Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическая работа №3. Индивидуальное задание.**»

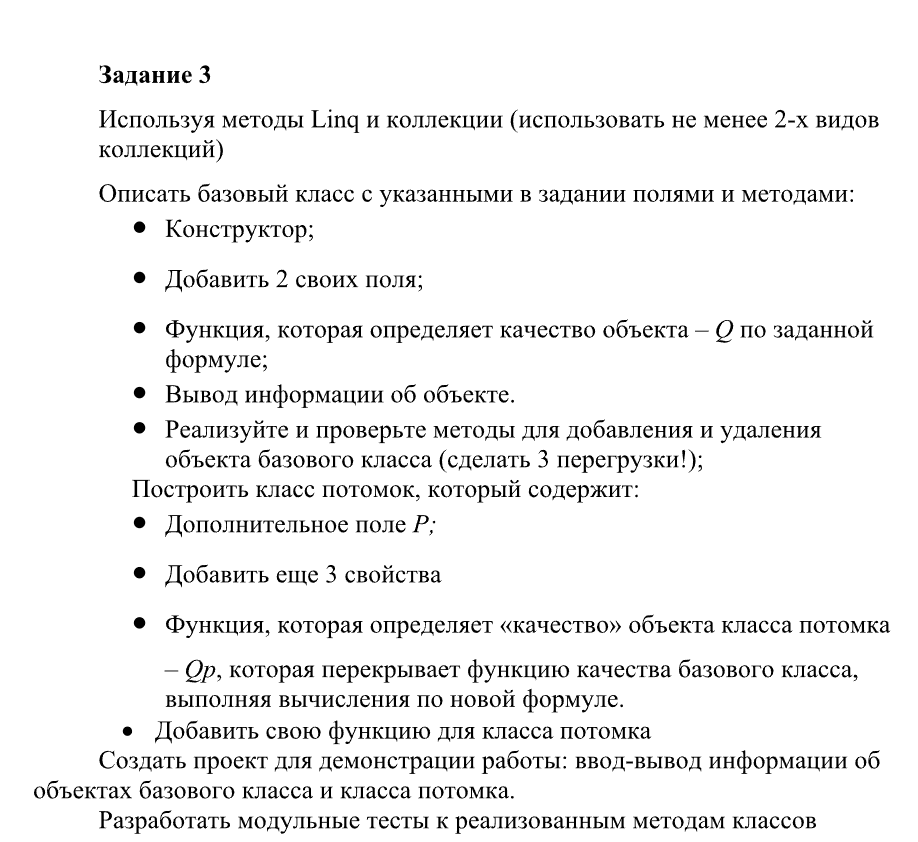
Выполнил: Сергеева Агата Андреевна

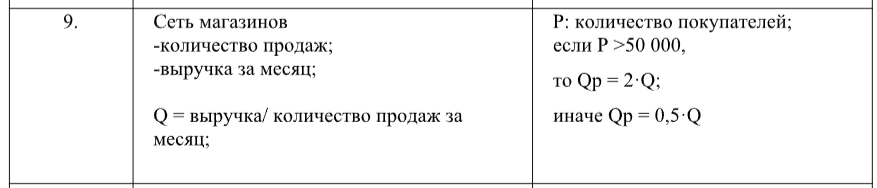
Группа: ПР-31

Преподаватель: Мирошниченко Г.В

2024 г.

**Задание:** (вариант 9)

****

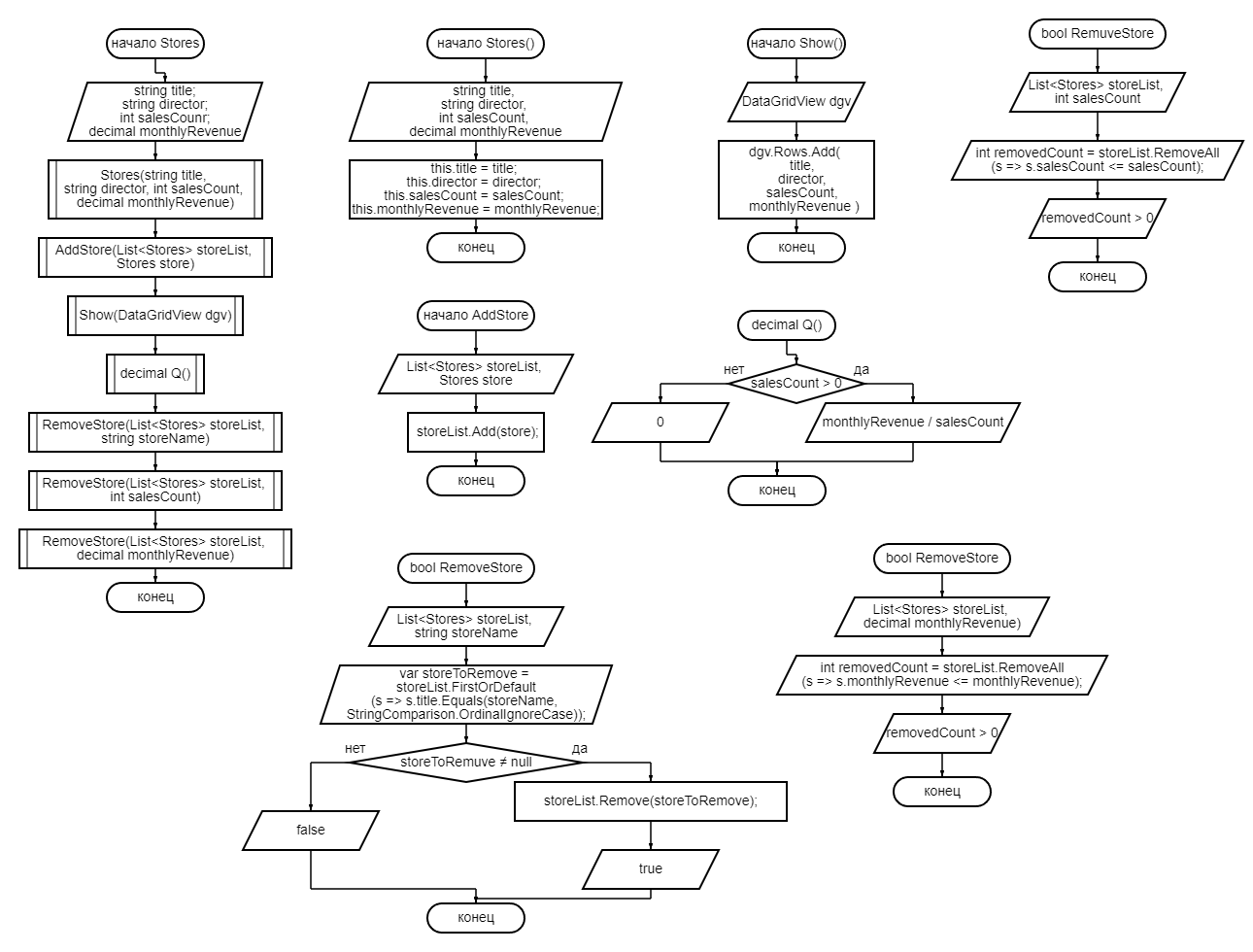
****

**Входные данные:** *title* -название магазина , string; *director* – директор магазина, string; s*alesCount* – количество продаж за месяц (октябрь), int; *monthlyRevenue* – выручка за месяц (октябрь), decimal, *p* – количество посетителей, int; *rating* – рейтинг (оценка),string; *address* – адрес магазина, string; *isOnline* – физический или онлайн-магазин, bool.

