнал ошибки клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->SetDisplay2("ЗАПАЛ КЛ. " + IntToStr(ClapanID+1));
}
//Сбросить Сигнал "Ошибка клапана"
void TClapan::z6()
{
    AppendToLog("*", "z2 - сбросить сигнал ошибки клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->SetDisplay2("");
}
//Сигнал старта задержки времени на открытие
void TClapan::t1()
{
    AppendToLog("*", "t1 - запуск задержки времени на открытие клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->t1(ClapanID);
}
//Сигнал старта задержки времени на паузу
void TClapan::t2()
{
    AppendToLog("*", "t2 - запуск задержки времени на открытое состояние
клапана" + IntToStr(ClapanID+1));
    CoffeeModel->t2(ClapanID);
}
//Сигнал старта задержки времени на закрытие
void TClapan::t3()
{
    AppendToLog("*", "t3 - запуск задержки времени на закрытие клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->t3(ClapanID);
}
```

```
//Сигнал сброса задержки времени на открытие
void TClapan::tz1()
{
    AppendToLog("*", "tz1 - сброс задержки времени на открытие клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->tz1(ClapanID);
}
//Сигнал сброса задержки времени на паузу в открытом состоянии
void TClapan::tz2()
{
    AppendToLog("*", "tz2 - сброс задержки времени на открытое состояние
клапана" + IntToStr(ClapanID+1));
    CoffeeModel->tz2(ClapanID);
}
//Сигнал сброса задержки времени на закрытие
void TClapan::tz3()
{
    AppendToLog("*", "tz3 - сброс задержки времени на закрытие клапана" +
IntToStr(ClapanID+1));
    CoffeeModel->tz3(ClapanID);
}
//Процедура автомата
void TClapan::A3(int e)
{
    int old_y3;
    AnsiString StateDescr;
    AnsiString NewStateDescr;
    AnsiString EventDescr;
    bool bx1;
    bool bx2;
    bool bT1;
    bool bT2;
    bool bT3;

    old_y3 = y3;
    //Записываем в лог
    switch (y3)
        {
        case 0:
            StateDescr = " (\"Готов к работе\")";
        break;

        case 1:
            StateDescr = " (\"Открывается\")";
        break;

        case 2:
            StateDescr = " (\"Открыт\")";
        break;

        case 3:
            StateDescr = " (\"Закрывается\")";
        break;

        case 4:
            StateDescr = " (\"Неисправность\")";
        break;

        default:
            StateDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }
```

```
    switch (e)
        {
        case 0:
            EventDescr = " (\"Проверка условий переходов\")";
        break;

        case 1:
            EventDescr = " (\"Запуск цикла работы клапана\")";
        break;

        case 2:
            EventDescr = " (\"Сброс ошибки клапана\")";
        break;

        default:
            EventDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }
    AppendToLog("{", "A3_"+ IntToStr(ClapanID+1)+": в состоянии " + IntToStr(y3)
+ StateDescr + " запущен с событием e" + IntToStr(e) + EventDescr);

    bx1 = x1();
    bx2 = x2();
    bT1 = T1();
    bT2 = T2();
    bT3 = T3();

    switch (y3)
    {
        case 0:
            if (e == 1)       {z1(); t1();                    y3 = 1;};
        break;

        case 1:
            if (bT1 && bx1)   {tz1(); z2(); t2();             y3 = 2;}
            else
            if (bT1 &&!bx1)   {tz1(); z5(); z2();             y3 = 4;}
        break;

        case 2:
            if (bT2 )          {tz2(); z3(); t3();            y3 = 3;}
        break;

        case 3:
            if (bT3 && bx2)   {tz3(); z4();                   y3 = 0;}
            else
            if (bT3 &&!bx2)   {tz3(); z5(); z4();             y3 = 4;}
        break;

        case 4:
            if (e == 2)         {z6();                        y3 = 0;}
        break;
    }

    switch (y3)
        {
        case 4:
            z4(); z5();
        break;
        }
```

