|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**КУРСОВАЯ РАБОТА**

по дисциплине «Объектно-ориентрованное программирование»

**Тема курсовой работы: «**Проектирование и реализация программной системы с использованием объектного подхода. Разработка автомата по продаже кофе»

**Студент группы** ИКБО-06-17 Политов Александр Юрьевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель курсовой работы** ст. преподаватель Гусев К.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена к защите «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Допущен к защите «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Москва 2018

|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»** **РТУ МИРЭА** |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Утверждаю

Заведующий кафедрой МОСИТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Головин С.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2018 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы по дисциплине**

«Объектно-ориентрованное программирование»

Студент Политов Александр Юрьевич Группа ИКБО-06-17

**Тема работы:** Проектирование и реализация программной системы с использованием объектного подхода

**Исходные данные:** задание на курсовую работу,тема №1, вариант №2. Разработать систему работы автомата по продаже кофе. Функционал системы: приготовление кофе по заказу покупателя, после поступления нужной суммы и выдача сдачи, имеющимися купюрами (и/или монетами), выдача заказа.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

Моделирование объектной системы с использованием обозначений языка моделирования UML

Реализация программной системы средствами языка С++

Отчет по курсовой работе в виде расчетно-пояснительной записки .

**Срок представления к защите курсовой работы:** до «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

**Задание на курсовую работу выдал** \_\_\_\_\_\_\_\_\_\_\_\_ (Гусев К.В.)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

**Задание на курсовую работу получил**  \_\_\_\_\_\_\_\_\_\_\_\_ (Политов А.Ю.)

ОГЛАВЛЕНИЕ

[**КУРСОВАЯ РАБОТА** 1](#_Toc515908538)

[**ЗАДАНИЕ** 2](#_Toc515908539)

[**ВВЕДЕНИЕ** 3](#_Toc515908540)

[**ТЕОРИТИЧЕСКОЕ ВВЕДЕНИЕ** 4](#_Toc515908541)

[Объектно-ориентированное программирование 5](#_Toc515908542)

[Модульное программирование 6](#_Toc515908543)

[Работа с внешними файлами 7](#_Toc515908544)

[UML проектирование 8](#_Toc515908545)

[**ПРОЕКТИРОВАНИЕ** 11](#_Toc515908546)

[Спецификации программы 11](#_Toc515908547)

[Список выявленных понятий 12](#_Toc515908548)

[Проектирование программного обеспечения 13](#_Toc515908549)

[Уточнение отношений между классами 13](#_Toc515908550)

[Составление текстового интерфейса классов 13](#_Toc515908551)

[Структура файлов для хранения данных программы 15](#_Toc515908552)

[Реализация графического интерфейса 15](#_Toc515908553)

[**ТЕСТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ** 17](#_Toc515908554)

[Руководство пользователю 17](#_Toc515908555)

[Тестирование 17](#_Toc515908556)

[**ЗАКЛЮЧЕНИЕ** 20](#_Toc515908557)

[**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** 21](#_Toc515908558)

[**ПРИЛОЖЕНИЕ** 21](#_Toc515908559)

# **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.

# **ТЕОРИТИЧЕСКОЕ 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:

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

Модификаторы доступа в C++:

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

* 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:

* Более быстрое редактирование кода, так как программист изменяет только один модуль, не вмешиваясь в код других.
* Более быстрая компиляция, так как можно компилировать модули отдельно.
* 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.

* Физический файл – это файл на внешнем носителе;
* 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;

Связать файл с потоком можно двумя способами:

1. При создании потока:

ifstream if(“file.txt”);

1. С помощью метода open():

if.open(“file.txt”);

Ввод-вывод данных через поток:

1. С помощью оператора >> или <<:

if >> name;

1. С помощью функции getline():

getline(name, if);

Закрытие файлового потока происходит через метод close():

if.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.

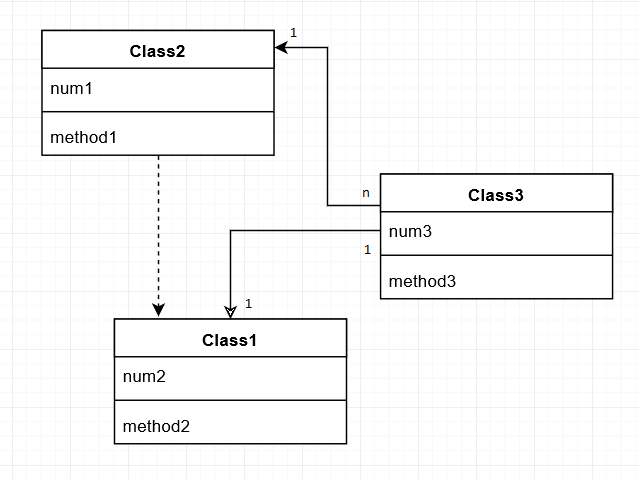


Рисунок 1 - Пример диаграммы классов

Основные элементы диаграммы классов:

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

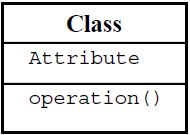


Рисунок 2 - Пиктограмма класса

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

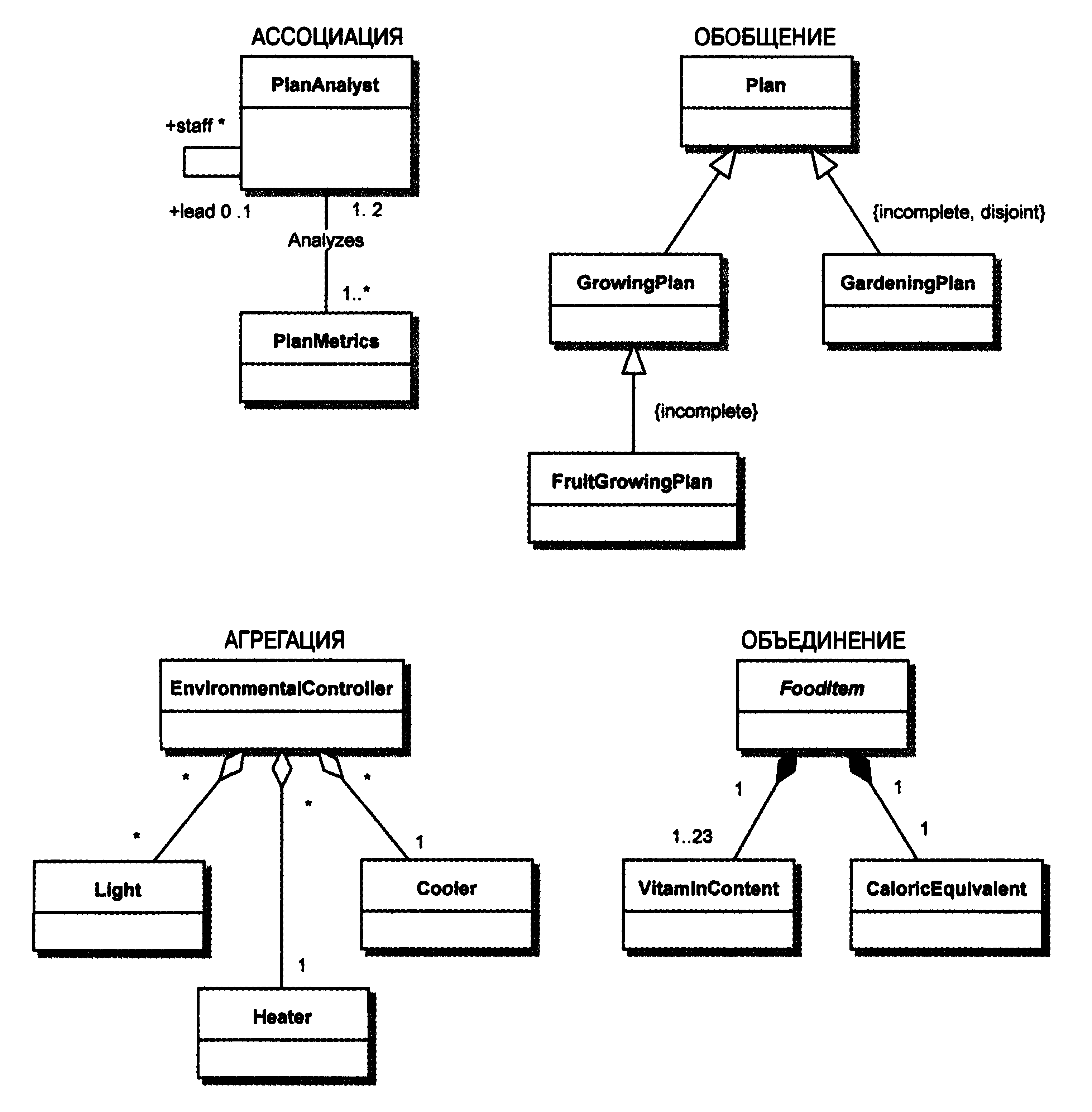


Рисунок 3 - Виды атрибутов

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;
* Классы :

Coffee\_machine – класс кофейного автомата

Сonsumables – класс расходников, представляющий структуру ингредиента

Bank – класс описывающий кассу

Money – класс, описывающий структуру денег;

CourseWorkForm – класс, через который реализуется графическая оболочка 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;

## 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.

Концептуальная модель предметной области, диаграмма классов (Рисунок 4)