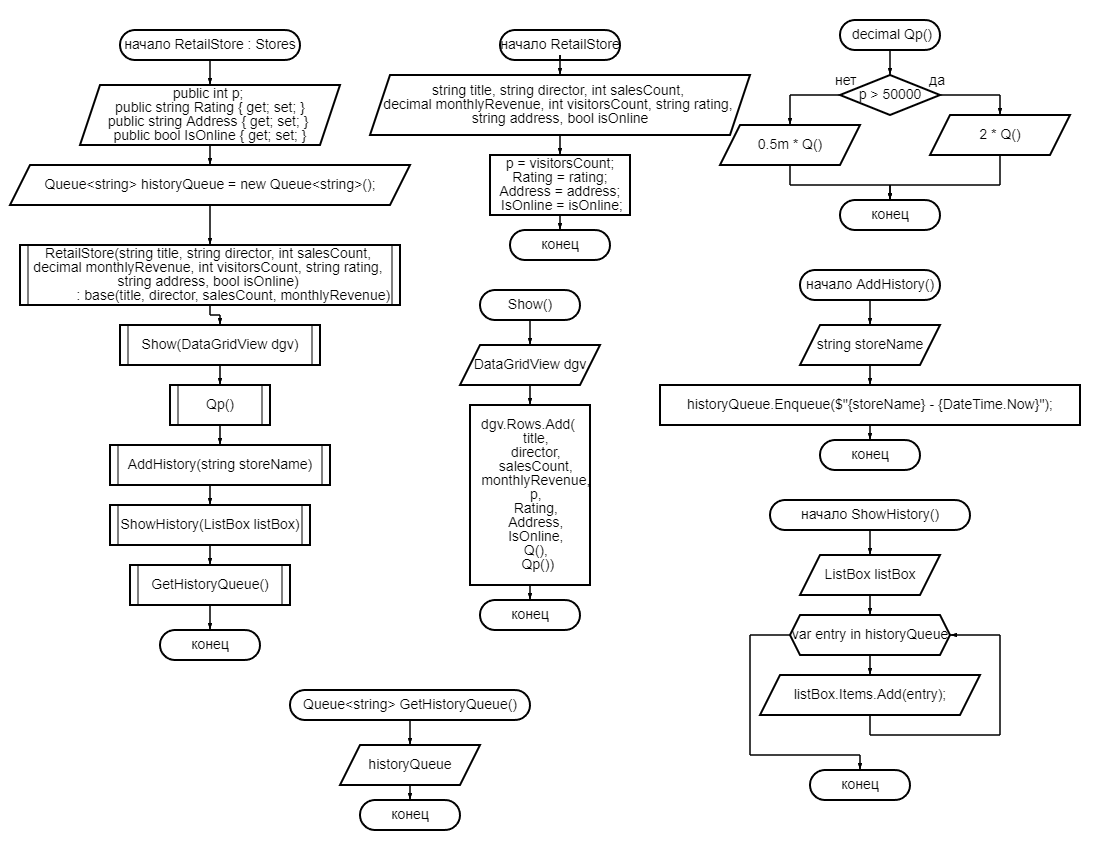
**Выходные данные:** объекты класса, занесённые в коллекцию, List<Stores>; элементы очереди, в которой хранятся все добавления магазинов, Queue<string>.

