Содержание

[Содержание 3](#_Toc516314397)

[Реферат 4](#_Toc516314398)

[Введение 5](#_Toc516314399)

[1. Теоретическое введение 6](#_Toc516314400)

[1.1 Объектно-ориентированное программирование 6](#_Toc516314401)

[1.2 Модульное программирование 7](#_Toc516314402)

[1.3 Работа с внешними файлами 8](#_Toc516314403)

[1.4 UML проектирование 9](#_Toc516314404)

[2. Проектирование 12](#_Toc516314405)

[2.1 Спецификации программы 12](#_Toc516314406)

[2.2. Список выявленных понятий 13](#_Toc516314407)

[2.3. Проектирование программного обеспечения 15](#_Toc516314408)

[2.3.1. Уточнение отношений между классами 15](#_Toc516314409)

[2.3.2. Составление текстового интерфейса классов 16](#_Toc516314410)

[2.3.3. Структура файлов для хранения данных программы 18](#_Toc516314411)

[2.4. Реализация графического интерфейса 19](#_Toc516314412)

[3. Тестирование программной системы 20](#_Toc516314413)

[3.1. Руководство пользователю 20](#_Toc516314414)

[3.2. Тестирование 20](#_Toc516314415)

[Заключение 24](#_Toc516314416)

[Список литературы 25](#_Toc516314417)

[Приложения 26](#_Toc516314418)

# **Реферат**

Курсовая работа содержит введение, теоретическое введение, проектирование, экспериментальную часть и заключение:

В введении изложены основные цели и задачи курсовой работы, а также подробное описание объекта проектирования. Также в введении указана среда разработки работы.

В теоретическом введении кратко описаны основные принципы ООП, модульного программирования, работы с файлами и проектирования UML.

В проектировании подробно описан весь процесс создания программного обеспечения, представлены следующие диаграммы:

1. Диаграмма основных вариантов использования.
2. Диаграммы вариантов использования
3. Диаграмма последовательности действий
4. Диаграмма классов

В экспериментальной части проведена инструкция пользователя и тестирование, показывающее основные аспекты работы программы.

В списке литературы указаны основные источники литературы, которые были использованы в ходе создания теоретического введения и проектирования.

Курсовая работа содержит: 79 страниц, основных разделов, включая приложение с листингами, 4 таблицы, 13 рисунков, 4 источника литературы.

# **Введение**

Цель курсовой работы - получение практических навыков по проектированию и реализации программной системы с применением объектно-ориентированного подхода

Данная тема позволит на практике разработать объектно-ориентированную систему на примере кофейного автомата. Реализация системы покажет основные направления развития программного обеспечения автоматов, которое не получало больших изменений последнее десятилетие.

Объект курсовой работы – кофейный автомат.

Основной функционал:

* Поступление суммы
* Приготовление кофе по заказу покупателя
* Выдача сдачи
* Выдача заказа

Принцип работы: после внесения достаточной суммы автомат предлагает выбрать вид напитка и количество сахара, и после подтверждения покупателем заказа, приготовление и выдача напитка клиенту вместе со сдачей.

Основные задачи:

1. выделить основные классы;
2. составить объектную систему;
3. реализовать методы классов;
4. разработать графический интерфейс и работу с внешними файлами.

Реализация программной части выполнена на языке C++ в программной среде MS Visual Studio 2017.

# **Теоретическое введение**

## Объектно-ориентированное программирование

Объектно-ориентированное программирование – подход в программировании, при котором программа представляется как система объектов со своими свойствами и методами.

Основные принципы ООП:

1. Наследование – механизм, позволяющий создать новый класс на основе базового класса, используя его методы и свойства.
2. Инкапсуляция – механизм, позволяющий объединять и скрывать от внешнего доступа методы и свойства класса.

Модификаторы доступа в C++:

* public – доступ к полю с этим модификатором.
* protected – доступ к полю с этим модификатором доступен только методам класса и наследникам.
* private – доступ к полю с этим модификатором закрыт всем, кроме методов класса.

1. Полиморфизм – свойство системы, позволяющее использовать объект без информации о его структуре.

В C++ класс – абстрактный тип данных, описывающий методы и свойства несуществующего объекта.

Общая структура класса:

class Class

{

int num1;

Class();

void set\_num(int);

};

Объект – сущность, созданная на основе класса и обладающая всеми его методами и свойствами.

Основные компоненты класса:

1. Поля (атрибуты) класса – переменные, описывающие свойства объектов.
2. Конструктор – специальный метод класса, вызываемый при инициализации экземпляра класса и присваивающие ему начальные значения, имеет такое же имя что и класс.
3. Деструктор – специальный метод класса, вызываемый при уничтожении экземпляра класса
4. Методы - функции, объявленные внутри класса и имеющие доступ ко всем атрибутам класса.

## Модульное программирование

Модульное программирование – подход разработки программного обеспечения, в котором программа делится на более простые части (функции). Похожие функции могут быть объединены и вынесены в отдельный файл(модуль).

Преимущества модульного программирования:

1. Более быстрое редактирование кода, так как программист изменяет только один модуль, не вмешиваясь в код других.
2. Более быстрая компиляция, так как можно компилировать модули отдельно.
3. Облегченный процесс отладки, так как программа разбита на модули.

Функция – обособленная и функционально законченная часть кода, выполняющая определённую задачу. В С++ выделяют два типа функций: с типом и без типа. Связь программы с функцией в С++ осуществляется через механизм параметров.

Основные способы передачи значений в функцию:

* Параметр значение (int a) – используется для передачи значения в функцию, для этого выделяется память в стеке и создаётся копия, поэтому функция не может изменить значение переменной
* Параметр указатель (int\* a) – используется для передачи значения в функцию, в стеке выделяется память для адреса копии переменной, созданной в куче, функция может изменить значение переменной. Используется для передачи в функцию массива или переменной, значение которой является побочным результатом функции.
* Параметр ссылка (int& a) – используется для передачи значения в функцию, в стеке выделяется память для адреса переменной, и функция работает с переменной. Этот способ экономит память, но применим только к переменным.
* Параметр константа (const int a) – используется для передачи значения в функцию, но значение параметра нельзя изменить в функции и использовать в левой части выражений.

Модуль в C++ - это совокупность схожих функций, вынесенных из основной программы в дополнительный файл(\*.cpp или \*.h). В главном модуле программы обычно находится функция int main(), являющаяся точкой входа в программу. Для использования других модулей в программе необходимо подключить их заголовочный файл через директиву предпроцессора #include.

## Работа с внешними файлами

Файл – это структура данных для хранения информации на внешнем носителе. Взаимодействие программы с внешними текстовыми файлами в C++ удобно реализовать с помощью потоков ввода-вывода библиотеки fstream.

1. Физический файл – это файл на внешнем носителе.
2. Логический файл – объект, представляющий файл в программе.

Поток – программный интерфейс, используемый для ввода вывода информации через файлы, сокеты или консоль.

Для работы с файлами необходимо создать переменную класса потока:

1. ifstream if; - для записи данных в файл
2. ofstream of; - для чтения данных из файла
3. fstream f; - для чтения – записи данных из файла
4. Связать файл с потоком можно двумя способами:
   1. При создании потока:
   2. С помощью метода open();
5. Ввод-вывод данных через поток:
   1. С помощью оператора >> или <<;
   2. С помощью функции getline();
6. Закрытие файлового потока происходит через метод close();

## UML проектирование

UML – унифицированный язык моделирования, использующийся для проектирования, описания и реализации систем программного обеспечения.

В UML существует множество диаграмм, которые охватывают все аспекты работы программной системы.

Диаграмма классов (рисунок 1) - служит для визуализации системы классов и отображения отношений классов на этапе проектирования.

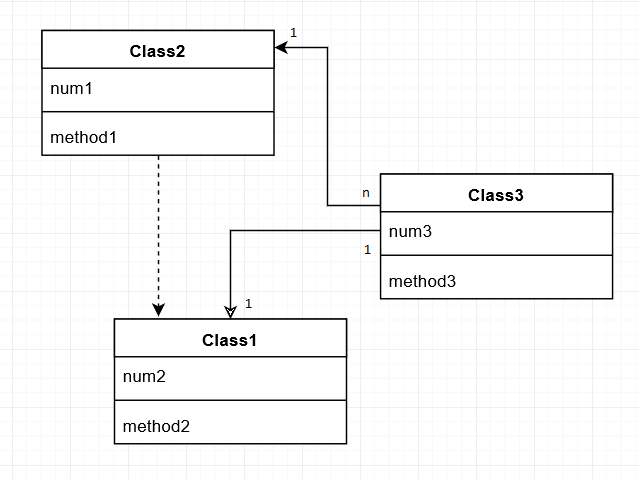


Рисунок 1. Пример диаграммы классов.

Основные элементы диаграммы классов:

1. Классы. Пиктограмма класса состоит из названия класса, свойств и методов (рисунок 2).

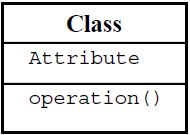


Рисунок 2. Пиктограмма класса.

1. Отношения между классами: ассоциация, обобщение, агрегация, объединение (рисунок 3).