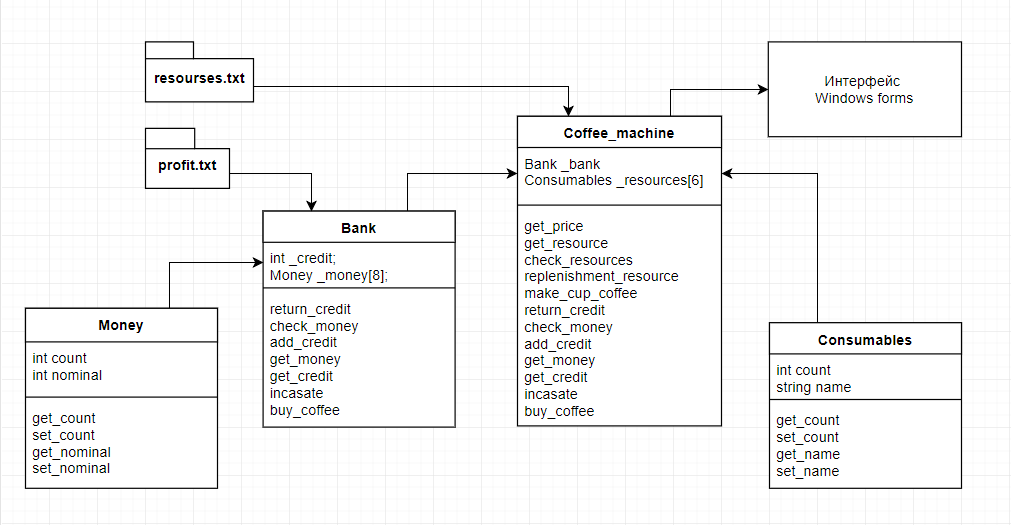


Рисунок 4. Диаграмма классов

## Проектирование программного обеспечения

### 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;

### 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 расходниках и их количестве. (Таблица 2)

Таблица 1 – структура файла profit.txt

|  |  |
| --- | --- |
| Номинал | Количество |
| 1 | 46 |
| 2 | 36 |
| 5 | 26 |
| 10 | 71 |
| 50 | 56 |
| 100 | 45 |
| 500 | 34 |
| 1000 | 23 |

Таблица 2 – 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 выполненана 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 5)

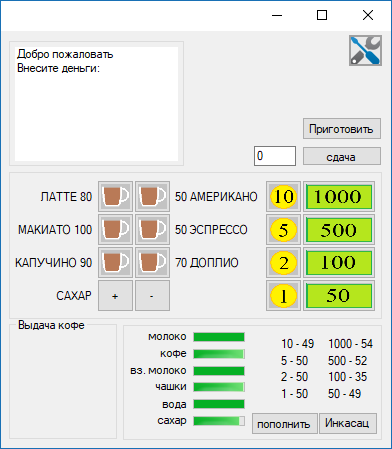


Рисунок 5 – Графический интерфейс программы

# **ТЕСТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ**

## Руководство пользователя

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.

Так же в этом меню можно узнать о текущем состоянии кассы и количестве ингредиентов.

Для пополнения ингредиентов нажать кнопку “Пополнить”

Для инкассации нажать кнопку “Инкассация”

## Тестирование

Все выполненные тесты программной системы приведены в (Таблица 3).

Весь исходный код программной системы представлен в Приложении к курсовой работе.

Таблица 3 – Тестирование программной системы

|  |  |  |
| --- | --- | --- |
| Номер теста | Тест | Комментарий к тесту |
| 1 |  | Покупатель внёс деньги, выбрал напиток и количество сахара, нажал кнопку приготовить. |
| 2 |  | Покупатель внёс сумму в автомат |

Продолжение Таблицы 3

|  |  |  |
| --- | --- | --- |
| 3 |  | Покупатель нажал кнопку выбора кофе перед внесением суммы |
| 4 |  | Нажата кнопка входа в меню управления ресурсами и кассой, также на тесте видно как автомат возвращается в исходное состояние после выдачи напитка и сдачи |

Продолжение Таблицы 3

|  |  |  |
| --- | --- | --- |
| 5 |  | Покупатель внёс деньги, выбрал напиток и количество сахара, нажал кнопку приготовить. |

# **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 была достигнута основная цель: я получил практические навыки по проектированию и разработке объектной системы.

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Бьёрн Страуструп - Программирование: принципы и практика с использованием C++ отдельное издание, пер в англ – Игорь Красиков Издательство “Вильямс” 2016 г. тираж 300 - 1328 стр.
2. Р.Лафоре - Объектно-Ориентированное программирование в C++ Издательство “Питер СПб” 2016г. - 928 стр.
3. Гради Буч, Роберт А. Максимчук - Объектно-ориентированный анализ и проектирование с примерами приложений 3-е издание Издательство “Вильямс” 2008 г. - 720 стр.
4. Михаил Пайсон - ООП с примерами [Электронный ресурс] URL: <https://habr.com/post/87119/> (Дата обращения 20.5.2018)