

Министерство образования Московской области  
ГБПОУ МО «Колледж «Коломна»

09.02.03

КУРСОВОЙ ПРОЕКТ

ПМ. 02 Осуществление интеграции программных модулей  
Тема: Разработка информационной системы «Вендинговая машина»  
ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
ККОО.КП4206.000ПЗ

Студент:	Иванов В.С.
Руководитель:	Карташова Е.В.
Нормоконтролер:	Грушникова Т.Н.
Дата защиты:	Оценка

2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ  
ГБПОУ МО «КОЛЛЕДЖ «КОЛОМНА»  
ЗАДАНИЕ  
на курсовой проект по ПМ 02 Осуществление интеграции программных  
модулей  
студента очной формы обучения 4 курса группы 402-ИСП-18

---

Тема: Разработка информационной системы «Вендинговая машина»

---

Постановка задачи:

Разработать информационную систему в соответствии с темой и приложение для работы с ней. Разработанная информационная система должна отражать основные бизнес-процессы в соответствии с техническим заданием на разработку. Для защиты информации необходимо разграничить права пользователей системы. Разработанные формы должны обеспечить возможность пользователям осуществлять работу с данными в соответствии с их правами.

При разработке информационной системы необходимо решить следующие задачи:

- сформулировать цель проектирования базы данных;
- описать возможного пользователя базы данных;
- определить круг запросов и задач, которые предполагается решать с использованием созданной базы данных;
- построить концептуальную модель;
- сформулировать требования к базе данных;
- построить реляционную модель и выполнить её нормализацию;
- осуществить выбор СУБД и технических средств;
- создать спроектированную базу данных в среде выбранной СУБД;
- разработать приложение для реализации запросов и решения задач;

- оценить ИС с точки зрения возможностей её дальнейшего развития.

Объем курсовой работы:

1) Пояснительная записка (25-50 листов печатного текста формата А4):

Введение

1. Разработка системного проекта
  - 1.1 Назначение разработки
  - 1.2 Требования к программному продукту
2. Обзор используемых технологий
3. Описание системы
  - 3.1 Проектирование базы данных
  - 3.2 Разработка проекта
  - 3.3 Интерфейс программы
4. Руководство пользователя

Заключение

Список литературы

2) Презентация для защиты курсового проекта (10-15 слайдов)

Дата выдачи задания

\_\_\_\_\_

Срок окончания

\_\_\_\_\_

Зав. структурным подразделением

\_\_\_\_\_ Емельянова В. А.

Руководитель

\_\_\_\_\_ Карташова Е.В.

## СОДЕРЖАНИЕ

1	Разработка системного проекта	6
1.1	Назначение разработки	6
1.2	Требования к программному продукту	7
2	Обзор используемых технологий	8
3	Описание системы	9
3.1	Проектирование базы данных	9
3.1.1.	Концептуальная модель	9
3.1.2	Даталогическая модель	11
3.2	Разработка проекта	13
3.2.1	Проектирование программного обеспечения	13
3.2.2	Структура проекта	14
3.2.3	Интерфейс программы	19
3.2.4	Обработка данных	24
4	Руководство пользователя	27
	Заключение	30
	Список литературы	31

					<b>ККОО.КП4206.000ПЗ</b>		
<b>Изм.</b>	<b>Лист</b>	<b>№ докум.</b>	<b>Подпись</b>	<b>Дата</b>			
Разраб.		Иванов			Курсовой проект Пояснительная записка	Лит.	Лист
Провер.		Карташова					25
Реценз.						Гр 402-ИСП-18	
Н. Контр.		Грушникова					
Утверд.							

Курсовое проектирование является завершающим этапом изучения программного модуля «Осуществление интеграции программных модулей», в ходе которого осуществляется обучение применению полученных знаний и умений при решении комплексных задач, связанных с профессиональной деятельностью будущих специалистов.

Целью курсового проекта является:

- углубления теоретических знаний в соответствии с заданной темой;
- формирования умений применять теоретические знания при решении поставленных вопросов;
- формирования умений использовать справочную, нормативную и правовую документацию;
- применение полученных знаний во время изучения программного модуля «Осуществление интеграции программных модулей»;
- демонстрация знаний, появившихся в процессе самостоятельного обучения технологий разработки приложений;
- демонстрация умений, необходимых для работы с системой контроля версий «Git»;
- научиться распределять время разработки приложения.

Завершением курсового проекта является рабочее, полностью функционирующее приложение, предназначенное для обеспечения должного функционирования клиент – приложения учёта заказов и всех напитков в аппарате. Приложение должно соответствовать всем требованиям и правилам разработки.

					ККОО.КП4206.000ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

# 1 РАЗРАБОТКА СИСТЕМНОГО ПРОЕКТА

## 1.1 Назначение разработки

Разработать информационную систему, предназначенную для имитирования работы вендинговой машины (аппарата для выдачи напитков, далее - аппарат).

Система разрабатывается для персонала, обслуживающего аппарат, а также для клиентов с целью покупки товара. Она позволит контролировать количество напитков и монет в машине, выполнять заказ клиентов, а также получать необходимую отчетность.

Система предназначена для четырех типов пользователей:

- клиент;
- администратор;

Система будет использоваться как настольное приложение (будет эксплуатироваться на офисных компьютерах).

					ККОО.КП4206.000ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

## 1.2 Требования к программному продукту

В системе будет четыре различных типа пользователей: Клиенты, Администраторы. Для клиентов не требуется проходить процедуру авторизации – покупка товаров доступна для них мгновенно. Для администраторов требуется проходить процедуру авторизации в случае, если они хотят попасть в панель управления аппаратом.

Программа должна учитывать следующие требования:

- 1) Клиент должен иметь возможность видеть свою корзину (выбранные товары), управлять ею.
- 2) Корзина должна группировать одинаковые напитки, в случае если клиент добавляет напитков по несколько штук.
- 3) Администратор имеет возможность зайти в панель администратора по специальному коду, который является индивидуальным для каждого аппарата.
- 4) Панель администратора предполагает:
  - возможность блокирования\пополнения монет в аппарате. Заблокированные монеты нельзя использовать в качестве оплаты клиентом заказа, однако их может использовать аппарат в качестве сдачи;
  - возможность добавления\удаления\изменения напитков и их изображений, а также их количества в аппарате;
  - формирование отчётности по каждому из напитков с учётом динамики изменения его цены.
- 5) При загрузке данных с удалённого сервера аппарат должен блокировать интерфейс и отображать характерную обложку (иконку с эффектом вращения на затемнённом фоне).