**Блок-схемы:**

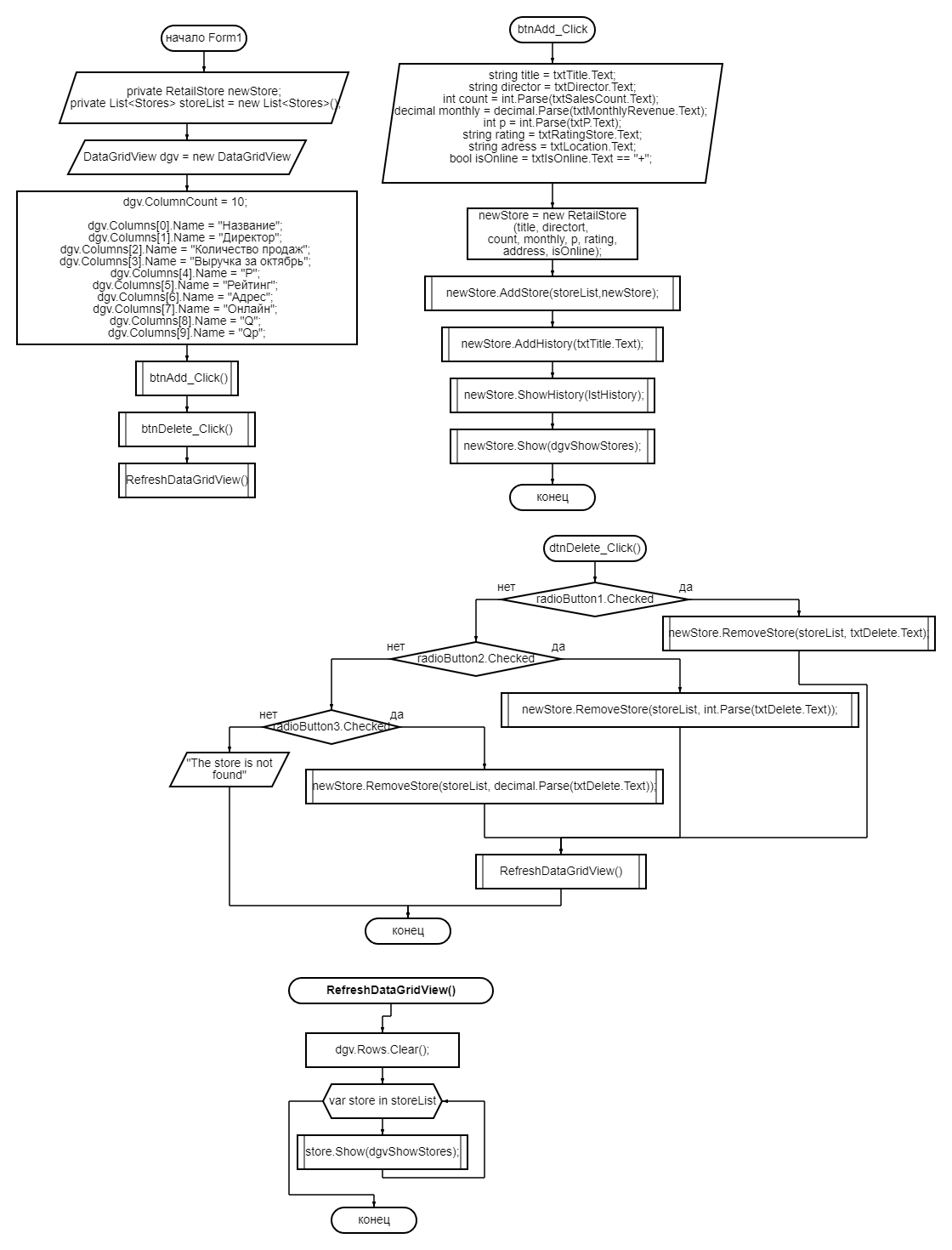
*Базовый класс:*

**

*Класс-потомок:*

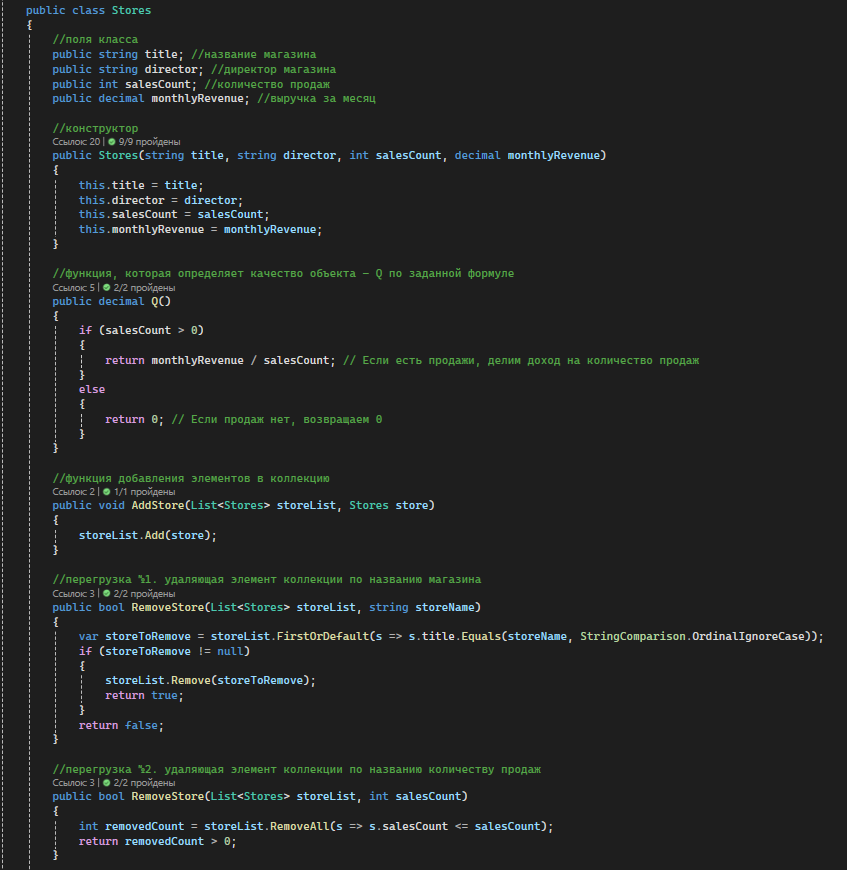
**

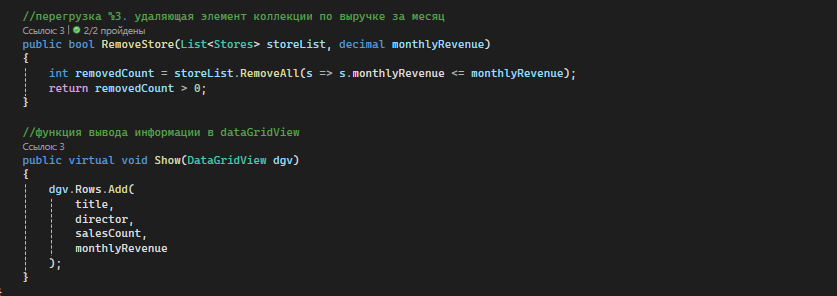
*Основная программа:*

**

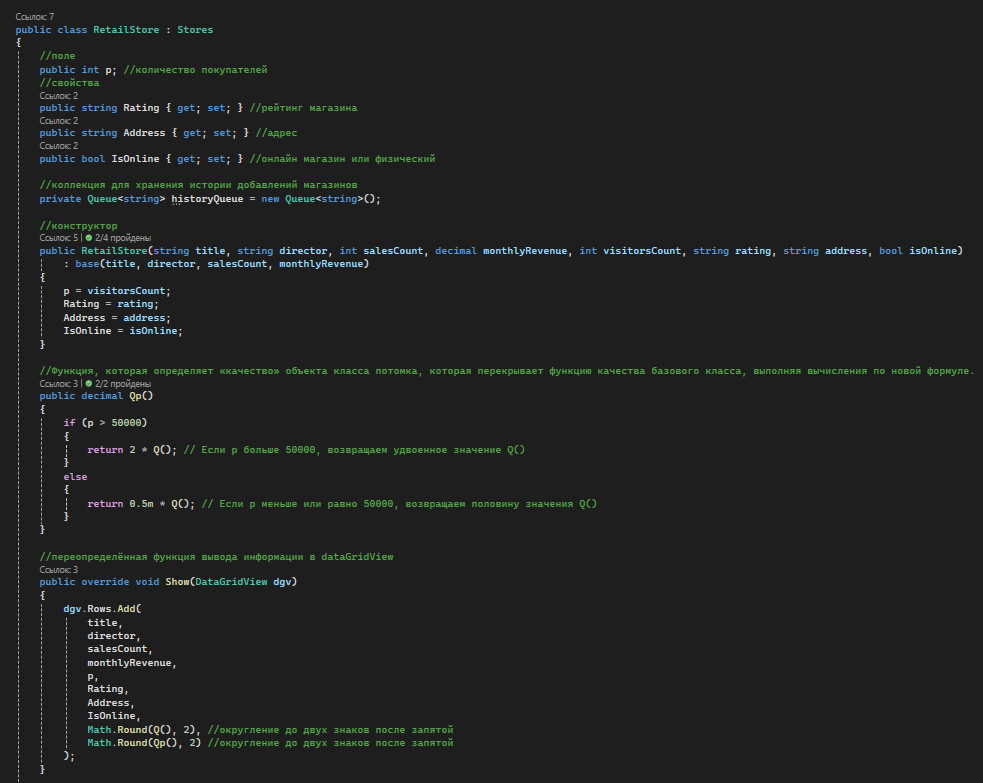
**Листинг программы:**

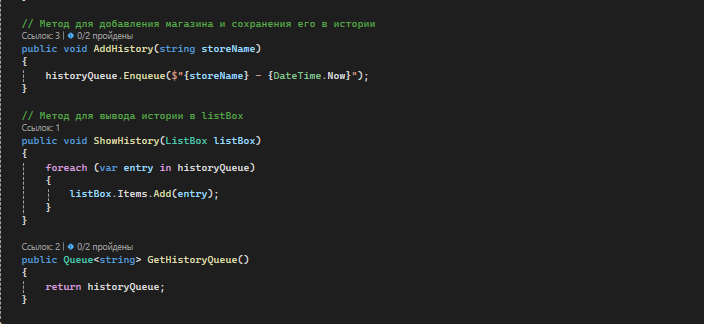
*Базовый класс:*

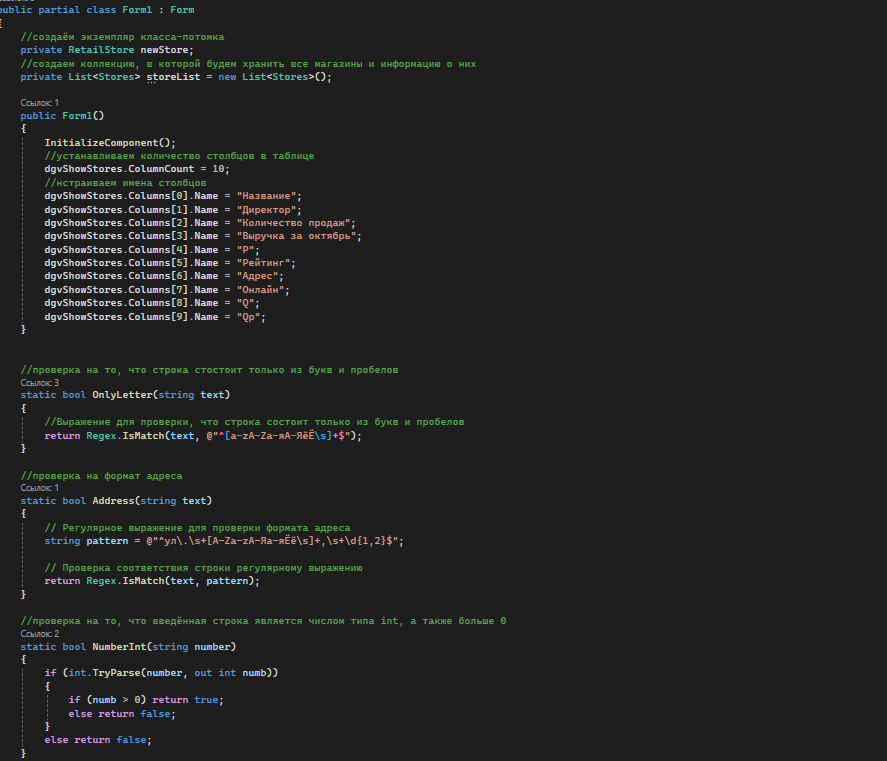
****

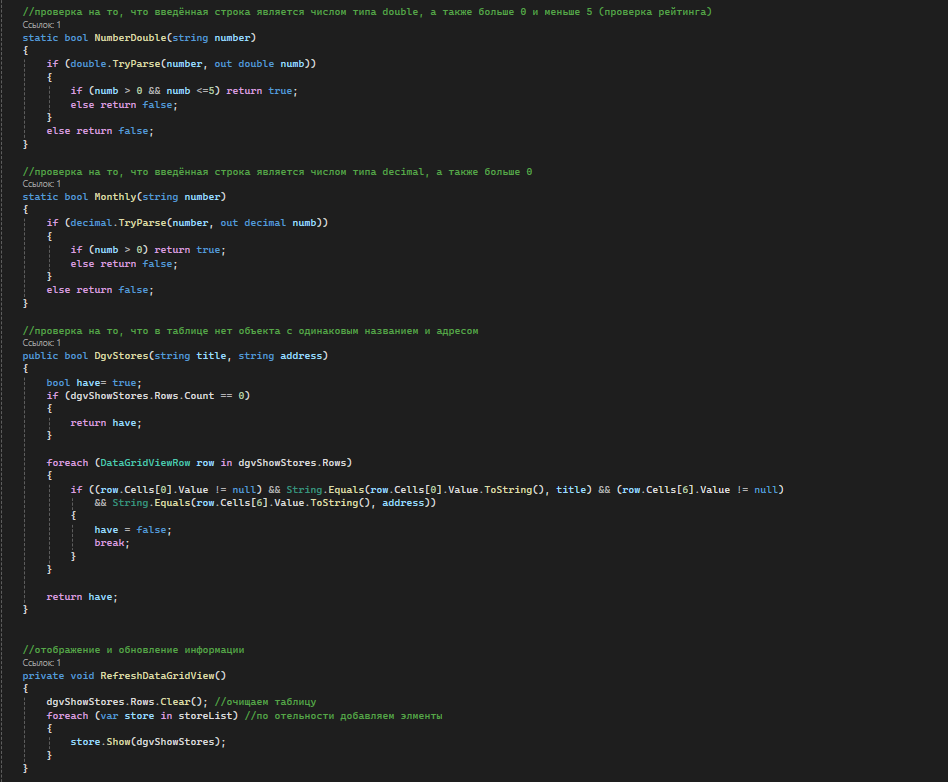
****

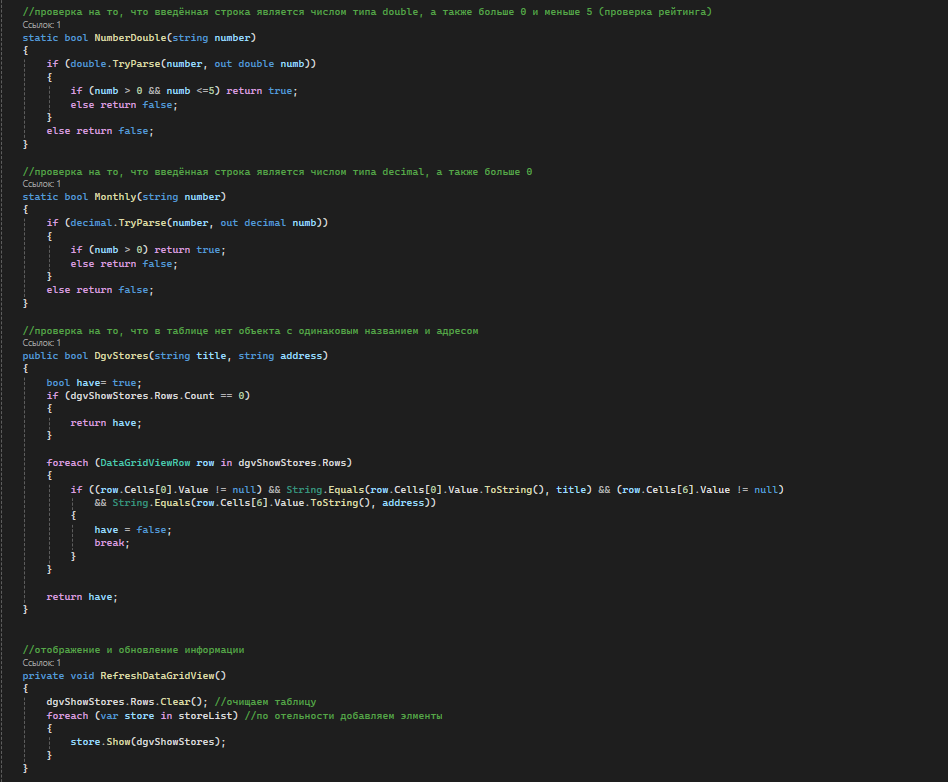
*Класс-потомок:*

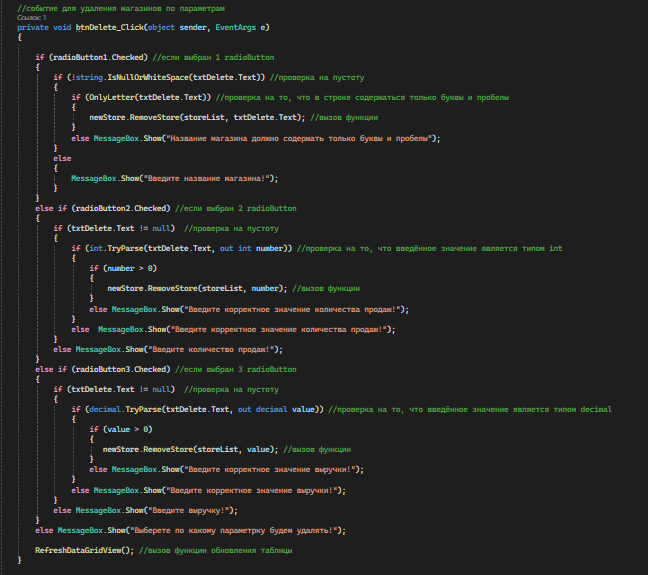
**

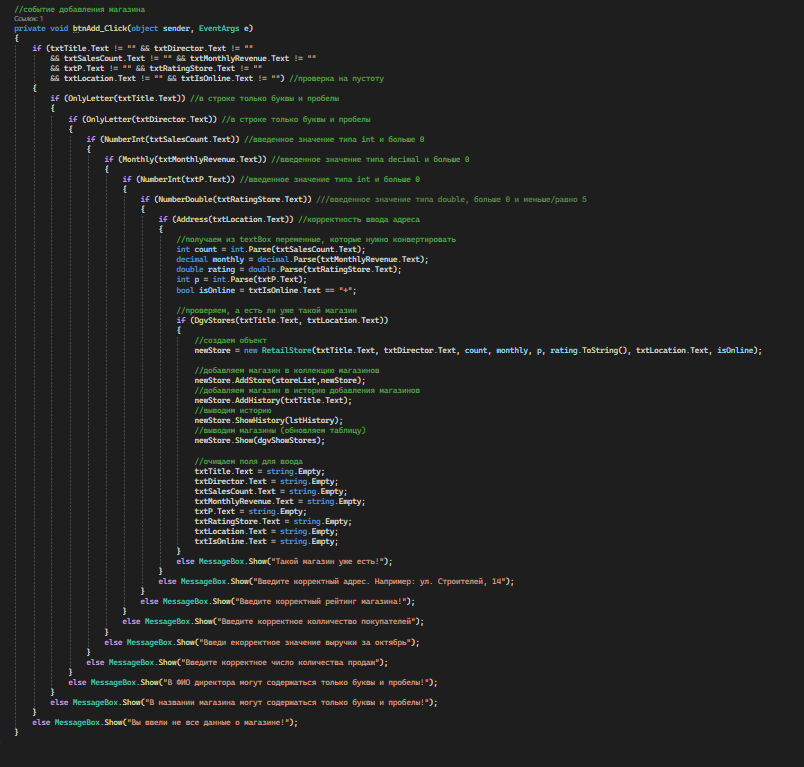
*Основной код:*

**

**

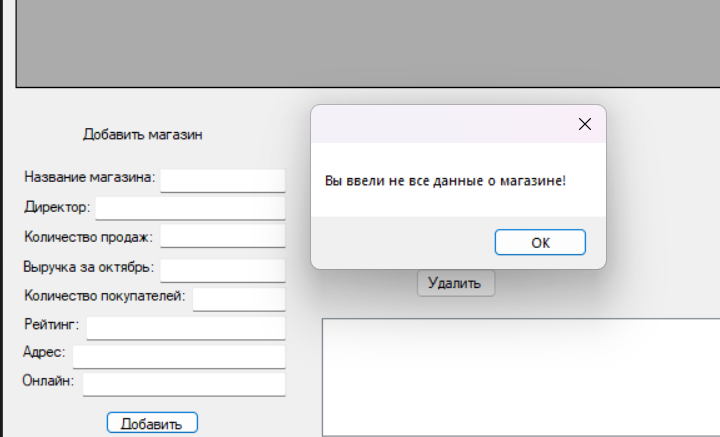