```
    //Запись в лог
    switch (y3)
        {
        case 0:
            NewStateDescr = " (\"Готов к работе\")";
        break;

        case 1:
            NewStateDescr = " (\"Открывается\")";
        break;

        case 2:
            NewStateDescr = " (\"Открыт\")";
        break;

        case 3:
            NewStateDescr = " (\"Закрывается\")";
        break;

        case 4:
            NewStateDescr = " (\"Неисправность\")";
        break;

        default:
            StateDescr = " !!!  ОШИБКА ЛОГА  !!!";
        break;
        }

    if (y3 != old_y3)
        {
        AppendToLog("T", "A3_"+ IntToStr(ClapanID+1)+": перешел из состояния " +
IntToStr(old_y3) + StateDescr + " в состояние " + IntToStr(y3) + NewStateDescr);
        }
    AppendToLog("}", "A3_"+ IntToStr(ClapanID+1)+": завершил обработку события
e" + IntToStr(e) + EventDescr + " в состоянии " + IntToStr(y3) + NewStateDescr);

}
```

11. Фрагмент протокола

```
20:35:04 { A0: в состоянии 3 ("Варит кофе") запущен с событием e0 ("Проверка условий переходов")
20:35:04 {     A1: в состоянии 3 ("Набирает воду") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >         x6 - нет воды - вернул false
20:35:04 >         x7 - сварено нужное количество порций - вернул false
20:35:04 >         T5 - Задержка времени на кипячение прошла - вернул false
20:35:04 {         A2_1: в состоянии 3 ("Выключен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя1 меньше заданной минимальной - вернул true
20:35:04 >             x4 - температура нагревателя1 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя1 больше заданной минимальной - вернул false
20:35:04 >             T4 - Задержка времени на прогрев нагревателя1 прошла - вернул false
20:35:04 *             z1 - установить сигнал нагревателя1
20:35:04 T             A2_1: перешел из состояния 3 ("Выключен") в состояние 2 ("Включен")
20:35:04 }         A2_1: завершил обработку события e0 ("Проверить условия переходов") в состоянии 2
("Включен")
20:35:04 {         A2_2: в состоянии 3 ("Выключен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя2 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя2 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя2 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя2 прошла - вернул false
20:35:04 }         A2_2: завершил обработку события e0 ("Проверить условия переходов") в состоянии 3
("Выключен")
20:35:04 {         A3_1: в состоянии 3 ("Закрывается") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >             x1 - клапан1 открыт - вернул false
20:35:04 >             x2 - клапан1 открыт - вернул true
20:35:04 >             T1 - Задержка времени на открытие клапана1 прошла - вернул false
20:35:04 >             T2 - Задержка времени на открытое состояние клапана1 прошла - вернул false
20:35:04 >             T3 - Задержка времени на закрытие клапана1 прошла - вернул false
20:35:04 }         A3_1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Закрывается")
20:35:04 }     A1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Набирает воду")
20:35:04 } A0: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3 ("Варит
кофе")
20:35:04 { A0: в состоянии 3 ("Варит кофе") запущен с событием e0 ("Проверка условий переходов")
20:35:04 {     A1: в состоянии 3 ("Набирает воду") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >         x6 - нет воды - вернул false
20:35:04 >         x7 - сварено нужное количество порций - вернул false
20:35:04 >         T5 - Задержка времени на кипячение прошла - вернул false
20:35:04 {         A2_1: в состоянии 2 ("Включен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя1 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя1 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя1 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя1 прошла - вернул false
20:35:04 }         A2_1: завершил обработку события e0 ("Проверить условия переходов") в состоянии 2
("Включен")
20:35:04 {         A2_2: в состоянии 3 ("Выключен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя2 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя2 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя2 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя2 прошла - вернул false
20:35:04 }         A2_2: завершил обработку события e0 ("Проверить условия переходов") в состоянии 3
("Выключен")
20:35:04 {         A3_1: в состоянии 3 ("Закрывается") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >             x1 - клапан1 открыт - вернул false
20:35:04 >             x2 - клапан1 открыт - вернул true
20:35:04 >             T1 - Задержка времени на открытие клапана1 прошла - вернул false
20:35:04 >             T2 - Задержка времени на открытое состояние клапана1 прошла - вернул false
20:35:04 >             T3 - Задержка времени на закрытие клапана1 прошла - вернул false
20:35:04 }         A3_1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Закрывается")
20:35:04 }     A1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Набирает воду")
20:35:04 } A0: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3 ("Варит
кофе")
20:35:04 { A0: в состоянии 3 ("Варит кофе") запущен с событием e0 ("Проверка условий переходов")
20:35:04 {     A1: в состоянии 3 ("Набирает воду") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >         x6 - нет воды - вернул false
```