					ККОО.КП4206.000ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

## 2 ОБЗОР ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ

В курсовой работе были использованы следующие технологии:

- Window Presentation Foundation (WPF) – аналог Windows Forms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .Net Framework, использующая язык XAML.

- Language-Integrated Query (LINQ) – проект Microsoft по добавлению синтаксиса языка запросов, напоминающего SQL, в языки программирования платформы .Net Framework. Ранее был реализован в языках C# и др.

- ASP.NET – платформа разработки веб-приложений, в состав которой входят: веб-сервисы, программная инфраструктура, модель программирования, от компании Майкрософт.

- MS SQL – система управления реляционными базами данных, разработанная корпорацией Microsoft. Основным используемым языком запросов – Transact-SQL, создан совместно Microsoft и Sybase.

Приложение было разработано на языке C# – объектно-ориентированный язык со строгой типизацией.

Объектно-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса

					ККОО.КП4206.000ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		



### 3 ОПИСАНИЕ СИСТЕМЫ

#### 3.1 Проектирование базы данных

##### 3.1.1. Концептуальная модель

В процессе проектирования были выявлены следующие сущности:

Каждый вендинговый аппарат имеет следующие параметры:

- Идентификатор аппарата
- Зашифрованный код для входа в панель администратора

Монеты характеризуются следующими параметрами:

- Идентификатор монеты
- Номинал

Напитки получили следующие поля:

- Идентификатор напитка
- Название
- Изображение
- Цена

У каждой напитка и монеты есть привязки к определённому аппарату, поэтому стоит различать следующие сущности:

Напитки аппарата характеризуются следующими параметрами:

- Идентификатор напитка аппарата
- Идентификатор аппарата
- Идентификатор напитка
- Количество

Монеты аппарата имеют параметры:

- Идентификатор монеты аппарата
- Идентификатор аппарата
- Идентификатор монеты

					ККОО.КП4206.000ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

- Количество
- Активная монета (true/false)

В результате действий с напитками аппарата и покупки товаров клиентом в аппарате фиксируется история продаж, для чего потребуется сущность «история напитков аппарата», которая имеет следующие параметры:

- Идентификатор продажи
- Идентификатор аппарата
- Идентификатор напитка
- Количество
- Цена
- Тип
- Дата и время операции

Данная сущность имеет тип, который показывает характер операции:

- Покупка (какое количество напитков куплено и в какую цену)
- Изменение цены (без изменения количества напитков в аппарате)
- Пополнение количества напитков (с возможным изменением цены

напитка)

В результате проектирования была получена следующая диаграмма «Сущность-связь» (рисунок 1)

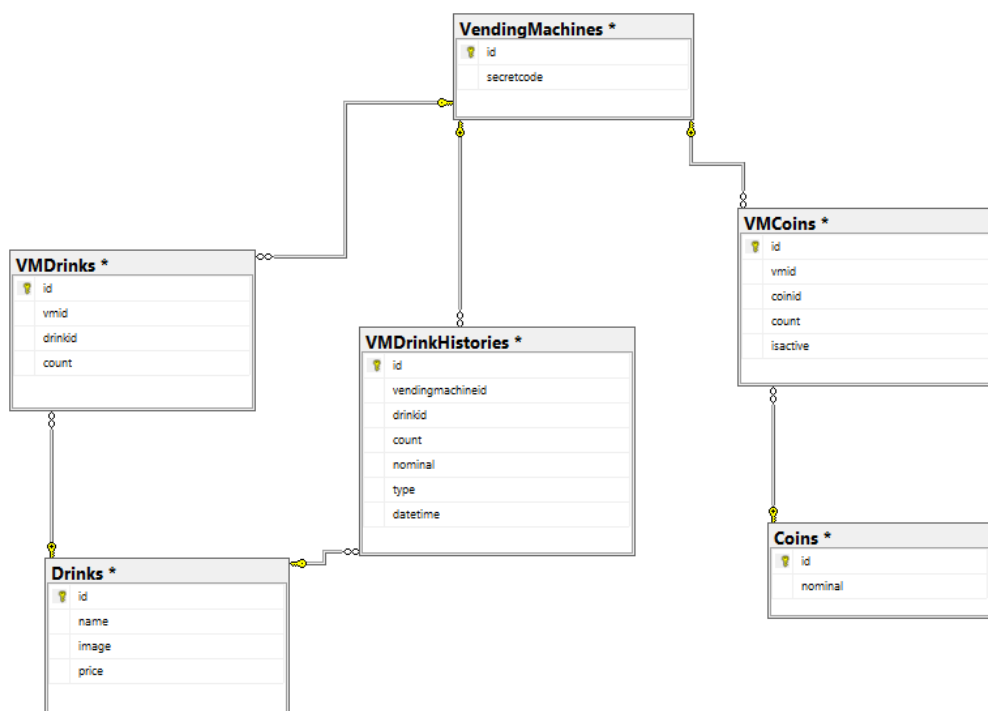


Рисунок 1 - Концептуальная модель

### 3.1.2 Даталогическая модель

Таблица 1 – Вендинговые машины (VendingMachines)

№	Наименование поля	Тип	Ключи	Примечание
1	id	uniqueidentifier	PK	Not Null
2	secretcode	nvarchar(MAX)		Not Null

Таблица 2 – Напитки (Drinks)

№	Наименование поля	Тип	Ключи	Примечание
1	id	uniqueidentifier	PK	Not Null
2	name	varchar(MAX)		Not Null
3	image	varbinary(MAX)		Not Null
4	price	int		Not Null

Таблица 3 – Монеты (Coins)

№	Наименование поля	Тип	Ключи	Примечание
1	id	uniqueidentifier	PK	Not Null
2	nominal	int		Not Null

Таблица 4 – Напитки аппарата (VMDrinks)

№	Наименование поля	Тип	Ключи	Примечание
1	id	uniqueidentifier	PK	Not Null
2	vmid	uniqueidentifier	FK	Not Null
3	drinkid	uniqueidentifier	FK	Not Null
4	count	int		Not Null

Таблица 5 – Монеты аппарата (VMCoins)

№	Наименование поля	Тип	Ключи	Примечание
1	id	uniqueidentifier	PK	Not Null
2	vmid	uniqueidentifier	FK	Not Null
3	coinid	uniqueidentifier	FK	Not Null
4	count	int		Not Null
5	isactive	bit		Not Null

Таблица 6 – Операции с напитками аппарата (VMDrinksHistory)

№	Наименование поля	Тип	Ключи	Примечание
1	id	uniqueidentifier	PK	Not Null
2	vendingmachineid	uniqueidentifier	FK	Not Null
3	drinkid	uniqueidentifier	FK	Not Null
4	count	int		Not Null
5	nominal	int		Not Null
6	type	int		Not Null

### Продолжение таблицы 6

№	Наименование поля	Тип	Ключи	Примечание
7	datetime	datetime		Not Null

## 3.2 Разработка проекта

### 3.2.1 Проектирование программного обеспечения

В системе существуют 2 вида пользователя, которые должны осуществлять различные действия. Необходимо предусмотреть соответствующие экраны для их исполнения.