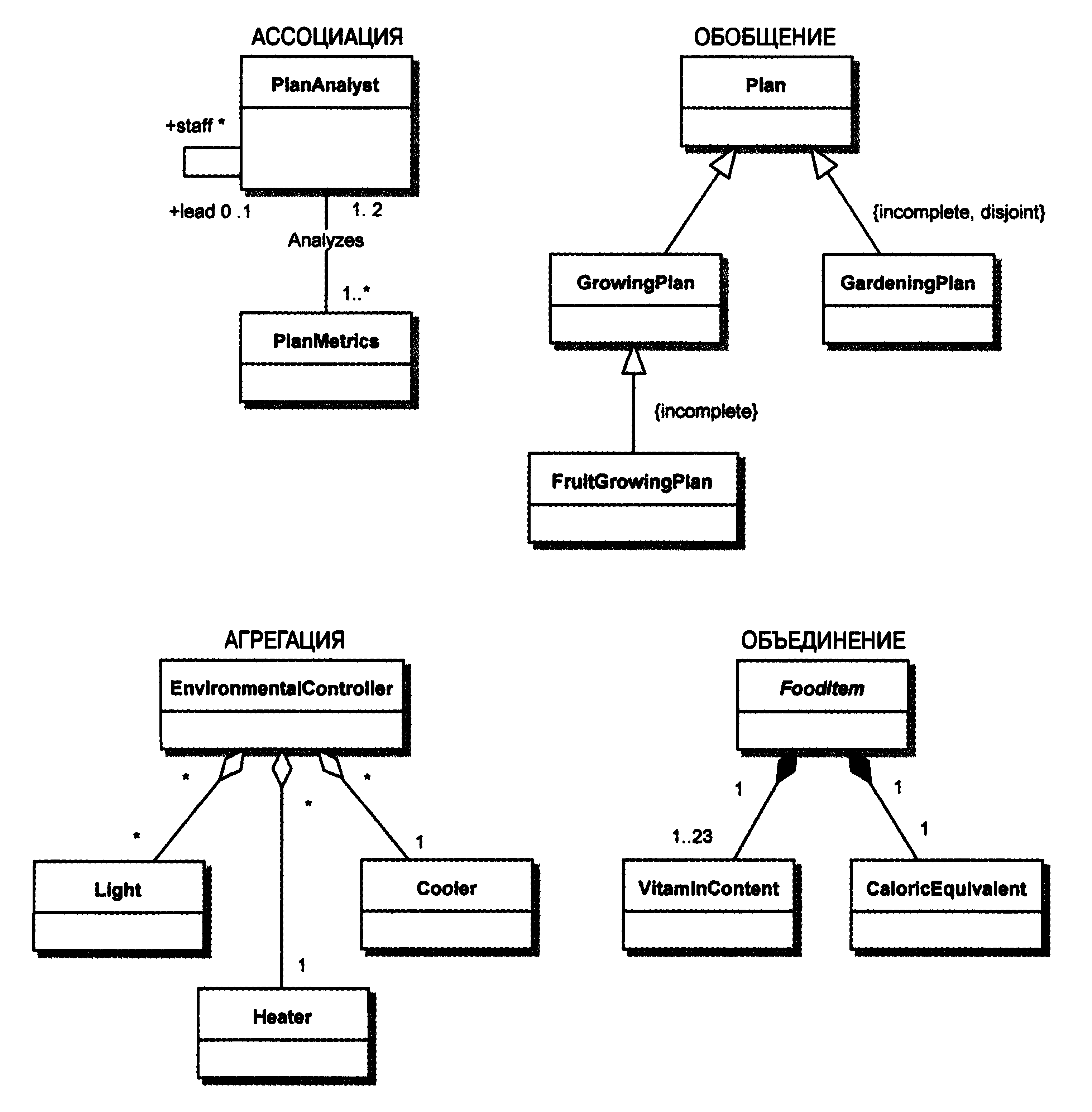


Рисунок 3. Виды атрибутов.

Видимость атрибутов и методов классов можно задать с помощью специальных символов:

“+” – public

“-“ – private

“#” – protected

Возле пиктограммы ассоциации обычно указывается кратность

“1” – одна связь

“\*” – неограниченное число связей

# **Проектирование**

## Спецификации программы

* Цель: работа программы направлена на успешную реализацию автомата по продаже кофе.
* Предполагаемая аудитория: посетители торгового центра без ограничения по возрасту.
* Масштаб проекта: несколько автоматов по продаже кофе.
* Функциональность: автомат должен принимать наличные деньги, выдавать сдачу, иметь интерфейс и предоставлять пользователю выбор типа напитка и количества кофе, предоставлять информацию о количестве ингредиентов, купюр и монет, иметь возможность обслуживания. Также информация о количестве ингредиентов, купюр и монет должна сохраняться во внешнем файле при закрытии программы.
* Среда функционирования продукта: операционная система MS Windows 7, 8, 8.1 , 10.
* Требования к производительности: программа должна функционировать стабильно на слабых системах, которыми обеспечиваются кофейные автоматы.
* Требования к сохранности файлов: информация о количестве ингредиентов, купюр и монет должна храниться на внешних файлах.
* Требования к безопасности системы: пользователь не должен иметь доступ к файлам с информацией о ресурсах и деньгах, также атрибуты классов связанные с кассой и ингредиентами должны быть закрыты внешним функциям модификатором доступа private.

На основе спецификаций была составлена диаграмма основных взаимодействий (рисунок 4).

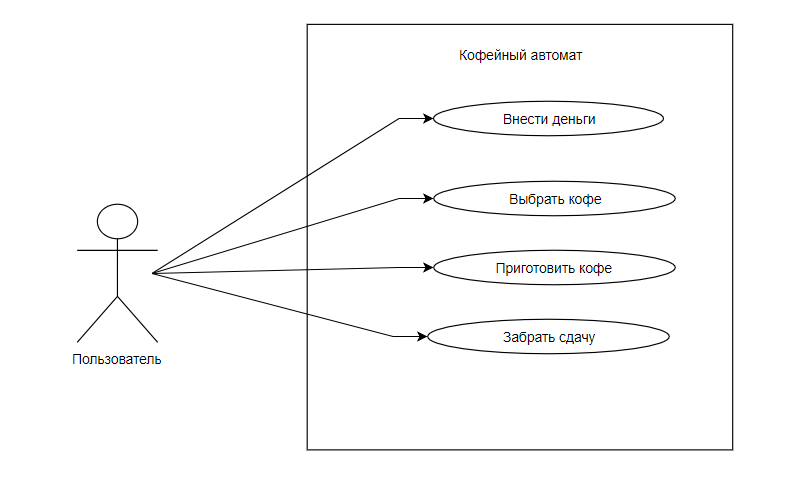


Рисунок 4. Диаграмма основных взаимодействий.

## Список выявленных понятий

Понятия: кофейный автомат, касса, деньги, купюры, монеты, ингредиенты, кофе, молоко, сахар, вода, взбитое молоко, чашки, графический интерфейс.

Список терминов, претендующих на класс: кофейный автомат, касса, деньги, ингредиенты, графический интерфейс.

На основе выявленного списка понятий, можно составить диаграммы взаимодействий:

Таблица 1. Описание варианта «Внос денег».

|  |  |
| --- | --- |
| Название варианта | Внос суммы |
| Цель | Совершить внос суммы в автомат |
| Действующие лица | Покупатель |
| Краткое описание | Система предлагает покупателю внести сумму в автомат. |
| Тип варианта | Основной |

Таблица 2. Описание варианта «Выбор кофе».

|  |  |
| --- | --- |
| Название варианта | Выбор кофе |
| Цель | Совершить выбор кофе |
| Действующие лица | Покупатель |
| Краткое описание | Система предлагает покупателю выбрать вид кофе из предложенных. |
| Тип варианта | Основной |

Таблица 3. Описание варианта «Приготовить кофе»

|  |  |
| --- | --- |
| Название варианта | Приготовить кофе |
| Цель | Совершить приготовление кофе |
| Действующие лица | Покупатель |
| Краткое описание | Система предлагает покупателю приготовить кофе после выбора типа кофе. |
| Тип варианта | Основной |

Таблица 4. Описание варианта «Забрать сдачу»

|  |  |
| --- | --- |
| Название варианта | Забрать сдачу |
| Цель | Совершить выдачу сдачи |
| Действующие лица | Покупатель |
| Краткое описание | Система предлагает покупателю выдать сдачу после приготовления кофе. |
| Тип варианта | Основной |

После выявления взаимодействий пользователя и программы, можно составить диаграмму последовательности действий (рисунок 6).

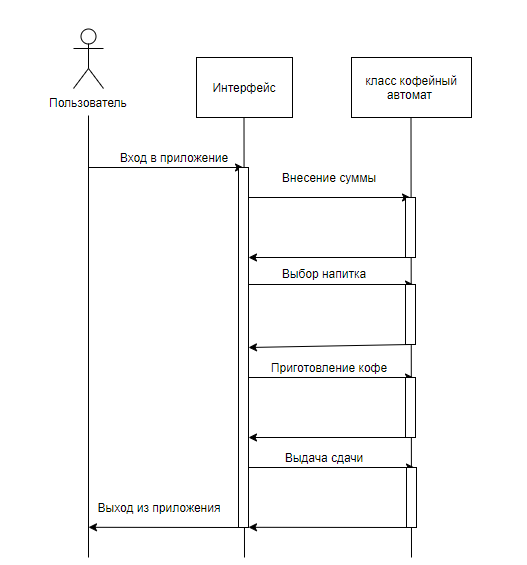


Рисунок 6. Диаграмма последовательности действий.

## Проектирование программного обеспечения

### 2.3.1. Уточнение отношений между классами

Класс CourseWorkForm связан композицией с классом Coffee\_machine.

Класс Coffee\_machine связан композицией с классами Bank и Consumables.

Класс Bank связан композицией с классом Money.

### Составление текстового интерфейса классов

Класс CourseWorkForm

Атрибуты класса:

Переменная класса Coffee\_machine

Методы класса: обработчики событий нажатий на кнопки.

Класс Coffee\_machine

Атрибуты класса:

1. Переменная класса Bank
2. Массив элементов класса Consumables размерностью 6

Методы класса:

1. Получить цену напитка по номеру
2. Проверка наличия ресурсов
3. Пополнение ресурсов
4. Выдача сдачи
5. Приготовление кофе
6. Проверка наличия денег
7. Покупка кофе
8. Инкассация кассы

Класс Bank

Атрибуты класса:

1. Переменная кредита типа int
2. Массив элементов класса Money размерностью 8

Методы класса

1. Выдать сдачу
2. Проверить количество купюр и монет
3. Купить кофе
4. Инкассация кассы
5. Установить количество купюр или монет
6. Получить количество купюр или монет
7. Увеличить кредит

Класс Money

Атрибуты класса:

1. Переменная номинала
2. Переменная количества элементов

Методы класса:

1. Установка значения номинала
2. Получение значения номинала
3. Установка значения количества
4. Получение значения количества

Класс Consumables

Атрибуты класса:

1. Переменная названия ингредиента
2. Переменная количества ингредиента

Методы класса:

1. Установка названия
2. Получение названия
3. Установка значения количества
4. Получение значения количества

На основе определённых классов и их методов, была разработана концептуальная модель предметной области, диаграмма классов (Рисунок 7)

### Диаграмма UML

Рисунок 7. Диаграмма классов.

### Структура файлов для хранения данных программы

Для работы программы требуется наличие двух текстовых файлов:

* profit.txt – для хранения номиналов и количества купюр. (Таблица 1)
* consumables.txt – для хранения информации о расходных материалах и их количестве. (Таблица 2)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица 5. Структура файла profit.txt.   |  |  | | --- | --- | | Номинал | Количество | | 1 | 46 | | 2 | 36 | | 5 | 26 | | 10 | 71 | | 50 | 56 | | 100 | 45 | | 500 | 34 | | 1000 | 23 | | Таблица 6. структура файла consumables.txt.   |  |  | | --- | --- | | Название | Количество | | milk | 23 | | sugar | 34 | | water | 52 | | cup | 56 | | coffee | 35 | | whipped milk | 12 | |

## Реализация графического интерфейса

Реализация графического интерфейса была выполнена в программной среде MS Visual studio с применением WindowsForms.