**

**

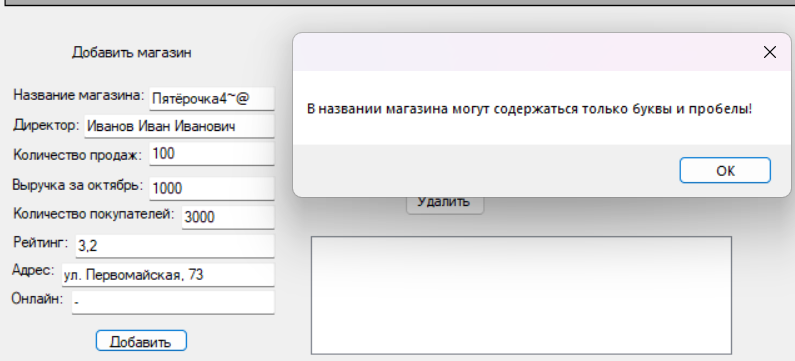
**

**Тестовые ситуации:**

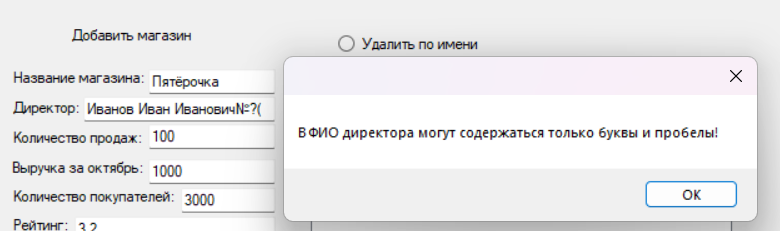
В случае, если пользователь не введёт данные в какое-либо поле для ввода и нажмет кнопку, то ему будет показано сообщение о том, что он ещё не закончил ввод данных:



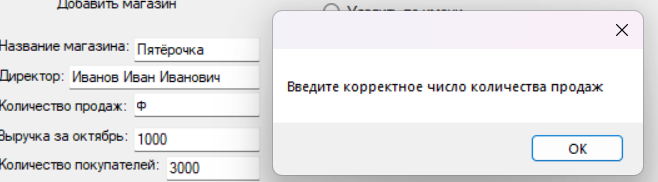
Если пользователь в поле «Название магазина» введёт какие-либо прочие символы, кроме букв и пробелов, то ему будет показано сообщение об ошибке:

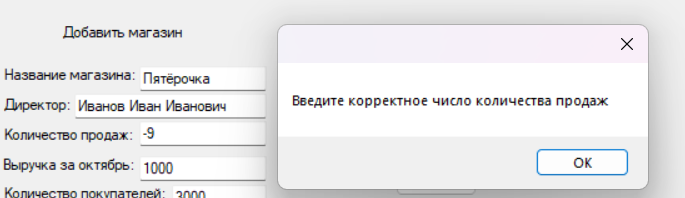


Если пользователь в поле «Директор» введёт какие-либо прочие символы, кроме букв и пробелов, то ему будет показано сообщение об ошибке:

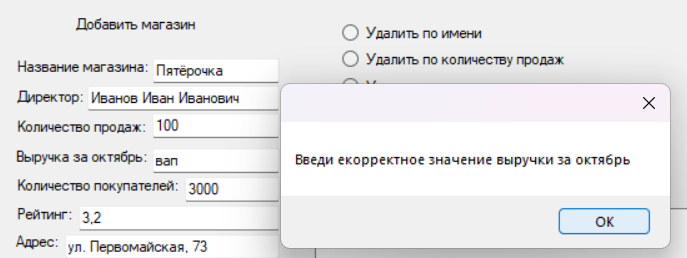


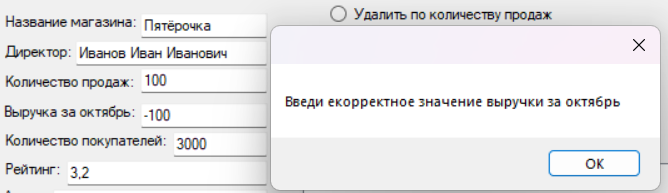
Если пользователь в поле «Количество продаж» введёт не число или отрицательное число, то ему будет показано сообщение об ошибке:



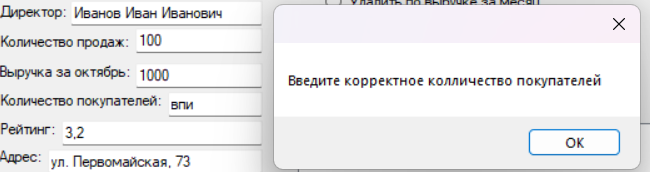


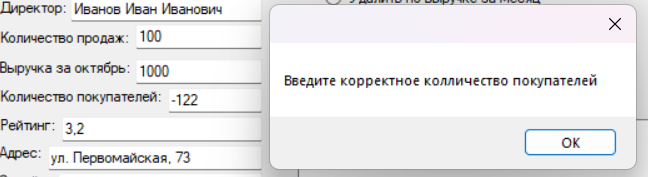
Если пользователь в поле «Выручка за октябрь» введёт не число или отрицательное число, то ему будет показано сообщение об ошибке:



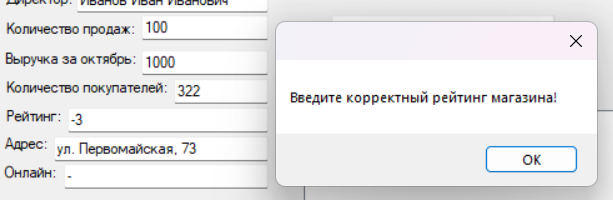


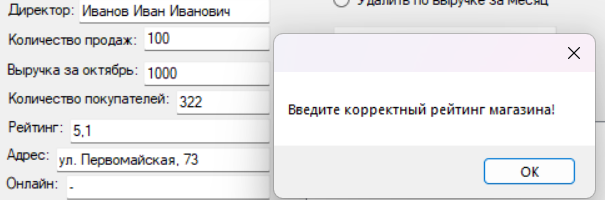
Если пользователь в поле «Количество покупателей» введёт не число или отрицательное число, то ему будет показано сообщение об ошибке:

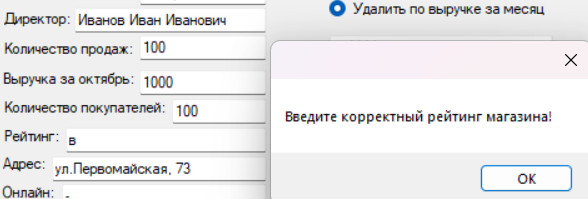




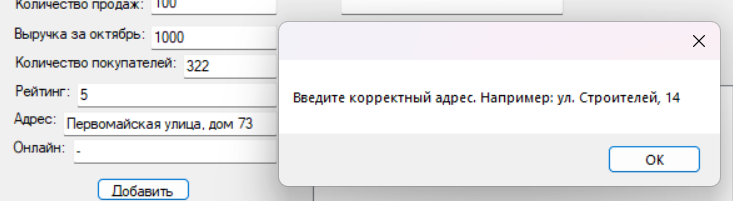
Если пользователь в поле «Количество покупателей» введёт число меньше 0 или больше 5, или не число, то ему будет показано сообщение об ошибке:

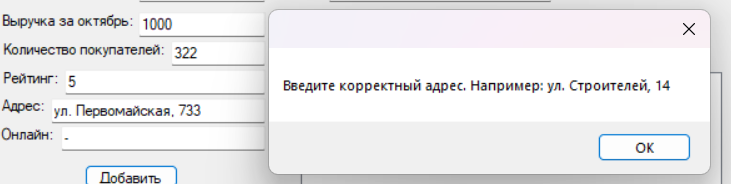






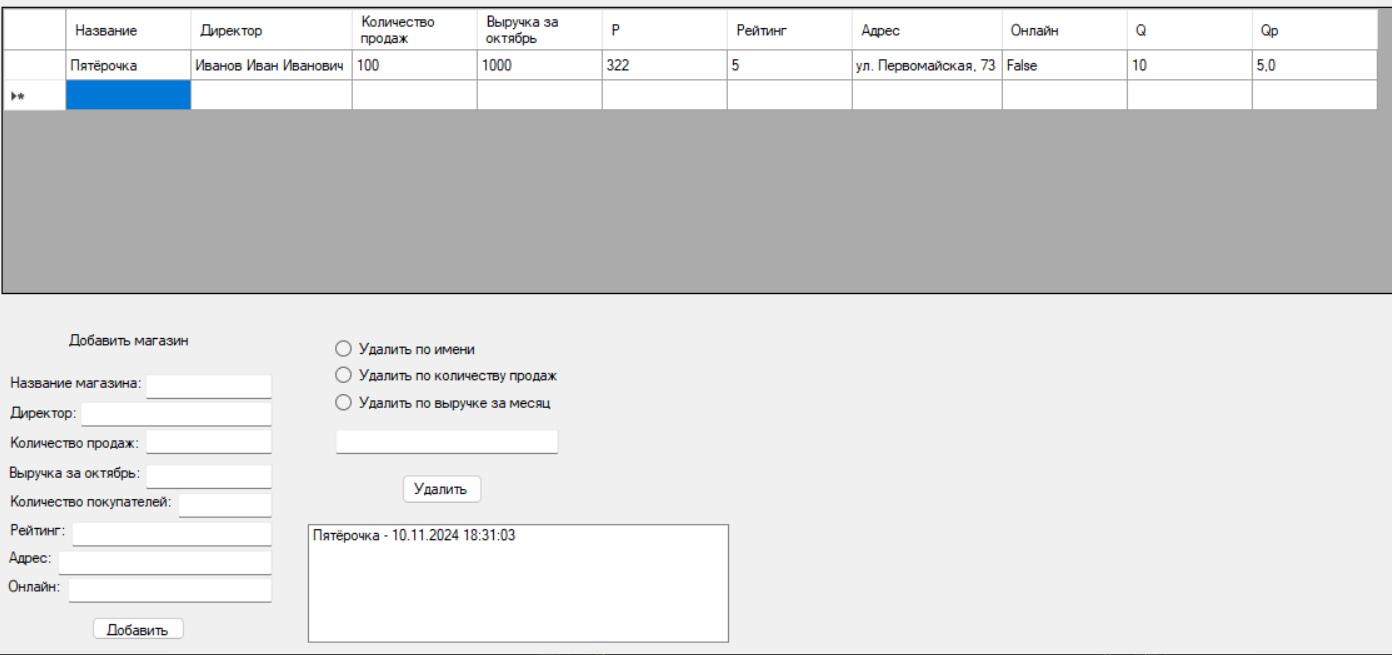
Если в поле «Адрес» пользователь введет адрес не по маске (ул.\_Название улицы,\_00) или номер дома окажется трёхзначным числом, то ему будет показано сообщение об ошибке:



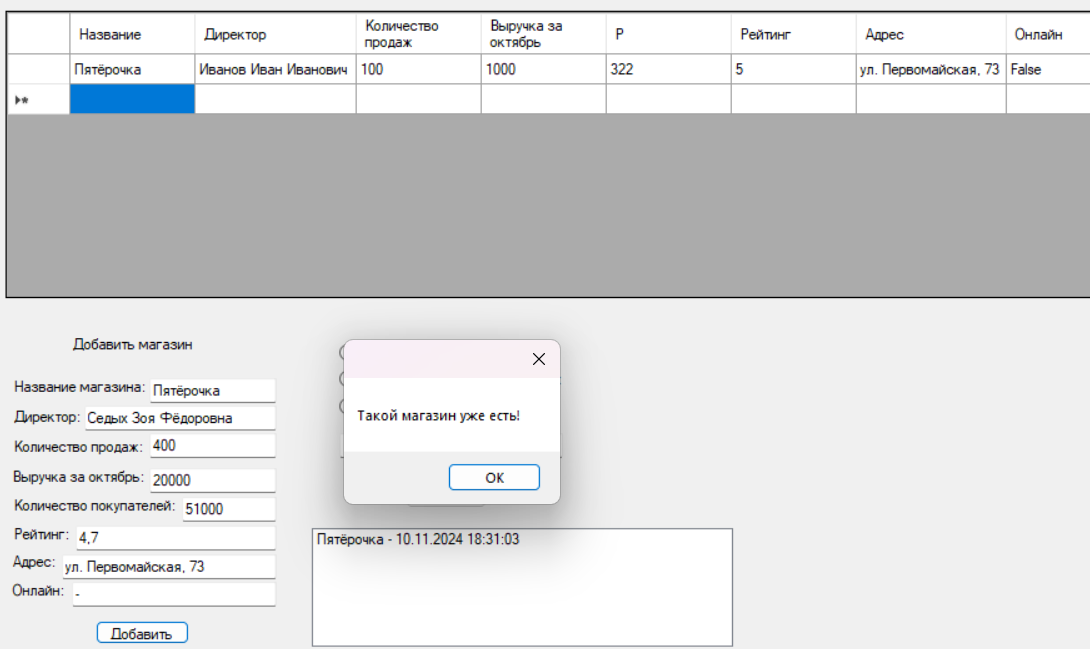


Если в поле «Онлайн» пользователь введёт «+», то магазин будет считаться онлайн. В любом другом случае, физическим.