Рисунок 2 - Диаграмма использования

Последовательность работы с ПО отображена на диаграмме последовательности действий (рисунок 3)

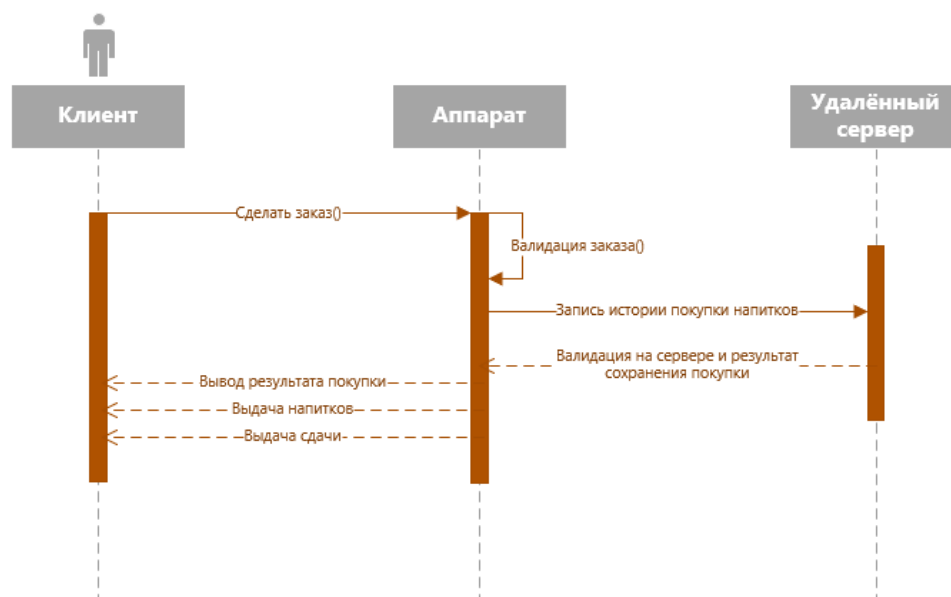


Рисунок 3 - Диаграмма последовательности

### 3.2.2 Структура проекта

Проект состоит из четырёх важных составляющих:

- VendingMachine.API;
- VendingMachine.Domain;
- VendingMachine.EntitiesCore;
- VendingMachine.UI.

Перед описанием каждой из составляющих необходимо описать определённую иерархию моделей в приложении VendingMachine. Можно выделить три разновидности моделей (для примера будет взята выдуманная сущность Object):

- Object является доменной моделью (Domain), модель называется без всяких префиксов и постфиксов. Предназначена для вывода информации, отправки данных, как ответа сервера.
- ObjectBlank относится к Blank-модели, которые являются своего рода «черновиком». Это означает, что модель была заполнена пользователем

и нуждается в немедленной проверке всех введенных данных. При выявленном нарушении сервер должен вернуть ошибку с объяснением своего решения.

– ObjectDb относится к Db-модели, которая предназначена для работы с базой данных. Для любого рода действия с БД требуется получение именно этой модели. Например: для сохранения сущности в БД, ObjectBlank сначала подвергается проверке, затем при её успешном результате, модель конвертируется в ObjectDb и работает логика сохранения в базу данных.

Проект VendingMachine.API имитирует работы удалённого сервера. Включает в себя контроллеры, которые являются точками входа на сервер. Данный проект связывает клиента (клиент любого вида, которые отправит нужные данные на нужную точку входа) с репозиторием (работа с БД). На данном этапе происходит валидация данных. Точки входа в контроллеры перечислены в таблице 7.

Таблица 7 – Точки входа в контроллеры сервера

Название метода	Путь от хоста	Действие
VendingMachinesController		
GetVendingMachine	VendingMachines/GetById	Получение вендинговой машины по идентификатору
Login	VendingMachines/Login	Вход в аппарат по секретному коду
DrinksController		
SaveDrink	Drinks/Save	Сохранение напитка
SaveDrinks	Drinks/SaveDrinks	Сохранение напитков
GetAllDrinks	Drinks/GetAll	Получение всех напитков
DeleteDrink	Drinks/Delete	Удаление напитка
GetDrinkReports	Drinks/GetReports	Получение отчётов

Продолжение таблицы 7

Название метода	Путь от хоста	Действие
CoinsController		
SaveCoins	Coins/Save	Сохранение монет
GetCoins	Coins/GetAll	Получение всех монет

Проект VendingMachine.Domain содержит в себе модели двух типов – Domain и Blank. Также в проекте могут содержаться различные перечисления, которые используются для разработки бизнес-логики.

Проект VendingMachine.EntitiesCore необходим для работы с БД. Содержит в себе все Db-модели данных, которые в свою очередь определяют модель контекста нашей БД.

Проект VendingMachine.UI реализован при помощи принципа SWA (Single Window Application) – приложение с единым окном. По этому принципу у приложения есть одно главное окно, которое предоставляет функционал для изменения текущего интерфейса. При этом, модальные окна для добавления\редактирования не запрещены, так как они необходимы для мелкозадачных действий.

Таким образом, в проекте существует главное окно (BaseWindow), которое содержит в себе пространство для расположения страниц, а также иконку загрузки, которая появляется только при загрузке данных из удалённого сервера.

Страницы приложения:

– AdminPage – экран администратора, содержит три вкладки (монеты, напитки, отчёт). Вход на данную страницу доступен только с помощью секретного кода;

– SalePage – экран покупки. Доступен сразу после запуска приложения, представляет функционал для формирования и оплаты корзины.