Основные цели:

* Реализовать в программе диалоговое окно для работы с пользователем как на настоящем аппарате
* Реализовать блок кнопок для выбора типа напитка и количества сахара
* Реализовать симуляцию выдачи сдачи и напитка
* Создать отдельный блок для контроля за количеством ресурсов и денег, а также для инкассации кассы и пополнения ресурсов.

Успешная реализация графического интерфейса, соответствующего основным целям и требованиям, представлена на (Рисунок 8)

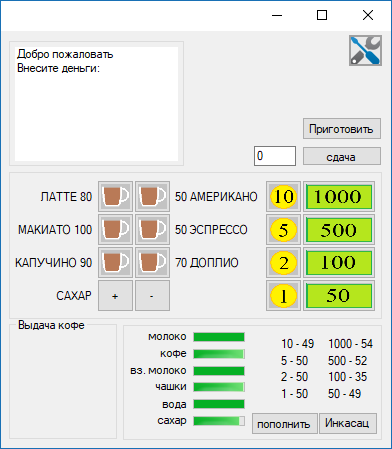


Рисунок 8. Графический интерфейс программы.

# **Тестирование программной системы**

## Руководство пользователю

Основная функция приложения: приготовление кофе.

Инструкция для пользователя:

* Для начала работы внесите сумму (в диалоговом окне появится соответствующее предложение).
* Для выбора типа напитка следует нажать на кнопку рядом с названием, предварительно ознакомившись с ценой. Если внесённая сумма меньше чем стоимость напитка, автомат вас об этом предупредит.
* После выбора напитка автомат предложит выбрать количество сахара (по умолчанию 0). Выбор количества сахара происходит путём нажатия 2х кнопок.
* Для подтверждения своего выбора следует нажать на кнопку «Приготовить».

Инструкция инженера:

* Основное меню для инкассации и пополнения ингредиентов находится в отдельном меню, которое отображается по нажатию специальной кнопки.
* Для пополнения ингредиентов нажать кнопку «Пополнить».
* Для инкассации нажать кнопку «Инкассация».

## 3.2. Тестирование

Для начала работы с программой пользователь вносит деньги, выбирает напиток и количество сахара, нажимает кнопку приготовить (рисунок 9).

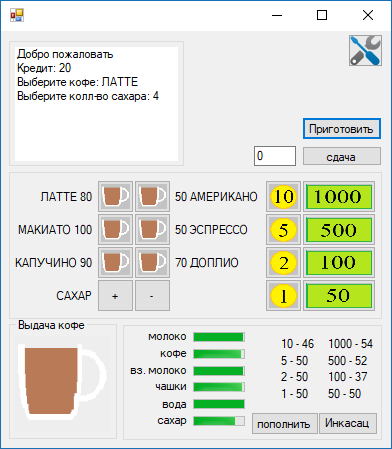


Рисунок 9. Прохождения полного цикла работы.

Автомат приготовил кофе и выдал его пользователю.

Для начала работы с автоматом пользователь внёс сумму в автомат (рисунок 10).

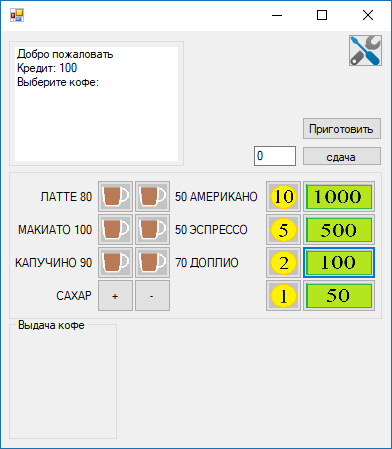


Рисунок 10. Внесение суммы в автомат.

Покупатель нажал кнопку выбора кофе перед внесением суммы (рисунок 11).

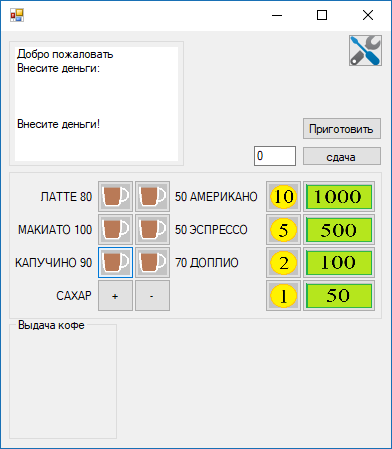


Рисунок 11. Внесение суммы в автомат.

Автомат предупредил пользователя, что необходимо внести сумму перед выбором напитка.

Пользователь нажал кнопку обслуживания автомата (рисунок 12).

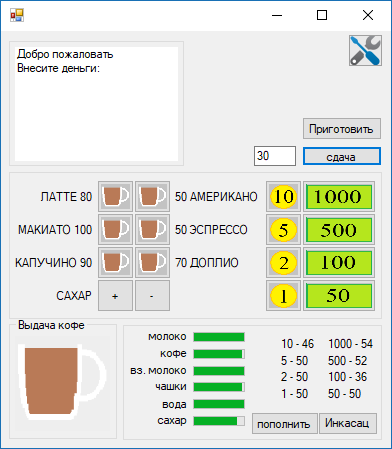


Рисунок 12. Нажатие кнопки обслуживания.

Покупатель внёс деньги, выбрал напиток и количество сахара, нажал кнопку приготовить (рисунок 13).

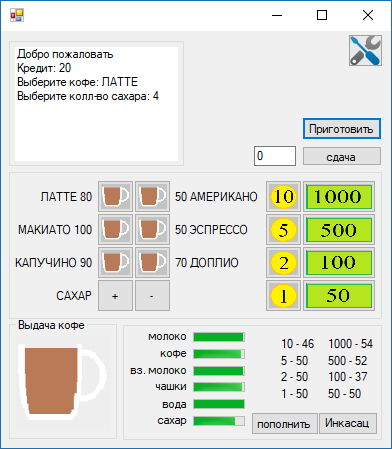


Рисунок 13. Выдача кофе.

Автоматвыполнил указанные пользователем действия, приготовил кофе и выдал его пользователю.

# **Заключение**

В ходе работы были выявлены и решены следующие задачи:

* При создании проекта возникла проблема с открытием консольного приложения вместе с графическим окном. Проблема была решена путём пересоздания проекта.
* При работе с графическим интерфейсом, было выявлено, что метод Text возвращает в формате String^. Для решения этой проблемы была написана функция преобразования String^ в string.
* При работе с файлами была выявлена проблема необходимости жёсткой привязки программы и файла к директории файловой системы. Проблема была успешно решена и теперь достаточно, чтоб файлы находились в папке и исполняемым файлом, независимо от компьютера.

По итогам была составлена программная система на основе объектного проектирования и реализовано графическое приложение.

Также в ходе выполнения курсовой работы были получены практические навыки по проектированию и разработке объектной системы.

# **Список литературы**

1. Бьёрн Страуструп – Программирование: принципы и практика с использованием C++ отдельное издание, пер в англ – Игорь Красиков Издательство «Вильямс» 2016 г. – 1328 с.
2. Р.Лафоре – Объектно-Ориентированное программирование в C++ Издательство «Питер СПб» 2016г. – 928 с.
3. Гради Буч, Роберт А. Максимчук – Объектно-ориентированный анализ и проектирование с примерами приложений 3-е издание Издательство «Вильямс» 2008 г. – 720 с.
4. Михаил Пайсон – ООП с примерами [Электронный ресурс] URL: https://habr.com/post/87119 (дата обращения 13.05.2018).
5. И.В. Черпаков – Основы программирования отдельное издание Издательство «Юрайт» 2017 – 200 с.

# **Приложения**

|  |
| --- |
| #include "CourseWorkForm.h"  using namespace System;  using namespace System::Windows::Forms;  [STAThreadAttribute]  int main()  {  Application::EnableVisualStyles();  Application::SetCompatibleTextRenderingDefault(false);  CourseWork::CourseWorkForm form;  Application::Run(%form);  } |

Листинг 1. Исходный код модуля CourseWorkForm.cpp.