```
20:35:04 >         x7 - сварено нужное количество порций - вернул false
20:35:04 >         T5 - Задержка времени на кипячение прошла - вернул false
20:35:04 {         A2_1: в состоянии 2 ("Включен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя1 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя1 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя1 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя1 прошла - вернул false
20:35:04 }         A2_1: завершил обработку события e0 ("Проверить условия переходов") в состоянии 2
("Включен")
20:35:04 {         A2_2: в состоянии 3 ("Выключен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя2 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя2 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя2 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя2 прошла - вернул false
20:35:04 }         A2_2: завершил обработку события e0 ("Проверить условия переходов") в состоянии 3
("Выключен")
20:35:04 {         A3_1: в состоянии 3 ("Закрывается") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >             x1 - клапан1 открыт - вернул false
20:35:04 >             x2 - клапан1 открыт - вернул true
20:35:04 >             T1 - Задержка времени на открытие клапана1 прошла - вернул false
20:35:04 >             T2 - Задержка времени на открытое состояние клапана1 прошла - вернул false
20:35:04 >             T3 - Задержка времени на закрытие клапана1 прошла - вернул false
20:35:04 }         A3_1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Закрывается")
20:35:04 }     A1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Набирает воду")
20:35:04 } A0: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3 ("Варит
кофе")
20:35:04 { A0: в состоянии 3 ("Варит кофе") запущен с событием e0 ("Проверка условий переходов")
20:35:04 {     A1: в состоянии 3 ("Набирает воду") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >         x6 - нет воды - вернул false
20:35:04 >         x7 - сварено нужное количество порций - вернул false
20:35:04 >         T5 - Задержка времени на кипячение прошла - вернул false
20:35:04 {         A2_1: в состоянии 2 ("Включен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя1 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя1 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя1 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя1 прошла - вернул false
20:35:04 }         A2_1: завершил обработку события e0 ("Проверить условия переходов") в состоянии 2
("Включен")
20:35:04 {         A2_2: в состоянии 3 ("Выключен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя2 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя2 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя2 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя2 прошла - вернул false
20:35:04 }         A2_2: завершил обработку события e0 ("Проверить условия переходов") в состоянии 3
("Выключен")
20:35:04 {         A3_1: в состоянии 3 ("Закрывается") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >             x1 - клапан1 открыт - вернул false
20:35:04 >             x2 - клапан1 открыт - вернул true
20:35:04 >             T1 - Задержка времени на открытие клапана1 прошла - вернул false
20:35:04 >             T2 - Задержка времени на открытое состояние клапана1 прошла - вернул false
20:35:04 >             T3 - Задержка времени на закрытие клапана1 прошла - вернул false
20:35:04 }         A3_1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Закрывается")
20:35:04 }     A1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Набирает воду")
20:35:04 } A0: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3 ("Варит
кофе")
20:35:04 { A0: в состоянии 3 ("Варит кофе") запущен с событием e0 ("Проверка условий переходов")
20:35:04 {     A1: в состоянии 3 ("Набирает воду") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >         x6 - нет воды - вернул false
20:35:04 >         x7 - сварено нужное количество порций - вернул false
20:35:04 >         T5 - Задержка времени на кипячение прошла - вернул false
20:35:04 {         A2_1: в состоянии 2 ("Включен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя1 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя1 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя1 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя1 прошла - вернул false
20:35:04 }         A2_1: завершил обработку события e0 ("Проверить условия переходов") в состоянии 2
("Включен")
```