Другие окна приложения:

					ККОО.КП4206.000ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		



– AuthorizeWindow – окно, предназначенное для ввода секретного кода аппарата. Проверка кода происходит не локально, а на удалённом сервере, что исключает ситуацию со взломом.

– DrinkEditor – окно-редактор для сущности «Напиток». Представляет функционал для добавления\редактирования напитков, а также изменения количества напитков в автомате.

Структура проекта представлена на рисунке 4.

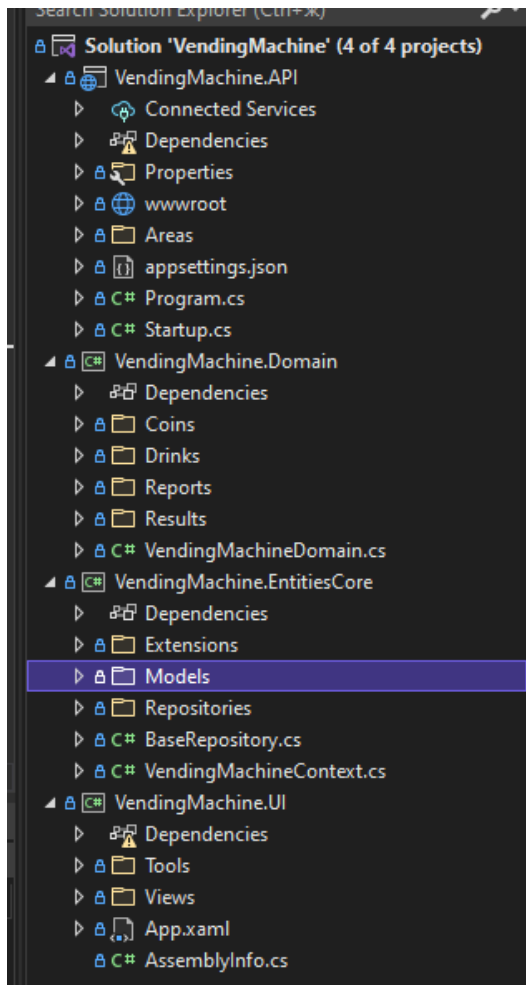


Рисунок 4 – Структура проекта

Каждая папка в проекте имеет продуманное название и своё специальное предназначение. Папки служат логическими границами для классов проекта. Название каждой из них полностью отражает классы, которые находятся в них.

Описание папок и их назначение описаны в таблице 8.

Таблица 8 – Описание папок проекта

Папка	Назначение
VendingMachine.API	
Areas	Содержит в себе контроллеры (каждый из которых можно назвать Area – зона)
Properties	Содержит в себе ключевые настройки ASP.NET проекта, например, параметры запуска проекта и начальный URL
wwwroot	Содержит в себе файлы контента для отображения клиенту, такие как JS-файлы, файлы CSS-стилей, различные библиотеки и иконки для сайта
VendingMachine.Domain	
Coins	Содержит в себе модели для сущности «Монета»
Drinks	Содержит в себе модели для сущности «Напиток»
Reports	Содержит в себе модель истории действия с напитком, а также модель отчёта для напитка
Results	Содержит в себе модели для сущности «Результат», которая показывает успешность действия

Продолжение таблицы 8

Папка	Назначение
VendingMachine.EntitiesCore	
Extensions	Содержит расширения для классов контекста, а также расширение для шифрования строк
Models	Содержит Db-модели для всех сущностей, контекст данных для связи с БД, а также конвертеры моделей в другие.
Repositories	Содержит интерфейсы и классы репозиториев, которые соответственно наследуются. Реализация с интерфейсами необходима для работы технологии DI и IoC-контейнера
VendingMachine.UI	
Tools	Содержит в себе классы-помощники, такие как клиенты для HTTP-запросов и блокирования интерфейса
Views	Содержит в себе все окна и страницы приложения

### 3.2.3 Интерфейс программы

При разработке приложения были созданы следующие формы, представленные на рисунках с 5 по 13.