|  |
| --- |
| #pragma once  #include "fstream"  #include "iostream"  #include "string"  #include "Coffee\_machine.h"  #include "Bank.h"  namespace CourseWork  {  using namespace System;  using namespace System::ComponentModel;  using namespace System::Collections;  using namespace System::Windows::Forms;  using namespace System::Data;  using namespace System::Drawing;    Coffee\_machine coffee\_machine;  int type\_coffee = 0, sugar = -1;  /// <summary>  /// Сводка для CourseWorkForm  /// </summary>  public ref class CourseWorkForm : public System::Windows::Forms::Form  {  public:  CourseWorkForm(void)  {  InitializeComponent();  //  //TODO: добавьте код конструктора  //  }  protected:  /// <summary>  /// Освободить все используемые ресурсы.  /// </summary>  ~CourseWorkForm()  {  if (components)  {  delete components;  }  }  private: System::Windows::Forms::Button^ button1;  private: System::Windows::Forms::Button^ button2;  private: System::Windows::Forms::Button^ button3;  private: System::Windows::Forms::Button^ button4;  private: System::Windows::Forms::Button^ button5;  private: System::Windows::Forms::Button^ button6;  private: System::Windows::Forms::Button^ button7;  private: System::Windows::Forms::Button^ button8;  private: System::Windows::Forms::Button^ button9;  private: System::Windows::Forms::Button^ button10;  private: System::Windows::Forms::Button^ button11;  private: System::Windows::Forms::Button^ button12;  private: System::Windows::Forms::Button^ button13;  private: System::Windows::Forms::Button^ button14;  private: System::Windows::Forms::Label^ label1;  private: System::Windows::Forms::Label^ label2;  private: System::Windows::Forms::Label^ label3;  private: System::Windows::Forms::Label^ label4;  private: System::Windows::Forms::Label^ label5;  private: System::Windows::Forms::Label^ label6;  private: System::Windows::Forms::Button^ button15;  private: System::Windows::Forms::Button^ button16;  private: System::Windows::Forms::Label^ label7;  private: System::Windows::Forms::PictureBox^ pictureBox1;  private: System::Windows::Forms::GroupBox^ groupBox1;  private: System::Windows::Forms::Button^ button17;  private: System::Windows::Forms::ListView^ listView1;  private: System::Windows::Forms::TextBox^ textBox1;  private: System::Windows::Forms::Button^ button18;  private: System::Windows::Forms::Timer^ timer1;  private: System::Windows::Forms::GroupBox^ groupBox2;  private: System::Windows::Forms::ProgressBar^ progressBar1;  private: System::Windows::Forms::ProgressBar^ progressBar2;  private: System::Windows::Forms::ProgressBar^ progressBar3;  private: System::Windows::Forms::ProgressBar^ progressBar4;  private: System::Windows::Forms::ProgressBar^ progressBar5;  private: System::Windows::Forms::ProgressBar^ progressBar6;  private: System::Windows::Forms::Label^ label8;  private: System::Windows::Forms::Label^ label9;  private: System::Windows::Forms::Label^ label10;  private: System::Windows::Forms::Label^ label11;  private: System::Windows::Forms::Label^ label12;  private: System::Windows::Forms::Label^ label13;  private: System::Windows::Forms::GroupBox^ groupBox3;  private: System::Windows::Forms::Button^ button19;  private: System::Windows::Forms::GroupBox^ groupBox4;  private: System::Windows::Forms::Button^ button20;  private: System::Windows::Forms::Label^ label18;  private: System::Windows::Forms::Label^ label19;  private: System::Windows::Forms::Label^ label20;  private: System::Windows::Forms::Label^ label21;  private: System::Windows::Forms::Label^ label16;  private: System::Windows::Forms::Label^ label17;  private: System::Windows::Forms::Label^ label15;  private: System::Windows::Forms::Label^ label14;  private: System::Windows::Forms::Button^ button21;  private: System::Windows::Forms::OpenFileDialog^ openFileDialog1;  private: System::ComponentModel::IContainer^ components;  protected:  private:  /// <summary>  /// Обязательная переменная конструктора.  /// </summary>  #pragma region Windows Form Designer generated code  /// <summary>  /// Требуемый метод для поддержки конструктора — не изменяйте  /// содержимое этого метода с помощью редактора кода.  /// </summary>  void InitializeComponent(void)  {  this->components = (gcnew  System::ComponentModel::Container());  System::ComponentModel::ComponentResourceManager^  resources = (gcnew System::ComponentModel::  ComponentResourceManager(CourseWorkForm::typeid));  this->button1 = (gcnew System::Windows::Forms::Button());  this->button2 = (gcnew System::Windows::Forms::Button());  this->button3 = (gcnew System::Windows::Forms::Button());  this->button4 = (gcnew System::Windows::Forms::Button());  this->button5 = (gcnew System::Windows::Forms::Button());  this->button6 = (gcnew System::Windows::Forms::Button());  this->button7 = (gcnew System::Windows::Forms::Button());  this->button8 = (gcnew System::Windows::Forms::Button());  this->button9 = (gcnew System::Windows::Forms::Button());  this->button10 = (gcnew System::Windows::Forms::Button());  this->button11 = (gcnew System::Windows::Forms::Button());  this->button12 = (gcnew System::Windows::Forms::Button());  this->button13 = (gcnew System::Windows::Forms::Button());  this->button14 = (gcnew System::Windows::Forms::Button());  this->label1 = (gcnew System::Windows::Forms::Label());  this->label2 = (gcnew System::Windows::Forms::Label());  this->label3 = (gcnew System::Windows::Forms::Label());  this->label4 = (gcnew System::Windows::Forms::Label());  this->label5 = (gcnew System::Windows::Forms::Label());  this->label6 = (gcnew System::Windows::Forms::Label());  this->button15 = (gcnew System::Windows::Forms::Button());  this->button16 = (gcnew System::Windows::Forms::Button());  this->label7 = (gcnew System::Windows::Forms::Label());  this->pictureBox1 = (gcnew  System::Windows::Forms::PictureBox());  this->groupBox1 = (gcnew  System::Windows::Forms::GroupBox());  this->button17 = (gcnew System::Windows::Forms::Button());  this->listView1 = (gcnew  System::Windows::Forms::ListView());  this->textBox1 = (gcnew  System::Windows::Forms::TextBox());  this->button18 = (gcnew System::Windows::Forms::Button());  this->timer1 = (gcnew System::Windows::Forms::Timer(this-  >components));  this->groupBox2 = (gcnew  System::Windows::Forms::GroupBox());  this->progressBar1 = (gcnew  System::Windows::Forms::ProgressBar());  this->progressBar2 = (gcnew  System::Windows::Forms::ProgressBar());  this->progressBar3 = (gcnew  System::Windows::Forms::ProgressBar());  this->progressBar4 = (gcnew  System::Windows::Forms::ProgressBar());  this->progressBar5 = (gcnew  System::Windows::Forms::ProgressBar());  this->progressBar6 = (gcnew  System::Windows::Forms::ProgressBar());  this->label8 = (gcnew System::Windows::Forms::Label());  this->label9 = (gcnew System::Windows::Forms::Label());  this->label10 = (gcnew System::Windows::Forms::Label());  this->label11 = (gcnew System::Windows::Forms::Label());  this->label12 = (gcnew System::Windows::Forms::Label());  this->label13 = (gcnew System::Windows::Forms::Label());  this->groupBox3 = (gcnew  System::Windows::Forms::GroupBox());  this->button19 = (gcnew System::Windows::Forms::Button());  this->groupBox4 = (gcnew  System::Windows::Forms::GroupBox());  this->label18 = (gcnew System::Windows::Forms::Label());  this->label19 = (gcnew System::Windows::Forms::Label());  this->label20 = (gcnew System::Windows::Forms::Label());  this->label21 = (gcnew System::Windows::Forms::Label());  this->label16 = (gcnew System::Windows::Forms::Label());  this->label17 = (gcnew System::Windows::Forms::Label());  this->label15 = (gcnew System::Windows::Forms::Label());  this->label14 = (gcnew System::Windows::Forms::Label());  this->button21 = (gcnew System::Windows::Forms::Button());  this->button20 = (gcnew System::Windows::Forms::Button());  this->openFileDialog1 = (gcnew  System::Windows::Forms::OpenFileDialog());  (cli::safe\_cast<System::ComponentModel::  ISupportInitialize^>(this->pictureBox1))->BeginInit();  this->groupBox1->SuspendLayout();  this->groupBox2->SuspendLayout();  this->groupBox3->SuspendLayout();  this->groupBox4->SuspendLayout();  this->SuspendLayout();  //  // button1  //  this->button1->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button1.BackgroundImage")));  this->button1->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button1->Location = System::Drawing::Point(256,  113);  this->button1->Name = L"button1";  this->button1->Size = System::Drawing::Size(37, 33);  this->button1->TabIndex = 0;  this->button1->UseVisualStyleBackColor = true;  this->button1->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button1\_Click);  //  // button2  //  this->button2->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button2.BackgroundImage")));  this->button2->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button2->Location = System::Drawing::Point(256, 80);  this->button2->Name = L"button2";  this->button2->Size = System::Drawing::Size(37, 33);  this->button2->TabIndex = 3;  this->button2->UseVisualStyleBackColor = true;  this->button2->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button2\_Click);  //  // button3  //  this->button3->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button3.BackgroundImage")));  this->button3->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button3->Location = System::Drawing::Point(256, 47);  this->button3->Name = L"button3";  this->button3->Size = System::Drawing::Size(37, 33);  this->button3->TabIndex = 4;  this->button3->UseVisualStyleBackColor = true;  this->button3->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button3\_Click);  //  // button4  //  this->button4->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button4.BackgroundImage")));  this->button4->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button4->Location = System::Drawing::Point(256, 14);  this->button4->Name = L"button4";  this->button4->Size = System::Drawing::Size(37, 33);  this->button4->TabIndex = 5;  this->button4->UseVisualStyleBackColor = true;  this->button4->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button4\_Click);  //  // button5  //  this->button5->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button5.BackgroundImage")));  this->button5->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button5->Location = System::Drawing::Point(293, 14);  this->button5->Name = L"button5";  this->button5->Size = System::Drawing::Size(74, 33);  this->button5->TabIndex = 6;  this->button5->UseVisualStyleBackColor = true;  this->button5->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button5\_Click);  //  // button6  //  this->button6->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button6.BackgroundImage")));  this->button6->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button6->Location = System::Drawing::Point(293, 47);  this->button6->Name = L"button6";  this->button6->Size = System::Drawing::Size(74, 33);  this->button6->TabIndex = 7;  this->button6->UseVisualStyleBackColor = true;  this->button6->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button6\_Click);  //  // button7  //  this->button7->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button7.BackgroundImage")));  this->button7->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button7->Location = System::Drawing::Point(293, 80);  this->button7->Name = L"button7";  this->button7->Size = System::Drawing::Size(74, 33);  this->button7->TabIndex = 8;  this->button7->UseVisualStyleBackColor = true;  this->button7->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button7\_Click);  //  // button8  //  this->button8->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button8.BackgroundImage")));  this->button8->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button8->Location = System::Drawing::Point(293,  113);  this->button8->Name = L"button8";  this->button8->Size = System::Drawing::Size(74, 33);  this->button8->TabIndex = 9;  this->button8->UseVisualStyleBackColor = true;  this->button8->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button8\_Click);  //  // button9  //  this->button9->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button9.BackgroundImage")));  this->button9->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button9->Location = System::Drawing::Point(88, 47);  this->button9->Name = L"button9";  this->button9->Size = System::Drawing::Size(37, 33);  this->button9->TabIndex = 10;  this->button9->UseVisualStyleBackColor = true;  this->button9->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button9\_Click);  //  // button10  //  this->button10->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button10.BackgroundImage")));  this->button10->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button10->Location = System::Drawing::Point(88, 80);  this->button10->Name = L"button10";  this->button10->Size = System::Drawing::Size(37, 33);  this->button10->TabIndex = 11;  this->button10->UseVisualStyleBackColor = true;  this->button10->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button10\_Click);  //  // button11  //  this->button11->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button11.BackgroundImage")));  this->button11->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button11->Location = System::Drawing::Point(88, 14);  this->button11->Name = L"button11";  this->button11->Size = System::Drawing::Size(37, 33);  this->button11->TabIndex = 12;  this->button11->UseVisualStyleBackColor = true;  this->button11->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button11\_Click);  //  // button12  //  this->button12->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button12.BackgroundImage")));  this->button12->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button12->Location = System::Drawing::Point(125,  14);  this->button12->Name = L"button12";  this->button12->Size = System::Drawing::Size(37, 33);  this->button12->TabIndex = 15;  this->button12->UseVisualStyleBackColor = true;  this->button12->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button12\_Click);  //  // button13  //  this->button13->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button13.BackgroundImage")));  this->button13->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button13->Location = System::Drawing::Point(125,  80);  this->button13->Name = L"button13";  this->button13->Size = System::Drawing::Size(37, 33);  this->button13->TabIndex = 14;  this->button13->UseVisualStyleBackColor = true;  this->button13->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button13\_Click);  //  // button14  //  this->button14->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button14.BackgroundImage")));  this->button14->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button14->Location = System::Drawing::Point(125,  47);  this->button14->Name = L"button14";  this->button14->Size = System::Drawing::Size(37, 33);  this->button14->TabIndex = 13;  this->button14->UseVisualStyleBackColor = true;  this->button14->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button14\_Click);  //  // label1  //  this->label1->AutoSize = true;  this->label1->Location = System::Drawing::Point(163, 24);  this->label1->Name = L"label1";  this->label1->Size = System::Drawing::Size(90, 13);  this->label1->TabIndex = 16;  this->label1->Text = L"50 АМЕРИКАНО";  //  // label2  //  this->label2->AutoSize = true;  this->label2->Location = System::Drawing::Point(163, 57);  this->label2->Name = L"label2";  this->label2->Size = System::Drawing::Size(80, 13);  this->label2->TabIndex = 17;  this->label2->Text = L"50 ЭСПРЕССО";  //  // label3  //  this->label3->AutoSize = true;  this->label3->Location = System::Drawing::Point(163, 90);  this->label3->Name = L"label3";  this->label3->Size = System::Drawing::Size(71, 13);  this->label3->TabIndex = 18;  this->label3->Text = L"70 ДОПЛИО";  //  // label4  //  this->label4->AutoSize = true;  this->label4->Location = System::Drawing::Point(29, 24);  this->label4->Name = L"label4";  this->label4->Size = System::Drawing::Size(58, 13);  this->label4->TabIndex = 19;  this->label4->Text = L"ЛАТТЕ 80";  //  // label5  //  this->label5->AutoSize = true;  this->label5->Location = System::Drawing::Point(6, 57);  this->label5->Name = L"label5";  this->label5->Size = System::Drawing::Size(81, 13);  this->label5->TabIndex = 20;  this->label5->Text = L"МАКИАТО 100";  //  // label6  //  this->label6->AutoSize = true;  this->label6->Location = System::Drawing::Point(3, 90);  this->label6->Name = L"label6";  this->label6->Size = System::Drawing::Size(84, 13);  this->label6->TabIndex = 21;  this->label6->Text = L"КАПУЧИНО 90";  //  // button15  //  this->button15->Location = System::Drawing::Point(88,  113);  this->button15->Name = L"button15";  this->button15->Size = System::Drawing::Size(37, 33);  this->button15->TabIndex = 22;  this->button15->Text = L"+";  this->button15->UseVisualStyleBackColor = true;  this->button15->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button15\_Click);  //  // button16  //  this->button16->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::None;  this->button16->ForeColor =  System::Drawing::SystemColors::ActiveCaptionText;  this->button16->Location = System::Drawing::Point(125,  113);  this->button16->Name = L"button16";  this->button16->Size = System::Drawing::Size(37, 33);  this->button16->TabIndex = 23;  this->button16->Text = L"-";  this->button16->UseVisualStyleBackColor = true;  this->button16->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button16\_Click);  //  // label7  //  this->label7->AutoSize = true;  this->label7->Location = System::Drawing::Point(45, 123);  this->label7->Name = L"label7";  this->label7->Size = System::Drawing::Size(42, 13);  this->label7->TabIndex = 24;  this->label7->Text = L"САХАР";  //  // pictureBox1  //  this->pictureBox1->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"pictureBox1.BackgroundImage")));  this->pictureBox1->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->pictureBox1->InitialImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"pictureBox1.InitialImage")));  this->pictureBox1->Location = System::Drawing::Point(6,  18);  this->pictureBox1->Name = L"pictureBox1";  this->pictureBox1->Size = System::Drawing::Size(95, 95);  this->pictureBox1->TabIndex = 25;  this->pictureBox1->TabStop = false;  this->pictureBox1->Visible = false;  this->pictureBox1->Click += gcnew  System::EventHandler(this,  &CourseWorkForm::pictureBox1\_Click);  //  // groupBox1  //  this->groupBox1->BackColor =  System::Drawing::SystemColors::Control;  this->groupBox1->Controls->Add(this->pictureBox1);  this->groupBox1->Location = System::Drawing::Point(8,  287);  this->groupBox1->Name = L"groupBox1";  this->groupBox1->Size = System::Drawing::Size(108, 122);  this->groupBox1->TabIndex = 26;  this->groupBox1->TabStop = false;  this->groupBox1->Text = L"Выдача кофе";  //  // button17  //  this->button17->Location = System::Drawing::Point(301,  115);  this->button17->Name = L"button17";  this->button17->Size = System::Drawing::Size(80, 20);  this->button17->TabIndex = 27;  this->button17->Text = L"Сдача";  this->button17->UseVisualStyleBackColor = true;  this->button17->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button17\_Click);  //  // listView1  //  this->listView1->BackColor =  System::Drawing::SystemColors::ButtonHighlight;  this->listView1->BorderStyle =  System::Windows::Forms::BorderStyle::None;  this->listView1->Location = System::Drawing::Point(6, 12);  this->listView1->Name = L"listView1";  this->listView1->Scrollable = false;  this->listView1->Size = System::Drawing::Size(163, 114);  this->listView1->TabIndex = 40;  this->listView1->UseCompatibleStateImageBehavior = false;  this->listView1->View =  System::Windows::Forms::View::List;  //  // textBox1  //  this->textBox1->BackColor =  System::Drawing::SystemColors::ButtonHighlight;  this->textBox1->Location = System::Drawing::Point(253,  115);  this->textBox1->Name = L"textBox1";  this->textBox1->ReadOnly = true;  this->textBox1->Size = System::Drawing::Size(42, 20);  this->textBox1->TabIndex = 41;  this->textBox1->Text = L"0";  this->textBox1->Click += gcnew System::EventHandler(this,  &CourseWorkForm::textBox1\_Click);  //  // button18  //  this->button18->Location = System::Drawing::Point(301,  86);  this->button18->Name = L"button18";  this->button18->Size = System::Drawing::Size(80, 23);  this->button18->TabIndex = 42;  this->button18->Text = L"Приготовить";  this->button18->UseVisualStyleBackColor = true;  this->button18->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button18\_Click);  //  // timer1  //  this->timer1->Enabled = true;  this->timer1->Interval = 1000;  this->timer1->Tick += gcnew System::EventHandler(this,  &CourseWorkForm::timer1\_Tick);  //  // groupBox2  //  this->groupBox2->BackColor =  System::Drawing::SystemColors::Control;  this->groupBox2->Controls->Add(this->listView1);  this->groupBox2->Location = System::Drawing::Point(8, 4);  this->groupBox2->Name = L"groupBox2";  this->groupBox2->Size = System::Drawing::Size(175, 132);  this->groupBox2->TabIndex = 43;  this->groupBox2->TabStop = false;  //  // progressBar1  //  this->progressBar1->Location = System::Drawing::Point(70,  13);  this->progressBar1->Name = L"progressBar1";  this->progressBar1->Size = System::Drawing::Size(33, 10);  this->progressBar1->TabIndex = 44;  //  // progressBar2  //  this->progressBar2->Location = System::Drawing::Point(70,  30);  this->progressBar2->Name = L"progressBar2";  this->progressBar2->Size = System::Drawing::Size(33, 10);  this->progressBar2->TabIndex = 45;  //  // progressBar3  //  this->progressBar3->Location = System::Drawing::Point(70,  47);  this->progressBar3->Name = L"progressBar3";  this->progressBar3->Size = System::Drawing::Size(33, 10);  this->progressBar3->TabIndex = 46;  //  // progressBar4  //  this->progressBar4->Location = System::Drawing::Point(70,  63);  this->progressBar4->Name = L"progressBar4";  this->progressBar4->Size = System::Drawing::Size(33, 10);  this->progressBar4->Step = 1;  this->progressBar4->TabIndex = 47;  //  // progressBar5  //  this->progressBar5->Location = System::Drawing::Point(70,  97);  this->progressBar5->Name = L"progressBar5";  this->progressBar5->Size = System::Drawing::Size(33, 10);  this->progressBar5->Step = 1;  this->progressBar5->TabIndex = 49;  //  // progressBar6  //  this->progressBar6->Location = System::Drawing::Point(70,  80);  this->progressBar6->Name = L"progressBar6";  this->progressBar6->Size = System::Drawing::Size(33, 10);  this->progressBar6->Step = 1;  this->progressBar6->TabIndex = 48;  //  // label8  //  this->label8->AutoSize = true;  this->label8->Location = System::Drawing::Point(23, 10);  this->label8->Name = L"label8";  this->label8->Size = System::Drawing::Size(45, 13);  this->label8->TabIndex = 50;  this->label8->Text = L"молоко";  //  // label9  //  this->label9->AutoSize = true;  this->label9->Location = System::Drawing::Point(35, 27);  this->label9->Name = L"label9";  this->label9->Size = System::Drawing::Size(33, 13);  this->label9->TabIndex = 51;  this->label9->Text = L"кофе";  //  // label10  //  this->label10->AutoSize = true;  this->label10->Location = System::Drawing::Point(5, 44);  this->label10->Name = L"label10";  this->label10->Size = System::Drawing::Size(63, 13);  this->label10->TabIndex = 52;  this->label10->Text = L"вз. молоко";  //  // label11  //  this->label11->AutoSize = true;  this->label11->Location = System::Drawing::Point(30, 60);  this->label11->Name = L"label11";  this->label11->Size = System::Drawing::Size(38, 13);  this->label11->TabIndex = 53;  this->label11->Text = L"чашки";  //  // label12  //  this->label12->AutoSize = true;  this->label12->Location = System::Drawing::Point(37, 78);  this->label12->Name = L"label12";  this->label12->Size = System::Drawing::Size(31, 13);  this->label12->TabIndex = 54;  this->label12->Text = L"вода";  //  // label13  //  this->label13->AutoSize = true;  this->label13->Location = System::Drawing::Point(32, 94);  this->label13->Name = L"label13";  this->label13->Size = System::Drawing::Size(36, 13);  this->label13->TabIndex = 55;  this->label13->Text = L"сахар";  //  // groupBox3  //  this->groupBox3->Controls->Add(this->button12);  this->groupBox3->Controls->Add(this->button1);  this->groupBox3->Controls->Add(this->button2);  this->groupBox3->Controls->Add(this->button3);  this->groupBox3->Controls->Add(this->button4);  this->groupBox3->Controls->Add(this->button5);  this->groupBox3->Controls->Add(this->button6);  this->groupBox3->Controls->Add(this->button7);  this->groupBox3->Controls->Add(this->button8);  this->groupBox3->Controls->Add(this->button9);  this->groupBox3->Controls->Add(this->button10);  this->groupBox3->Controls->Add(this->button11);  this->groupBox3->Controls->Add(this->button14);  this->groupBox3->Controls->Add(this->button13);  this->groupBox3->Controls->Add(this->label1);  this->groupBox3->Controls->Add(this->label2);  this->groupBox3->Controls->Add(this->label3);  this->groupBox3->Controls->Add(this->label4);  this->groupBox3->Controls->Add(this->label7);  this->groupBox3->Controls->Add(this->label5);  this->groupBox3->Controls->Add(this->button16);  this->groupBox3->Controls->Add(this->label6);  this->groupBox3->Controls->Add(this->button15);  this->groupBox3->Location = System::Drawing::Point(8,  135);  this->groupBox3->Name = L"groupBox3";  this->groupBox3->Size = System::Drawing::Size(373, 154);  this->groupBox3->TabIndex = 56;  this->groupBox3->TabStop = false;  //  // button19  //  this->button19->Location = System::Drawing::Point(107,  93);  this->button19->Name = L"button19";  this->button19->Size = System::Drawing::Size(74, 23);  this->button19->TabIndex = 57;  this->button19->Text = L"Пополнить";  this->button19->UseVisualStyleBackColor = true;  this->button19->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button19\_Click);  //  // groupBox4  //  this->groupBox4->Controls->Add(this->label18);  this->groupBox4->Controls->Add(this->label19);  this->groupBox4->Controls->Add(this->label20);  this->groupBox4->Controls->Add(this->label21);  this->groupBox4->Controls->Add(this->label16);  this->groupBox4->Controls->Add(this->label17);  this->groupBox4->Controls->Add(this->label15);  this->groupBox4->Controls->Add(this->label14);  this->groupBox4->Controls->Add(this->button21);  this->groupBox4->Controls->Add(this->label9);  this->groupBox4->Controls->Add(this->button19);  this->groupBox4->Controls->Add(this->progressBar1);  this->groupBox4->Controls->Add(this->progressBar2);  this->groupBox4->Controls->Add(this->label13);  this->groupBox4->Controls->Add(this->progressBar3);  this->groupBox4->Controls->Add(this->label12);  this->groupBox4->Controls->Add(this->progressBar4);  this->groupBox4->Controls->Add(this->label11);  this->groupBox4->Controls->Add(this->progressBar6);  this->groupBox4->Controls->Add(this->label10);  this->groupBox4->Controls->Add(this->progressBar5);  this->groupBox4->Controls->Add(this->label8);  this->groupBox4->Location = System::Drawing::Point(122,  288);  this->groupBox4->Name = L"groupBox4";  this->groupBox4->Size = System::Drawing::Size(259, 122);  this->groupBox4->TabIndex = 58;  this->groupBox4->TabStop = false;  this->groupBox4->Visible = false;  //  // label18  //  this->label18->AutoSize = true;  this->label18->Location = System::Drawing::Point(165, 18);  this->label18->Name = L"label18";  this->label18->Size = System::Drawing::Size(41, 13);  this->label18->TabIndex = 66;  this->label18->Text = L"label18";  //  // label19  //  this->label19->AutoSize = true;  this->label19->Location = System::Drawing::Point(165, 35);  this->label19->Name = L"label19";  this->label19->Size = System::Drawing::Size(41, 13);  this->label19->TabIndex = 65;  this->label19->Text = L"label19";  //  // label20  //  this->label20->AutoSize = true;  this->label20->Location = System::Drawing::Point(165, 51);  this->label20->Name = L"label20";  this->label20->Size = System::Drawing::Size(41, 13);  this->label20->TabIndex = 64;  this->label20->Text = L"label20";  //  // label21  //  this->label21->AutoSize = true;  this->label21->Location = System::Drawing::Point(165, 68);  this->label21->Name = L"label21";  this->label21->Size = System::Drawing::Size(41, 13);  this->label21->TabIndex = 63;  this->label21->Text = L"label21";  //  // label16  //  this->label16->AutoSize = true;  this->label16->Location = System::Drawing::Point(118, 18);  this->label16->Name = L"label16";  this->label16->Size = System::Drawing::Size(41, 