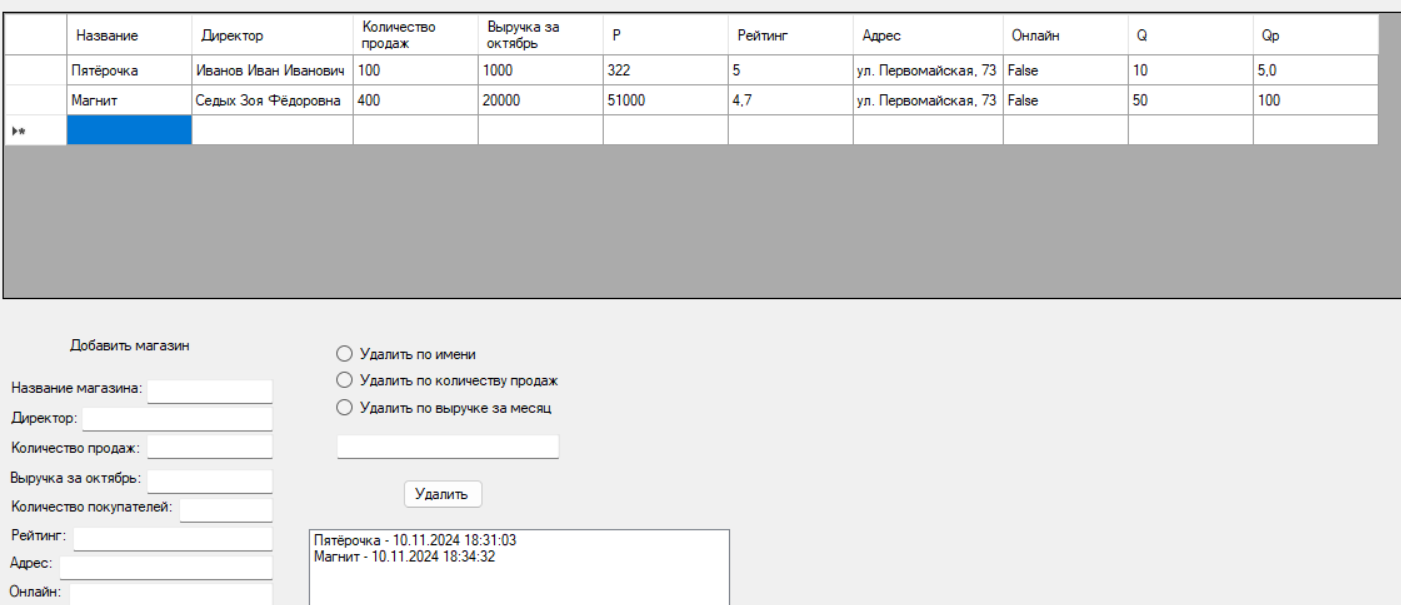
Как только пользователь вводит все данные о магазине корректно, они отображаются в таблице и листе с историей добавлений:



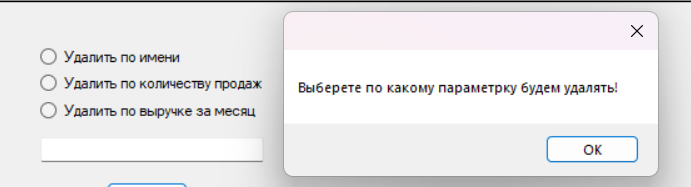
Если пользователь попытается ввести магазин, который уже существует на определённом адресе, то ему будет показано сообщение об ошибке:



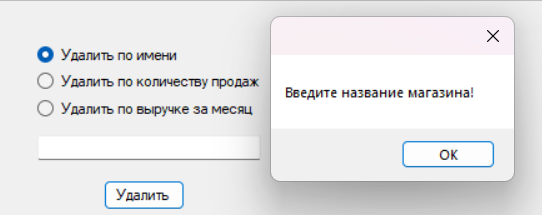
Но если пользователь поменяет название или адрес, то магазин спокойно появится в списке:



В программе также присутствует функция удаления магазинов по параметрам: по названию магазина, по количеству продаж и по выручке за месяц. Если пользователь не выберет по какому параметру будет совершаться удаление, то ему будет показано сообщение об ошибке:

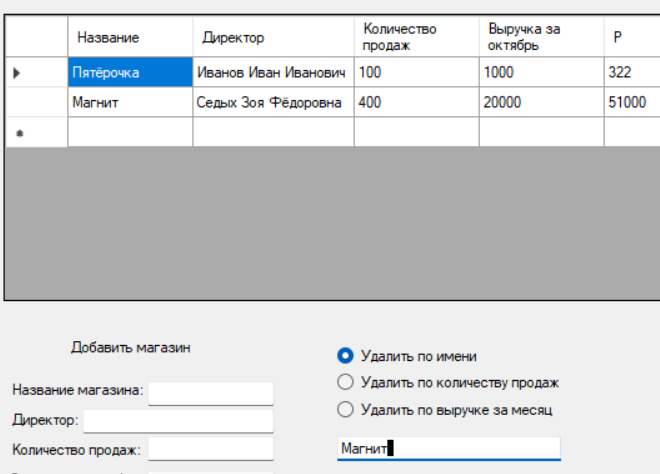
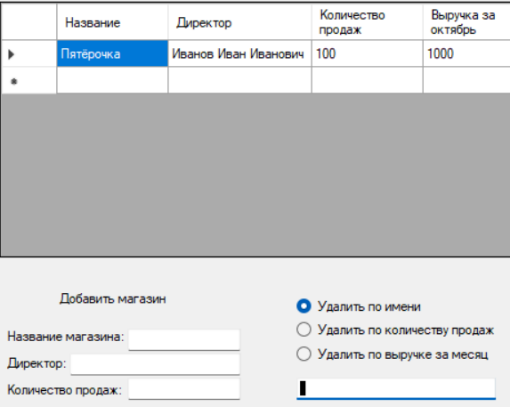


В случае, если он выберет, но оставит поле для ввода пустым, ему также будет показано сообщение об ошибке:

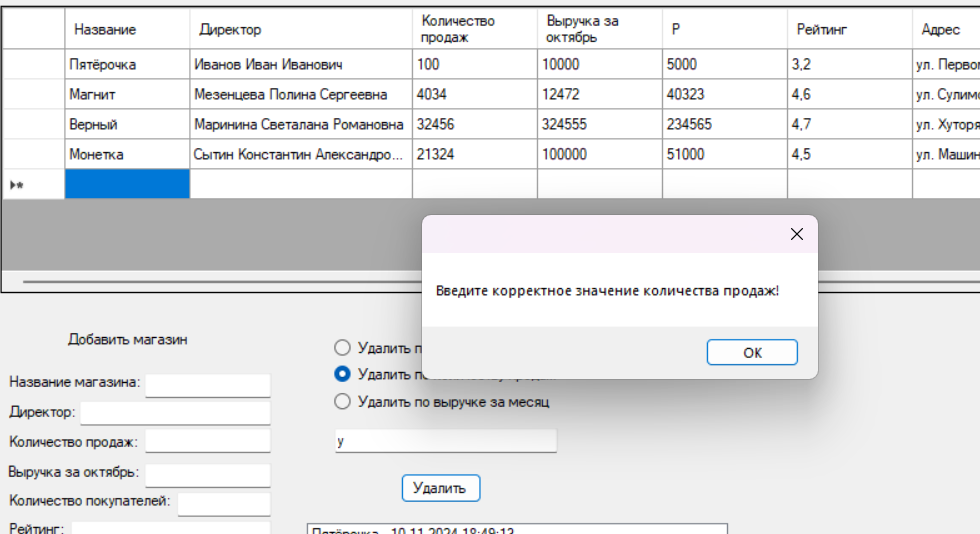


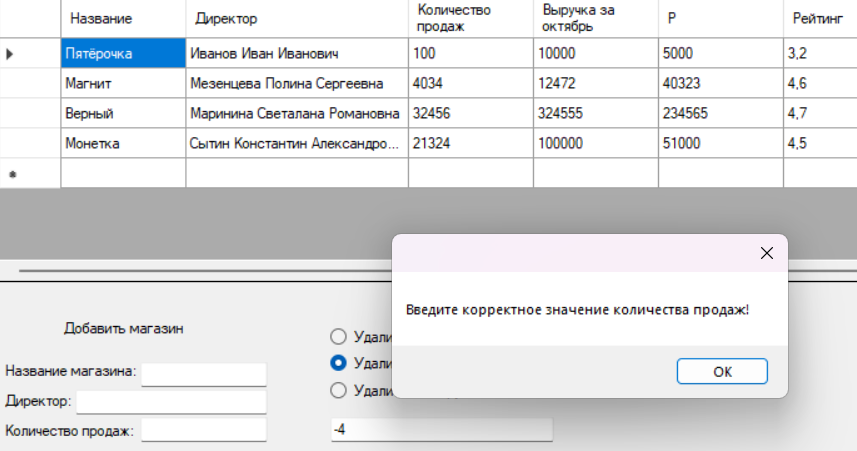
Если пользователь, выбрав параметр «Удалить по имени» укажет название магазина, которого нет в списке, то ничего не произойдёт: ни один магазин не удалится.