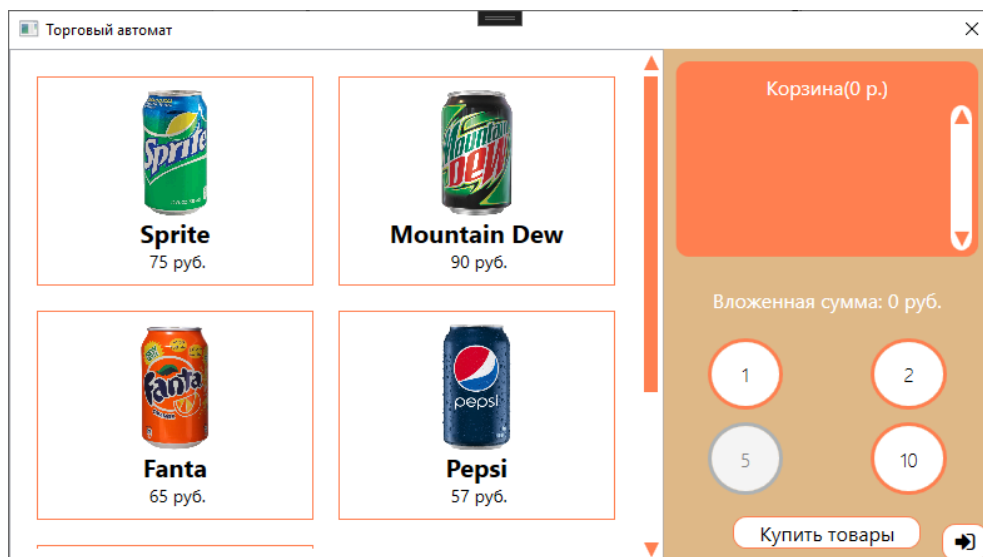


Рисунок 5 – Экран покупки напитков

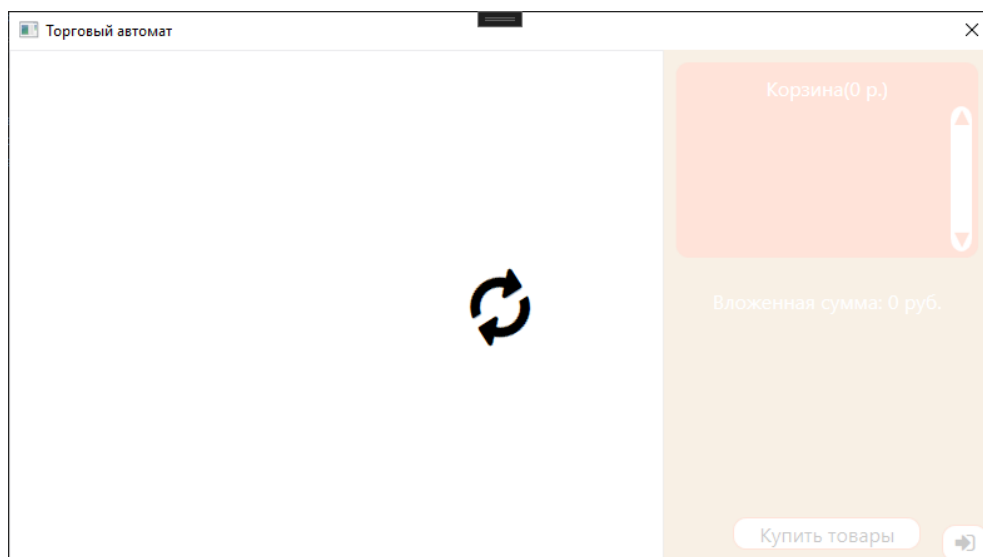


Рисунок 6 – Блокирование интерфейса при загрузке данных

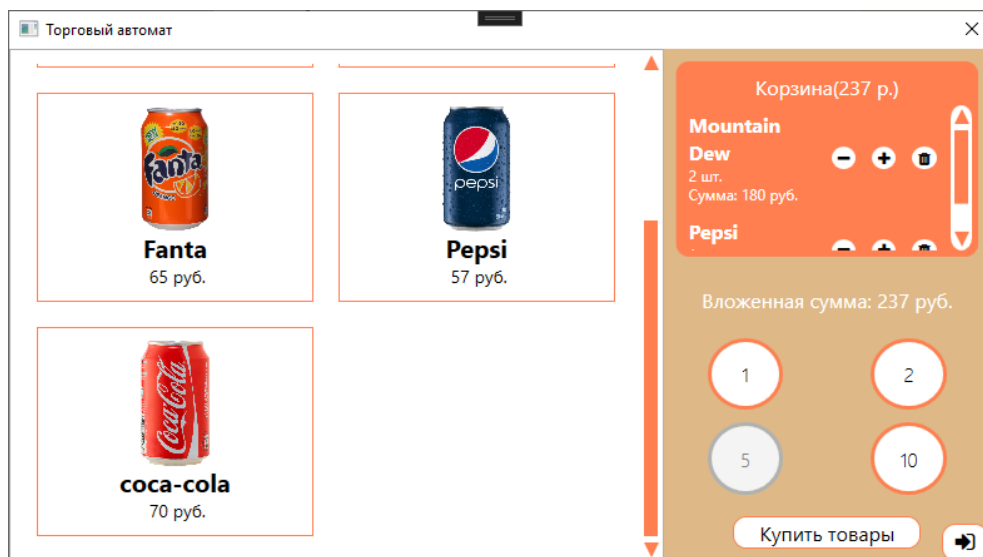


Рисунок 7 – Заполнение корзины и внесение денег

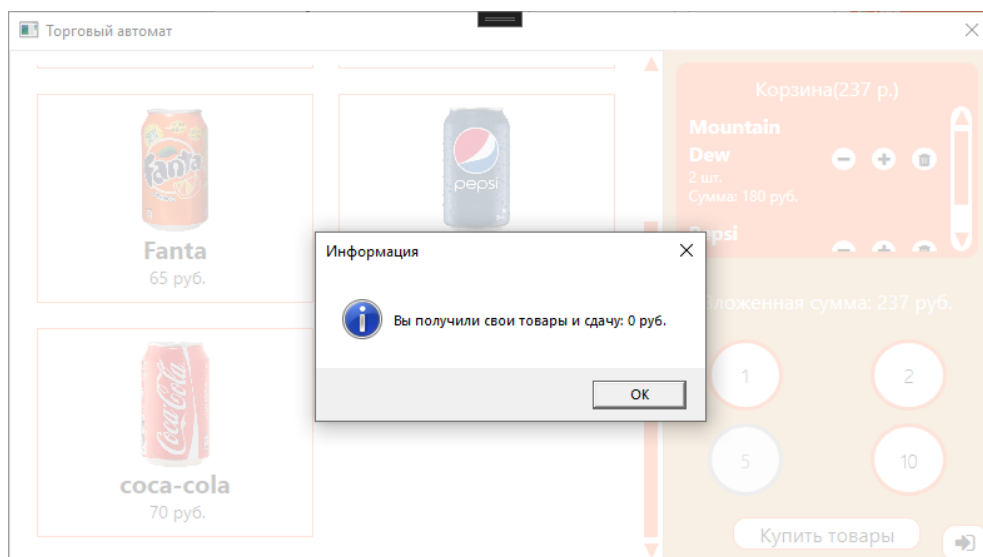


Рисунок 8 – Совершение покупки

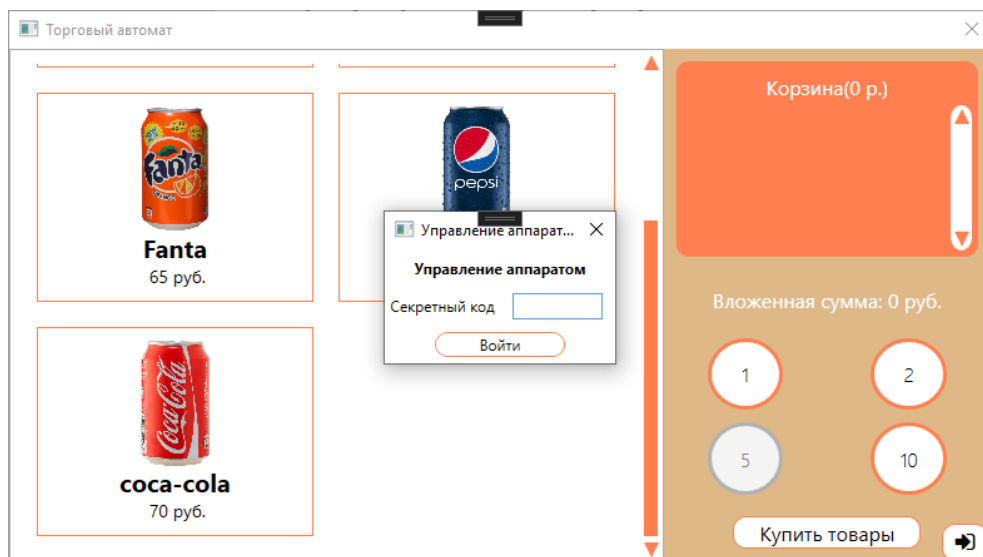


Рисунок 9 – Окно входа в панель администратора



Рисунок 10 – Панель администратора (вкладка –монеты)