13);  this->label16->TabIndex = 62;  this->label16->Text = L"label16";  //  // label17  //  this->label17->AutoSize = true;  this->label17->Location = System::Drawing::Point(118, 35);  this->label17->Name = L"label17";  this->label17->Size = System::Drawing::Size(41, 13);  this->label17->TabIndex = 61;  this->label17->Text = L"label17";  //  // label15  //  this->label15->AutoSize = true;  this->label15->Location = System::Drawing::Point(118, 51);  this->label15->Name = L"label15";  this->label15->Size = System::Drawing::Size(41, 13);  this->label15->TabIndex = 60;  this->label15->Text = L"label15";  //  // label14  //  this->label14->AutoSize = true;  this->label14->Location = System::Drawing::Point(118, 68);  this->label14->Name = L"label14";  this->label14->Size = System::Drawing::Size(41, 13);  this->label14->TabIndex = 59;  this->label14->Text = L"label14";  //  // button21  //  this->button21->Location = System::Drawing::Point(179,  93);  this->button21->Name = L"button21";  this->button21->Size = System::Drawing::Size(76, 23);  this->button21->TabIndex = 58;  this->button21->Text = L"Инкасация";  this->button21->UseVisualStyleBackColor = true;  this->button21->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button21\_Click);  //  // button20  //  this->button20->BackgroundImage =  (cli::safe\_cast<System::Drawing::Image^>(resources-  >GetObject(L"button20.BackgroundImage")));  this->button20->BackgroundImageLayout =  System::Windows::Forms::ImageLayout::Zoom;  this->button20->FlatStyle =  System::Windows::Forms::FlatStyle::Popup;  this->button20->Location = System::Drawing::Point(348, 4);  this->button20->Name = L"button20";  this->button20->Size = System::Drawing::Size(33, 31);  this->button20->TabIndex = 59;  this->button20->UseVisualStyleBackColor = true;  this->button20->Click += gcnew System::EventHandler(this,  &CourseWorkForm::button20\_Click);  //  // openFileDialog1  //  this->openFileDialog1->FileName = L"openFileDialog1";  //  // CourseWorkForm  //  this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);  this->AutoScaleMode =  System::Windows::Forms::AutoScaleMode::Font;  this->BackColor = System::Drawing::SystemColors::Control;  this->ClientSize = System::Drawing::Size(390, 417);  this->Controls->Add(this->button20);  this->Controls->Add(this->groupBox4);  this->Controls->Add(this->groupBox3);  this->Controls->Add(this->groupBox2);  this->Controls->Add(this->button18);  this->Controls->Add(this->textBox1);  this->Controls->Add(this->button17);  this->Controls->Add(this->groupBox1);  this->FormBorderStyle =  System::Windows::Forms::FormBorderStyle::FixedSingle;  this->Name = L"CourseWorkForm";  this->Text = L"Кофейный Автомат";  this->Load += gcnew System::EventHandler(this,  &CourseWorkForm::CourseWorkForm\_Load);  (cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this-  >pictureBox1))->EndInit();  this->groupBox1->ResumeLayout(false);  this->groupBox2->ResumeLayout(false);  this->groupBox3->ResumeLayout(false);  this->groupBox3->PerformLayout();  this->groupBox4->ResumeLayout(false);  this->groupBox4->PerformLayout();  this->ResumeLayout(false);  this->PerformLayout();  }  #pragma endregion  private: System::Void CourseWorkForm\_Load(System::Object^ sender,  System::EventArgs^ e)  {  listView1->Items->Insert(0, "Добро пожаловать");  listView1->Items->Insert(1, "Внесите деньги:");  listView1->Items->Insert(2, " ");  listView1->Items->Insert(3, " ");  listView1->Items->Insert(4, " ");  listView1->Items->Insert(5, " ");  if (!coffee\_machine.check\_resourсes() ||  !coffee\_machine.check\_money())  {  button17->Enabled = false;  groupBox3->Enabled = false;  listView1->Items->RemoveAt(0);  listView1->Items->Insert(0, "Аппарат временно не  работает");  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, " ");  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, " ");  type\_coffee = 0;  sugar = -1;  }  }  //Кнопка сдачи//////////////////////////////////////////  private: System::Void button17\_Click(System::Object^ sender,  System::EventArgs^ e)  {  listView1->Items->Clear();  listView1->Items->Insert(0, "Добро пожаловать");  listView1->Items->Insert(1, "Внесите деньги:");  listView1->Items->Insert(2, " ");  listView1->Items->Insert(3, " ");  listView1->Items->Insert(4, " ");  listView1->Items->Insert(5, " ");  textBox1->Text = coffee\_machine.return\_credit().ToString();  type\_coffee = 0;  sugar = -1;  if (!coffee\_machine.check\_money())  {  button17->Enabled = false;  groupBox3->Enabled = false;  listView1->Items->RemoveAt(0);  listView1->Items->Insert(0, "Аппарат временно не  работает");  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, " ");  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, " ");  type\_coffee = 0;  sugar = -1;  }  }  //Забрать кофе////////////////////////////////////////  private: System::Void pictureBox1\_Click(System::Object^ sender,  System::EventArgs^ e)  {  pictureBox1->Visible = false;  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  }  // кнопки с деньгами///////////////////////////////////  private: System::Void button8\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.add\_credit(4);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  if (type\_coffee == 0)  {  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе:");  }  }  private: System::Void button7\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.add\_credit(5);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  if (type\_coffee == 0)  {  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе:");  }  }  private: System::Void button6\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.add\_credit(6);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  if (type\_coffee == 0)  {  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе:");  }  }  private: System::Void button5\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.add\_credit(7);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  if (type\_coffee == 0)  {  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе:");  }  }  private: System::Void button4\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.add\_credit(3);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  if (type\_coffee == 0)  {  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе:");  }  }  private: System::Void button3\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.add\_credit(2);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  if (type\_coffee == 0)  {  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе:");  }  }    private: System::Void button1\_Click(System::Object^ sender,0  System::EventArgs^ e)  {  coffee\_machine.add\_credit(0);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  if (type\_coffee == 0)  {  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе:");  }  }  private: System::Void button2\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.add\_credit(1);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  if (type\_coffee == 0)  {  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе:");  }  }  // Кнопки с кофе////////////////////////////////////////  private: System::Void button12\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (coffee\_machine.get\_credit() >= 50)  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе: АМЕРИКАНО");  type\_coffee = 1;  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Внесите деньги!");  }  if (sugar == -1 && type\_coffee != 0)  {  sugar = 0;  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, "Выберите колл-во сахара: " +  sugar.ToString());  }  }  private: System::Void button13\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (coffee\_machine.get\_credit() >= 70)  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе: ДОПЛИО");  type\_coffee = 3;  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Внесите деньги!");  }  if (sugar == -1 && type\_coffee != 0)  {  sugar = 0;  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, "Выберите колл-во сахара: " +  sugar.ToString());  }  }  private: System::Void button14\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (coffee\_machine.get\_credit() >= 50)  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе: ЭСПРЕССО");  type\_coffee = 2;  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Внесите деньги!");  }  if (sugar == -1 && type\_coffee != 0)  {  sugar = 0;  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, "Выберите колл-во сахара: " +  sugar.ToString());  }  }  private: System::Void button10\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (coffee\_machine.get\_credit() >= 90)  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе: КАПУЧИНО");  type\_coffee = 6;  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Внесите деньги!");  }  if (sugar == -1 && type\_coffee != 0)  {  sugar = 0;  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, "Выберите колл-во сахара: " +  sugar.ToString());  }  }  private: System::Void button9\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (coffee\_machine.get\_credit() >= 100)  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе: МАККИАТО");  type\_coffee = 5;  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Внесите деньги!");  }  if (sugar == -1 && type\_coffee != 0)  {  sugar = 0;  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, "Выберите колл-во сахара: " +  sugar.ToString());  }  }  private: System::Void button11\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (coffee\_machine.get\_credit() >= 80)  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, "Выберите кофе: ЛАТТЕ");  type\_coffee = 4;  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Внесите деньги!");  }  if (sugar == -1 && type\_coffee != 0)  {  sugar = 0;  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, "Выберите колл-во сахара: " +  sugar.ToString());  }  }  //Кнопки сахара/////////////////////////////////////////  private: System::Void button15\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (type\_coffee != 0)  {  if (sugar != -1 && type\_coffee != 0 && sugar < 5)  {  sugar++;  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, "Выберите колл-во  сахара: " + sugar.ToString());  }  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Выбирите кофе");  }  }  private: System::Void button16\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (type\_coffee != 0)  {  if (sugar != -1 && type\_coffee != 0 && sugar > 0)  {  sugar--;  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, "Выберите колл-во  сахара: " + sugar.ToString());  }  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Выберите кофе");  }  }  ////////////////////////////////////////////////////////  private: System::Void timer1\_Tick(System::Object^ sender,  System::EventArgs^ e)  {  progressBar1->Value = coffee\_machine.get\_resource(0);  progressBar6->Value = coffee\_machine.get\_resource(1);  progressBar2->Value = coffee\_machine.get\_resource(2);  progressBar4->Value = coffee\_machine.get\_resource(3);  progressBar5->Value = coffee\_machine.get\_resource(4);  progressBar3->Value = coffee\_machine.get\_resource(5);  label14->Text = "1 - " + coffee\_machine.get\_money(0).ToString();  label15->Text = "2 - " + coffee\_machine.get\_money(1).ToString();  label17->Text = "5 - " + coffee\_machine.get\_money(2).ToString();  label16->Text = "10 - " + coffee\_machine.get\_money(3).ToString();  label21->Text = "50 - " + coffee\_machine.get\_money(4).ToString();  label20->Text = "100 - " + coffee\_machine.get\_money(5).ToString();  label19->Text = "500 - " + coffee\_machine.get\_money(6).ToString();  label18->Text = "1000 - " + coffee\_machine.get\_money(7)  .ToString();  }  //Кнопка приготовить////////////////////////////////////  private: System::Void button18\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (type\_coffee != 0 && sugar != -1 && coffee\_machine.get\_credit()  >= coffee\_machine.get\_price(type\_coffee))  {  if (pictureBox1->Visible == false)  {  coffee\_machine.buy\_coffee(coffee\_machine.get\_price(type\_coffee));  coffee\_machine.make\_cup\_coffee(type\_coffee,  sugar);  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Кредит: " +  coffee\_machine.get\_credit().ToString());  pictureBox1->Visible = true;  }  else  {  listView1->Items->RemoveAt(5);  listView1->Items->Insert(5, "Заберите кофе");  }  }  if (coffee\_machine.get\_credit() == 0)  {  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Внесите деньги: ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, " ");  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, " ");  listView1->Items->RemoveAt(4);  listView1->Items->Insert(4, " ");  type\_coffee = 0;  sugar = -1;  }  if (!coffee\_machine.check\_resourсes())  {  button17->Enabled = false;  groupBox3->Enabled = false;  listView1->Items->RemoveAt(0);  listView1->Items->Insert(0, "Аппарат временно не  работает");  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, " ");  listView1->Items->RemoveAt(2);  listView1->Items->Insert(2, " ");  listView1->Items->RemoveAt(3);  listView1->Items->Insert(3, " ");  textBox1->Text =  coffee\_machine.return\_credit().ToString();  type\_coffee = 0;  sugar = -1;  }  }  //Кнопка сдачи//////////////////////////////////////////  private: System::Void textBox1\_Click(System::Object^ sender,  System::EventArgs^ e)  {  textBox1->Text = "0";  }  //Кнопка пополнения ресурсов////////////////////////////  private: System::Void button19\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.replenishment\_resourсe();  if (coffee\_machine.check\_money())  {  groupBox3->Enabled = true;  button17->Enabled = true;  listView1->Items->RemoveAt(0);  listView1->Items->Insert(0, "Добро пожаловать");  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Внесите деньги:");  }  }  private: System::Void button20\_Click(System::Object^ sender,  System::EventArgs^ e)  {  if (groupBox4->Visible) groupBox4->Visible = false;  else groupBox4->Visible = true;  }  private: System::Void button21\_Click(System::Object^ sender,  System::EventArgs^ e)  {  coffee\_machine.incasate();  if (coffee\_machine.check\_resourсes())  {  groupBox3->Enabled = true;  button17->Enabled = true;  listView1->Items->RemoveAt(0);  listView1->Items->Insert(0, "Добро пожаловать");  listView1->Items->RemoveAt(1);  listView1->Items->Insert(1, "Внесите деньги:");  }  }  };  } |