```
20:35:04 {         A2_2: в состоянии 3 ("Выключен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя2 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя2 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя2 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя2 прошла - вернул false
20:35:04 }         A2_2: завершил обработку события e0 ("Проверить условия переходов") в состоянии 3
("Выключен")
20:35:04 {         A3_1: в состоянии 3 ("Закрывается") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >             x1 - клапан1 открыт - вернул false
20:35:04 >             x2 - клапан1 открыт - вернул true
20:35:04 >             T1 - Задержка времени на открытие клапана1 прошла - вернул false
20:35:04 >             T2 - Задержка времени на открытое состояние клапана1 прошла - вернул false
20:35:04 >             T3 - Задержка времени на закрытие клапана1 прошла - вернул true
20:35:04 *             tz3 - сброс задержки времени на закрытие клапана1
20:35:04 *             z4 - сбросить сигнал закрыть клапана1
20:35:04 T             A3_1: перешел из состояния 3 ("Закрывается") в состояние 0 ("Готов к работе")
20:35:04 }         A3_1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 0
("Готов к работе")
20:35:04 }     A1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3
("Набирает воду")
20:35:04 } A0: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3 ("Варит
кофе")
20:35:04 { A0: в состоянии 3 ("Варит кофе") запущен с событием e0 ("Проверка условий переходов")
20:35:04 {     A1: в состоянии 3 ("Набирает воду") запущен с событием e0 ("Проверка условий
переходов")
20:35:04 >         x6 - нет воды - вернул false
20:35:04 >         x7 - сварено нужное количество порций - вернул false
20:35:04 >         T5 - Задержка времени на кипячение прошла - вернул false
20:35:04 *         t4 - установка задержки времени на кипячение
20:35:04 {         A2_1: в состоянии 2 ("Включен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя1 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя1 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя1 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя1 прошла - вернул false
20:35:04 }         A2_1: завершил обработку события e0 ("Проверить условия переходов") в состоянии 2
("Включен")
20:35:04 {         A2_2: в состоянии 3 ("Выключен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя2 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя2 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя2 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя2 прошла - вернул false
20:35:04 }         A2_2: завершил обработку события e0 ("Проверить условия переходов") в состоянии 3
("Выключен")
20:35:04 T         A1: перешел из состояния 3 ("Набирает воду") в состояние 4 ("Кипятит")
20:35:04 }     A1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 4
("Кипятит")
20:35:04 } A0: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3 ("Варит
кофе")
20:35:04 { A0: в состоянии 3 ("Варит кофе") запущен с событием e0 ("Проверка условий переходов")
20:35:04 {     A1: в состоянии 4 ("Кипятит") запущен с событием e0 ("Проверка условий переходов")
20:35:04 >         x6 - нет воды - вернул false
20:35:04 >         x7 - сварено нужное количество порций - вернул false
20:35:04 >         T5 - Задержка времени на кипячение прошла - вернул false
20:35:04 {         A2_1: в состоянии 2 ("Включен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя1 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя1 больше заданной максимальной - вернул true
20:35:04 >             x5 - температура нагревателя1 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя1 прошла - вернул false
20:35:04 *             z1 - сбросить сигнал нагревателя1
20:35:04 T             A2_1: перешел из состояния 2 ("Включен") в состояние 3 ("Выключен")
20:35:04 }         A2_1: завершил обработку события e0 ("Проверить условия переходов") в состоянии 3
("Выключен")
20:35:04 {         A2_2: в состоянии 3 ("Выключен") запущен с событием e0 ("Проверить условия
переходов")
20:35:04 >             x3 - температура нагревателя2 меньше заданной минимальной - вернул false
20:35:04 >             x4 - температура нагревателя2 больше заданной максимальной - вернул false
20:35:04 >             x5 - температура нагревателя2 больше заданной минимальной - вернул true
20:35:04 >             T4 - Задержка времени на прогрев нагревателя2 прошла - вернул false
20:35:04 }         A2_2: завершил обработку события e0 ("Проверить условия переходов") в состоянии 3
("Выключен")
20:35:04 }     A1: завершил обработку события e0 ("Проверка условий переходов") в состоянии 4
("Кипятит")
20:35:04 } A0: завершил обработку события e0 ("Проверка условий переходов") в состоянии 3 ("Варит
кофе")
```