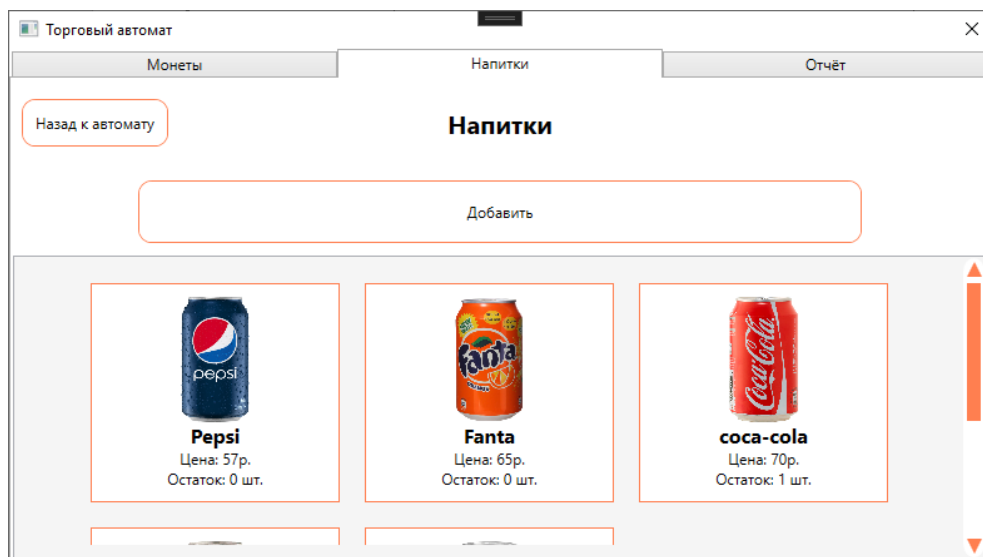


Рисунок 11 – Панель администратора (вкладка –напитки)

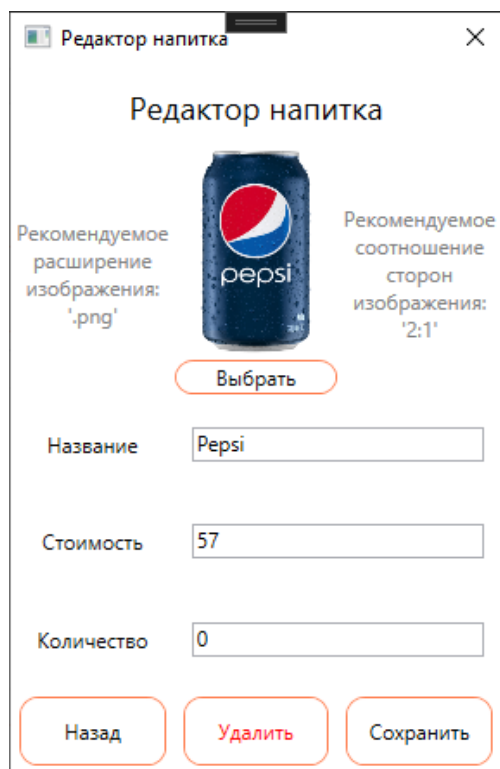


Рисунок 52 – Редактор для напитков

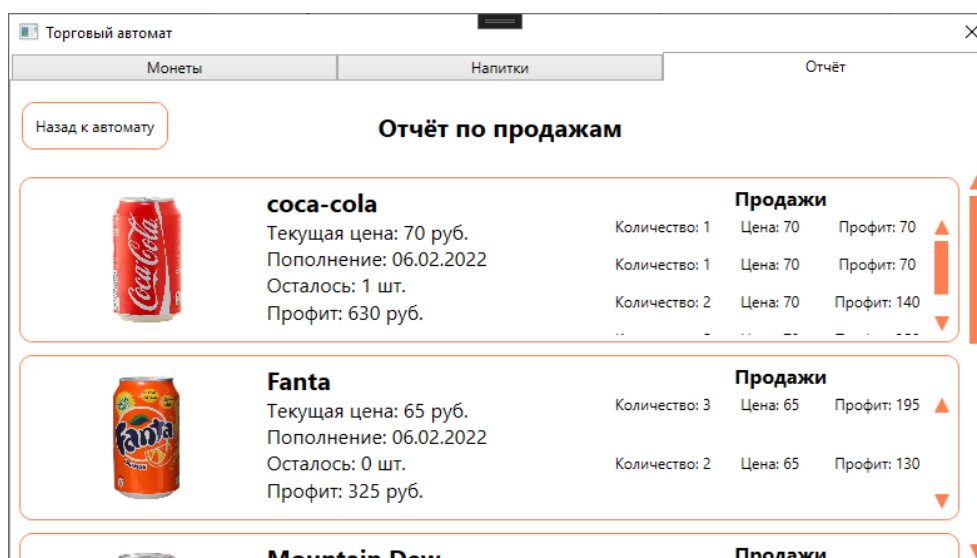


Рисунок 63 – Панель администратора (вкладка – отчёт)

### 3.2.4 Обработка данных

Обработка данных и иные операции с данными происходят посредством общения аппарата и удалённого сервера. При любом запросе на удалённый сервер на аппарате включатся режим «Freeze», который блокирует весь интерфейс и отображает соответствующую иконку.

Удалённый сервер принимает данные, проводит валидацию, завершает необходимые операции и может вернуть ответ с ошибкой, если такова произошла, либо успешный результат. Также результатом удалённого сервера может быть и массив каких-то данных.

Для работы с данными БД использовалась технология LINQ to Objects, которая облегчает работу, в случае, если был реализован подход Model First Entity Framework Core. Данный подход подразумевает собой составления основных сущностей в виде классов приложения, а уж только потом создание в БД.