Листинг 2. Исходный код модуля CourseWorkForm.h

|  |
| --- |
| #include "Bank.h"  #include "iostream"  // Реализация конструктора  Bank::Bank()  {  \_credit = 0;  ifstream file;  file.open("profit.txt");  if (file.is\_open())  {  for (int i = 0; i < 8; i++)  {  Money money;  try  {  if (file.peek() == EOF) throw 404;  int buffer;  file >> buffer;  money.set\_count(buffer);  file >> buffer;  money.set\_nominal(buffer);  this->\_money[i] = money;  }  catch(int throw1)  {  money.set\_count(10);  if (i == 0) money.set\_nominal(1);  if (i == 1) money.set\_nominal(2);  if (i == 2) money.set\_nominal(5);  if (i == 3) money.set\_nominal(10);  if (i == 4) money.set\_nominal(50);  if (i == 5) money.set\_nominal(100);  if (i == 6) money.set\_nominal(500);  if (i == 7) money.set\_nominal(1000);  this->\_money[i] = money;  }  }  }  file.close();  }  // Реализация деструктора  Bank::~Bank()  {  ofstream file;  file.open("profit.txt");  if (file.is\_open())  {  for (int i = 0; i < 8; i++)  {  file << \_money[i].get\_count() << " " <<  \_money[i].get\_nominal() << endl;  }  }  file.close();  }  // Реализация метода возврата сдачи  int Bank::return\_credit()  {  int credit = \_credit;  while (\_credit != 0)  {  if (\_credit >= \_money[7].get\_nominal())  {  \_credit -= \_money[7].get\_nominal();  \_money[7].set\_count(\_money[7].get\_count() - 1);  }  else  {  if (\_credit >= \_money[6].get\_nominal())  {  \_credit -= \_money[6].get\_nominal();  \_money[6].set\_count(\_money[6].get\_count() - 1);  }  else  {  if (\_credit >= \_money[5].get\_nominal())  {  \_credit -= \_money[5].get\_nominal();  \_money[5].set\_count(\_money[5].get\_count()  - 1);  }  else  {  if (\_credit >= \_money[4].get\_nominal())  {  \_credit -=  \_money[4].get\_nominal();  \_money[4].set\_count(\_money[4]  .get\_count() - 1);  }  else  {  if (\_credit >=  \_money[3].get\_nominal())  {  \_credit -= \_money[3]  .get\_nominal();  \_money[3].set\_count  (\_money[3].get\_count() - 1);  }  else  {  if (\_credit >=  \_money[2].get\_nominal())  {  \_credit -= \_money[2]  .get\_nominal();  \_money[2].set\_count  (\_money[2].get\_count()  - 1);  }  else  {  if (\_credit >= \_money[1]  .get\_nominal())  {  \_credit -= \_money[1]  .get\_nominal();  \_money[1].set\_count  (\_money[1].get\_count() –  1);  }  else  {  if (\_credit >= \_money[0]  .get\_nominal())  {  \_credit -= \_money[0]  .get\_nominal();  \_money[0].set\_count  (\_money[0].get\_count()  - 1);  }  }  }  }  }  }  }  }  }  return credit;  }  // Реализация метода проверки кассы на наличие бакнот и монет  bool Bank::check\_money()  {  for (int i = 0; i < 8; i++)  {  if (\_money[i].get\_count() < 5) return false;  }  return true;  }  // Реализация метода добавление денег к кредиту  void Bank::add\_credit(int index)  {  \_money[index].set\_count(\_money[index].get\_count() + 1);  \_credit += \_money[index].get\_nominal();  }  // Реализация метода возвращения колличества купюр или монет  int Bank::get\_money(int index)  {  return \_money[index].get\_count();  }  // Реализация метода взвращения кредита  int Bank::get\_credit()  {  return \_credit;  }  // Реализация метода инкасации  void Bank::incasate()  {  for (int i = 0; i < 8; i++) \_money[i].set\_count(50);  }  // Реализация метода пкупки кофе  void Bank::buy\_coffee(int price)  {  \_credit = \_credit - price;  } |