Примеры запросов LINQ to Objects в БД:

– `context.VMCoins.Where(c => c.VendingMachineId == vendingMachineId).ToArray()` – получение монет определённого аппарата;



- `context.VMDrinks.FirstOrDefault(d => d.Id == vmDrinkBlank.Id.Value)` – поиск существующего напитка аппарата;
- `context.Drinks.Where(d => drinkIds.Contains(d.Id)).ToArray()` – получение напитков, идентификатор которых найден в принятом в массиве идентификаторов.

Ранее рассказывалось про систему из трёх моделей одной сущности (Domain, Blank и Db), из-за чего на этапе разработки появилась потребность в разработке конвертеров, которые способны преобразовать одну модель сущности в другую. В частности, требовались механизмы преобразования из Db-моделей в Domain и из Blank в Db.

Например, возьмём сущность «монета» (Coin):

- 1) Преобразователь из Db-модели в Domain-модель для получения валидной модели из БД для дальнейших операций.

```
public static Coin ToCoin(this CoinDb db)
{
    return new(db.Id, db.Nominal);
}
```

- 2) Преобразователь массива Db-моделей в массив Domain-моделей для получения массива валидных объектов сущности.

```
public static Coin[] ToCoins(this IEnumerable<CoinDb> dbs)
{
    return dbs.Select(ToCoin).ToArray();
}
```

- 3) Преобразователь Blank-модели (черновика) в Db-модель. При этом нам даётся гарантия, что черновик валиден и прошёл полный цикл проверки в контроллере, в который было произведено обращение.

```
public static CoinDb ToCoinDb(this CoinBlank vmCoinBlank)
{
    return new(Guid.NewGuid(), vmCoinBlank.Nominal.Value);
}
```

Данные между проектов API и UI передаются через HTTP-запросы, из-за чего появляется необходимость в правильной передаче моделей. Наиболее распространённый способ передачи информации – JSON-форматы

JSON – текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми.

					ККОО.КП4206.000ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

## 4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Чтобы полноценно запустить аппарат с помощью одного файла, потребуется выложить проект VendingMachine на сервер, чтобы на него могло ссылаться наше приложения. Для того, чтобы можно было запустить приложение аппарата и удалённого сервера одновременно, необходимо запустить решение проекта в Visual Studio, затем кликнуть правой кнопкой мыши, выбрать пункт «Выбрать несколько запускаемых проектов» и выбрать положение переключателя «запуск» у проектов: UI, API.

После этих действий запуск решения произойдёт посредством старта сразу двух проектов на одной машине. После загрузки всех данных откроется следующее окно (рисунок 13).

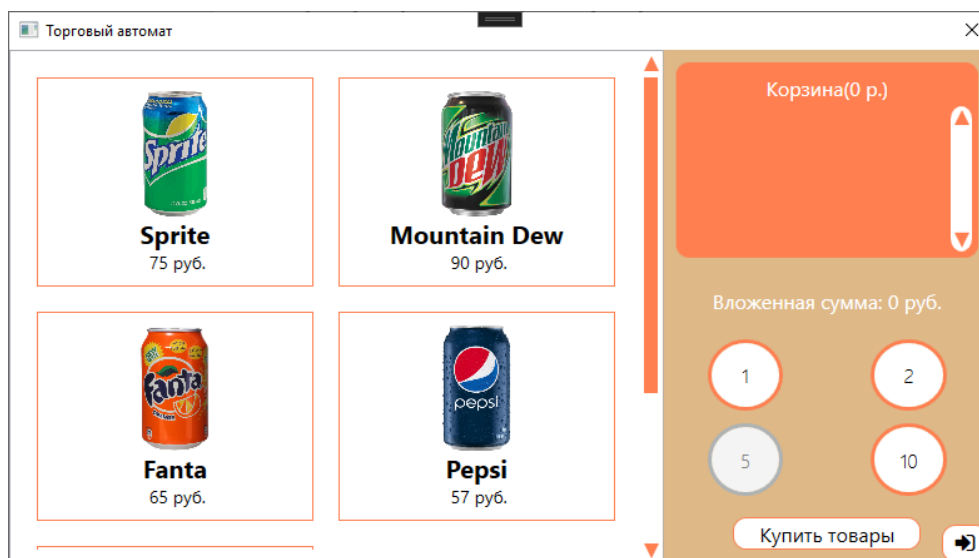


Рисунок 14 – Начальный экран программы

Для выбора напитка необходимо нажать на него, после чего один его экземпляр попадёт в корзину. С монетами ситуация аналогична – при выборе условных монет на аппарате, которые имитируют попадание монет клиента в монетоприёмник. На рисунке 14 был выбран напиток «Mountain Dew» и вкинуто нужное количество монет для покупки, после чего пошли соответствующие последствия.

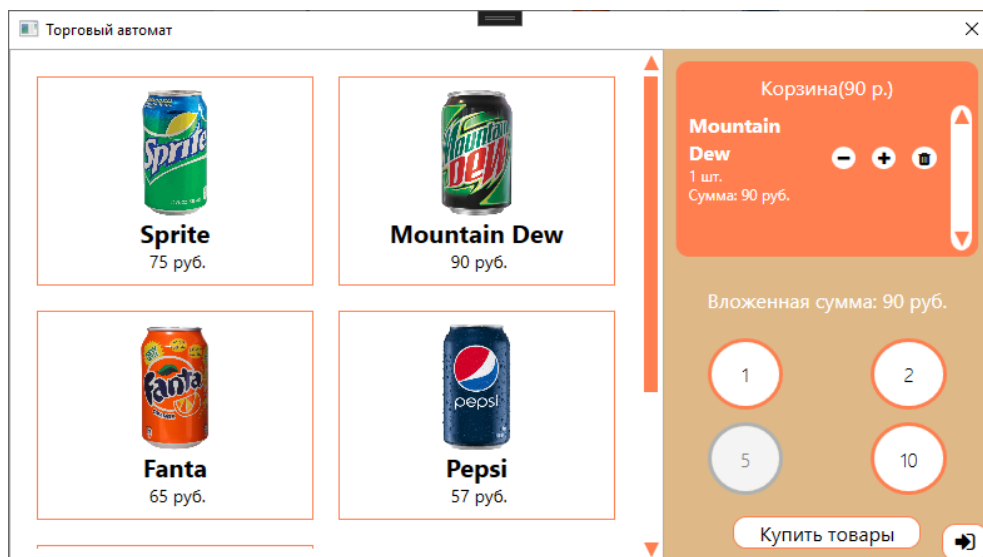


Рисунок 15 – Изменение корзины и кошелька

После нажатия кнопки «Купить товары» произойдёт проверка количества вложенных монет, сумма корзины и доступность напитков в автомате. Если всё пройдет успешно, то выведется соответствующее сообщение, как на рисунке 15.

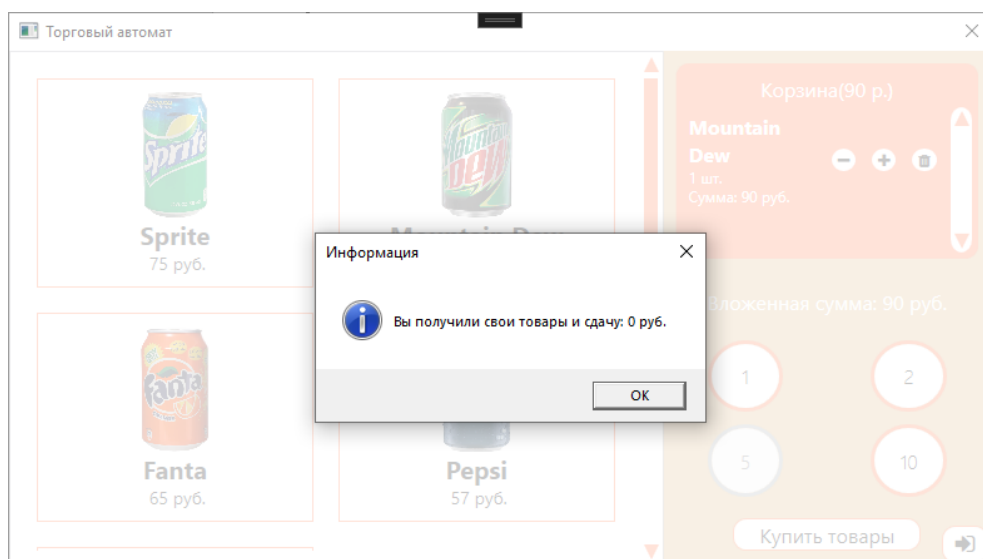


Рисунок 16 – Успешная покупка

В нижнем правом углу имеется кнопка для выхода в панель администратора, при её нажатии у вас появится следующее окно, см. рисунок 16.

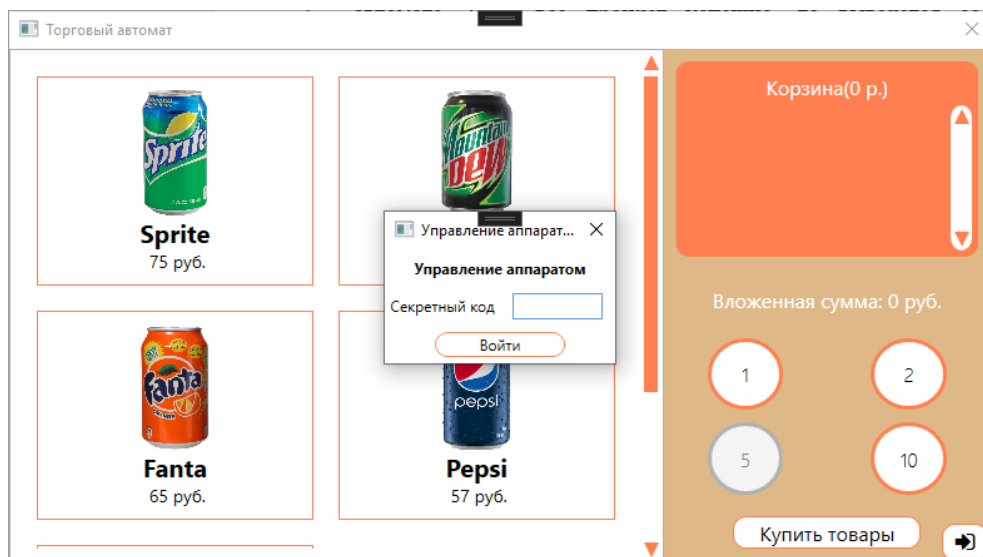


Рисунок 17 – Окно входа в панель администратора

Если вы владеете секретным кодом данного аппарата, то можете ввести его в поле ввода. После проверки кода на удалённом сервере будет понятен результат. Если код прошёл проверку, то вы перейдёте в панель администратора, где можно руководить основными данными в аппарате.

## ЗАКЛЮЧЕНИЕ

В ходе разработки проекта отрабатываются следующие навыки: проектирование интерфейса Windows приложений, программирование на языке C#, Создание запросов на языке SQL-T, Связывание базы данных с приложением с помощью конфигурационного файла, использование параметризованных запросов при формировании запроса обращения к базе данных.

Преимущества приложения является:

- Удобный и понятный для обычного пользователя интерфейс
- Лёгкое управление и учёт данных

В результате создано работоспособное приложение «Вендинговая машина». Проведенное тестирование не выявило существенных ошибок.

					ККОО.КП4206.000ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

## СПИСОК ЛИТЕРАТУРЫ

1) Федорова Г. Н. Разработка, администрирование и защита баз данных: учебник для студ. Учреждений сред. проф. образования / Г.Н. Федорова. – 2-е изд., стер. – М. Петкович Д.

2) Петкович П29 Microsoft® SQL Server™ 2012. Руководство для начинающих: Пер. с англ. — СПб.: БХВ-Петербург, 2013: Издательский центр «Академия», 2018.

3) А. П. Ганенко. Оформление текстовых и графических материалов при подготовке дипломных проектов, курсовых и письменных экзаменационных работ (требования ЕСКД) : учебно-метод. Пособие для студ. Учреждения сред. Проф. Образования / А.П. Ганенко, М. И. Лапсарь. – 9-е изд., стер. – М. : Издательский центр «Академия», 2017.

Электронные ресурсы:

4) MSDN – сеть разработчиков Microsoft [Электронный ресурс] Режим доступа: свободный <https://msdn.microsoft.com/ru-ru/dn308572.aspx> (Дата обращения 15.05.2019)

5) Национальный открытый университет [Электронный ресурс] Режим доступа: свободный <https://www.intuit.ru/> (Дата обращения 15.05.2019)

					ККОО.КП4206.000ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		