Листинг 3. Исходный код модуля Bank.cpp

|  |
| --- |
| #pragma once  #include "Money.h"  #include "fstream"  using namespace std;  class Bank  {  private:  int \_credit;  Money \_money[8];  public:  // Конструктор  Bank();  // Деструктор  ~Bank();  // Метод возвращения кредита  int return\_credit();  // Метод проверки наличия купюр и монет  bool check\_money();  // Метод добавления денег к кредиту  void add\_credit(int);  // Метод получения кол-ва денег  int get\_money(int);  // Метод получения credit  int get\_credit();  // Метод инкасации  void incasate();  // Метод покупки кофе  void buy\_coffee(int);  }; |

Листинг 4. Исходный код модуля Bank.h

|  |
| --- |
| #include "Coffee\_machine.h"  #include "iostream"  #include "string"  using namespace System;  // Функция преобразования String^ d string  void String\_a(System::String ^ s, string& os)  {  using namespace Runtime::InteropServices;  const char\* chars = (const  char\*)(Marshal::StringToHGlobalAnsi(s)).ToPointer();  os = chars;  Marshal::FreeHGlobal(IntPtr((void\*)chars));  }  // Реализация конструктора  Coffee\_machine::Coffee\_machine()  {  ifstream file;  file.open("resourses.txt");  if (file.is\_open())  for (int i = 0; i < 6; i++)  {  Consumables resourсe;  try  {  int buffer;  string name;  file >> buffer;  getline(file, name);  resourсe.set\_count(buffer);  resourсe.set\_name(name);  \_resourсes[i] = resourсe;  }  catch(int throw1)  {  if (i = 0)  {  resourсe.set\_count(10);  resourсe.set\_name("milk");  \_resourсes[i] = resourсe;  }  if (i = 1)  {  resourсe.set\_count(10);  resourсe.set\_name("water");  \_resourсes[i] = resourсe;  }  if (i = 2)  {  resourсe.set\_count(10);  resourсe.set\_name("coffee");  \_resourсes[i] = resourсe;  }  if (i = 3)  {  resourсe.set\_count(10);  resourсe.set\_name("cup");  \_resourсes[i] = resourсe;  }  if (i = 4)  {  resourсe.set\_count(10);  resourсe.set\_name("sugar");  \_resourсes[i] = resourсe;  }  if (i = 5)  {  resourсe.set\_count(10);  resourсe.set\_name("whipped\_milk");  \_resourсes[i] = resourсe;  }  }  }  file.close();  }  // Реализация деструктора  Coffee\_machine::~Coffee\_machine()  {  ofstream file;  string str;  file.open("resourses.txt");  if (file.is\_open())  {  for (int i = 0; i < 6; i++)  {  String\_a(\_resourсes[i].get\_count().ToString(), str);  file << str << \_resourсes[i].get\_name() << endl;  }  }  file.close();  }  // Реализация метода получения цены кофе по индексу  int Coffee\_machine::get\_price(int type\_coffee)  {  int price;  switch (type\_coffee)  {  case 1:  price = 50;  break;  case 2:  price = 50;  break;  case 3:  price = 70;  break;  case 4:  price = 80;  break;  case 5:  price = 100;  break;  case 6:  price = 90;  break;  default:  price = 0;  break;  }  return price;  }  // Реализация метода получения кол-ва ресурсов по индексу  int Coffee\_machine::get\_resource(int index)  {  return \_resourсes[index].get\_count();  }  // Реализация метода проверки наличия ресурсов  bool Coffee\_machine::check\_resourсes()  {  for (int i = 0; i < 6; i++)  if (\_resourсes[i].get\_count() < 5) return false;  return true;  }  // Реализация метода пополнения ресурсов  void Coffee\_machine::replenishment\_resourсe()  {  for (int i = 0; i < 6; i++) \_resourсes[i].set\_count(100);  }  // Реализация метода приготовления кофе  void Coffee\_machine::make\_cup\_coffee(int type\_coffee, int sugar)  {  switch (type\_coffee)  {  // Американо  case 1:  \_resourсes[0].take\_resourсe(0);  \_resourсes[1].take\_resourсe(3);  \_resourсes[2].take\_resourсe(1);  \_resourсes[3].take\_resourсe(1);  \_resourсes[4].take\_resourсe(sugar);  \_resourсes[5].take\_resourсe(0);  break;  // Эспрессо  case 2:  \_resourсes[0].take\_resourсe(0);  \_resourсes[1].take\_resourсe(0);  \_resourсes[2].take\_resourсe(1);  \_resourсes[3].take\_resourсe(1);  \_resourсes[4].take\_resourсe(sugar);  \_resourсes[5].take\_resourсe(0);  break;  // Доплио  case 3:  \_resourсes[0].take\_resourсe(0);  \_resourсes[1].take\_resourсe(0);  \_resourсes[2].take\_resourсe(2);  \_resourсes[3].take\_resourсe(1);  \_resourсes[4].take\_resourсe(sugar);  \_resourсes[5].take\_resourсe(0);  break;  // Латте  case 4:  \_resourсes[0].take\_resourсe(2);  \_resourсes[1].take\_resourсe(0);  \_resourсes[2].take\_resourсe(1);  \_resourсes[3].take\_resourсe(1);  \_resourсes[4].take\_resourсe(sugar);  \_resourсes[5].take\_resourсe(1);  break;  // Макиато  case 5:  \_resourсes[0].take\_resourсe(0);  \_resourсes[1].take\_resourсe(0);  \_resourсes[2].take\_resourсe(2);  \_resourсes[3].take\_resourсe(1);  \_resourсes[4].take\_resourсe(sugar);  \_resourсes[5].take\_resourсe(1);  break;  // Капучино  case 6:  \_resourсes[0].take\_resourсe(1);  \_resourсes[1].take\_resourсe(0);  \_resourсes[2].take\_resourсe(1);  \_resourсes[3].take\_resourсe(1);  \_resourсes[4].take\_resourсe(sugar);  \_resourсes[5].take\_resourсe(2);  break;  default:  break;  }  }  // Реализация метода выдачи сдачи  int Coffee\_machine::return\_credit()  {  return \_bank.return\_credit();  }  // Реализация метода проверки наличия купюр и монет  bool Coffee\_machine::check\_money()  {  return \_bank.check\_money();  }  // Реализация метода добавления денег к кедиту  void Coffee\_machine::add\_credit(int count)  {  \_bank.add\_credit(count);  }  // Реализация метода проверки кол-во купюр и монет  int Coffee\_machine::get\_money(int index)  {  return \_bank.get\_money(index);  }  // Реализация метода получения кредита  int Coffee\_machine::get\_credit()  {  return \_bank.get\_credit();  }  // Реализация метода инкасации  void Coffee\_machine::incasate()  {  \_bank.incasate();  }  // Реализация метода покупки кофе  void Coffee\_machine::buy\_coffee(int price)  {  \_bank.buy\_coffee(price);  } |

Листинг 5. Исходный код модуля Coffee\_machine.cpp

|  |
| --- |
| #pragma once  #include "Consumables.h"  #include "Bank.h"  #include "iostream"  #include "fstream"  #include "string"  using namespace std;  class Coffee\_machine  {  private:  Bank \_bank;  Consumables \_resourсes[6];  public:  // Конструктор по умолчению  Coffee\_machine();  // Деструктор  ~Coffee\_machine();  // Метод получения цены кофе  int get\_price(int);  // Метод получения кол-ва ресурсов по индексу  int get\_resource(int);  // Метод проверки ресурсов  bool check\_resourсes();  // Метод Пополнения ресурсов  void replenishment\_resourсe();  // Метод приготовления кофе  void make\_cup\_coffee(int, int);  // Метод выдачи сдачи  int return\_credit();  // Метод проверки наличия есурсов  bool check\_money();  // Метод добавления денег к кредиту  void add\_credit(int);    // Метод получения значения money  int get\_money(int);  // Метод получения значения credit  int get\_credit();  // Метод инкасации  void incasate();  // Метод покупки кофе  void buy\_coffee(int);  }; |

Листинг 6. Исходный код модуля Coffee\_machine.h

|  |
| --- |
| #include "Consumables.h"  // Реализация конструктора по умолчанию  Consumables::Consumables()  {  \_count = 0;  \_name = "";  }  // Реализация конструктора с параметрами  Consumables::Consumables(int count, string name)  {  this->\_count = count;  this->\_name = name;  }  // Реализация деструктора  Consumables::~Consumables()  {  }  // Реализация метода установки значения count  void Consumables::set\_count(int count)  {  this->\_count = count;  }  // Реализация метода установки значения name  void Consumables::set\_name(string name)  {  this->\_name = name;  }  // Реализация метода получения значения count  int Consumables::get\_count()  {  return \_count;  }  // Реализация метода получения значения name  string Consumables::get\_name()  {  return \_name;  }  // Реализация метода взятия ресурса  void Consumables::take\_resourсe(int count)  {  this->\_count -= count;  } |

Листинг 7. Исходный код модуля Consumables.cpp

|  |
| --- |
| #pragma once  #include "iostream"  using namespace std;  class Consumables  {  private:  int \_count;  string \_name;  public:  // Конструктор по умолчанию  Consumables();  // Конструктор с параметрами  Consumables(int, string);  // Деструктор  ~Consumables();  // Метод устанавки значения count  void set\_count(int);  // Метод установки значения name  void set\_name(string);  // Метод получения значения count  int get\_count();  // Метод получения значения name  string get\_name();  // Метод взятия ресурса  void take\_resourсe(int);  }; |

Листинг 8. Исходный код модуля Consumables.h

|  |
| --- |
| #include "Money.h"  // Реализация конструктора по умолчанию  Money::Money()  {  \_count = 0;  \_nominal = 0;  }  // Реализация конструктора с параметрами  Money::Money(int count, int nominal)  {  this->\_count = count;  this->\_nominal = nominal;  }  // Реализация деструктора  Money::~Money()  {  }  // Реализация метода возвращения count  int Money::get\_count()  {  return \_count;  }  // Реализация метода установки значения count  void Money::set\_count(int count)  {  this->\_count = count;  }  // Реализация метода возвращения nominal  int Money::get\_nominal()  {  return \_nominal;  }  // Реализация метода установки значения nominal  void Money::set\_nominal(int nominal)  {  this->\_nominal = nominal;  } |

Листинг 9. Исходный код модуля Money.cpp

|  |
| --- |
| #pragma once  class Money  {  private:  int \_count;  int \_nominal;  public:  // Конструктор по умолчанию  Money();  // Конструктор с параметрами  Money(int, int);  // Деструктор  ~Money();  // Метод возвращения count  int get\_count();  // Метод установки значения count  void set\_count(int);    // Метод возвращения nominal  int get\_nominal();  // метод установки значения nominal  void set\_nominal(int);  }; |

Листинг 10. Исходный код модуля