Если же он введёт магазин, который есть в списке, то удаление произойдет без проблем:

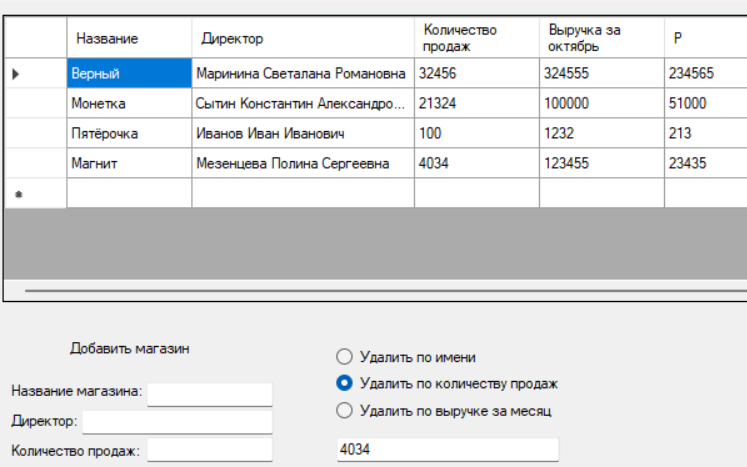
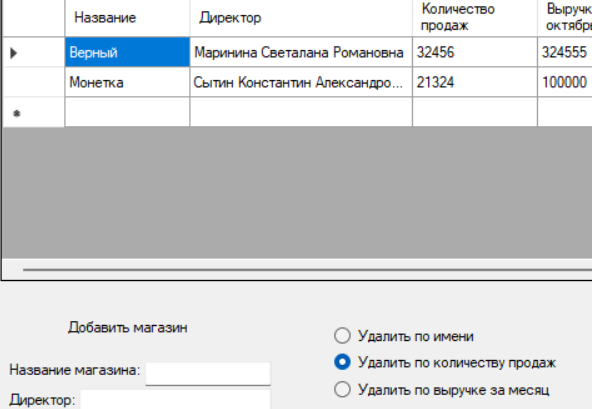


Если пользователь решит удалить магазины по количеству продаж, но введёт не число или отрицательное число, то ему будет показано сообщение об ошибке:

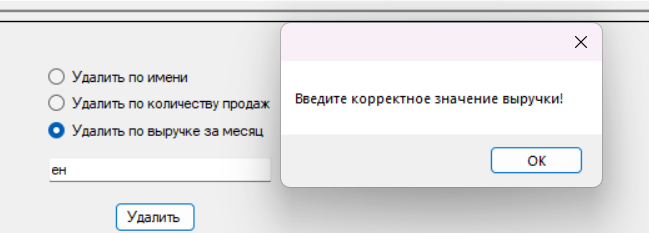


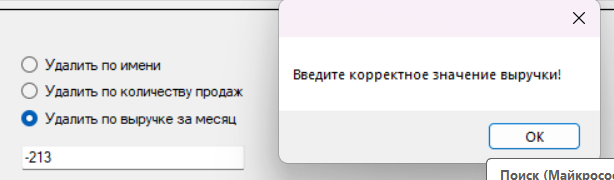


Если пользователь введёт корректное число то, те значения, которые будут меньше или равны введённому будут удалены:

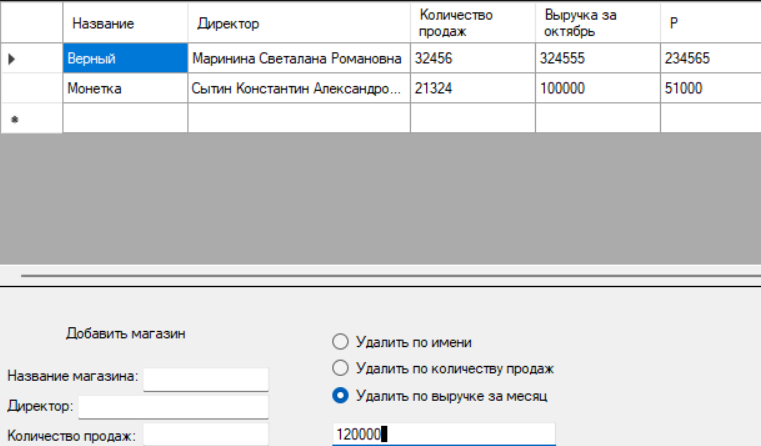
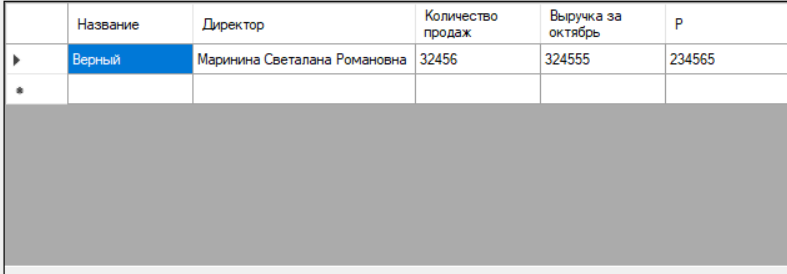


Если пользователь решит удалить магазины по выручке за месяц, но введёт не число или отрицательное число, то ему будет показано сообщение об ошибке:



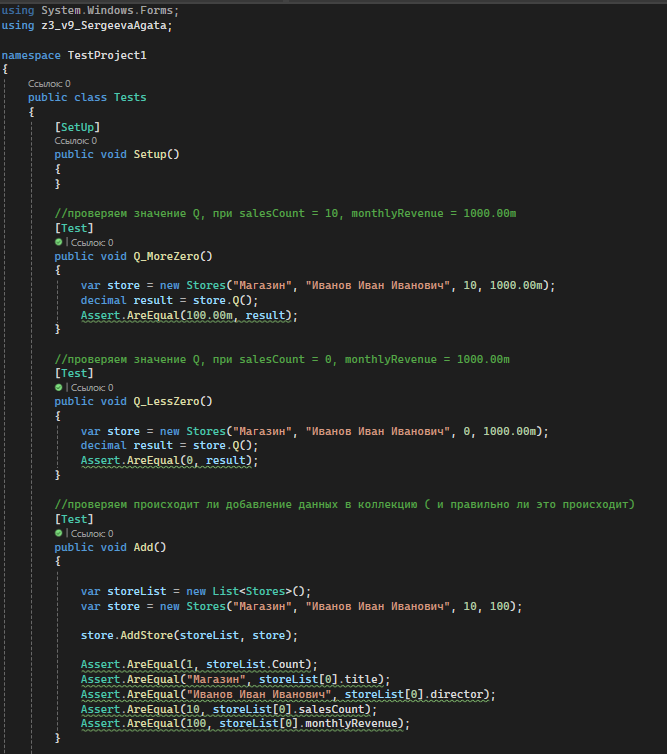


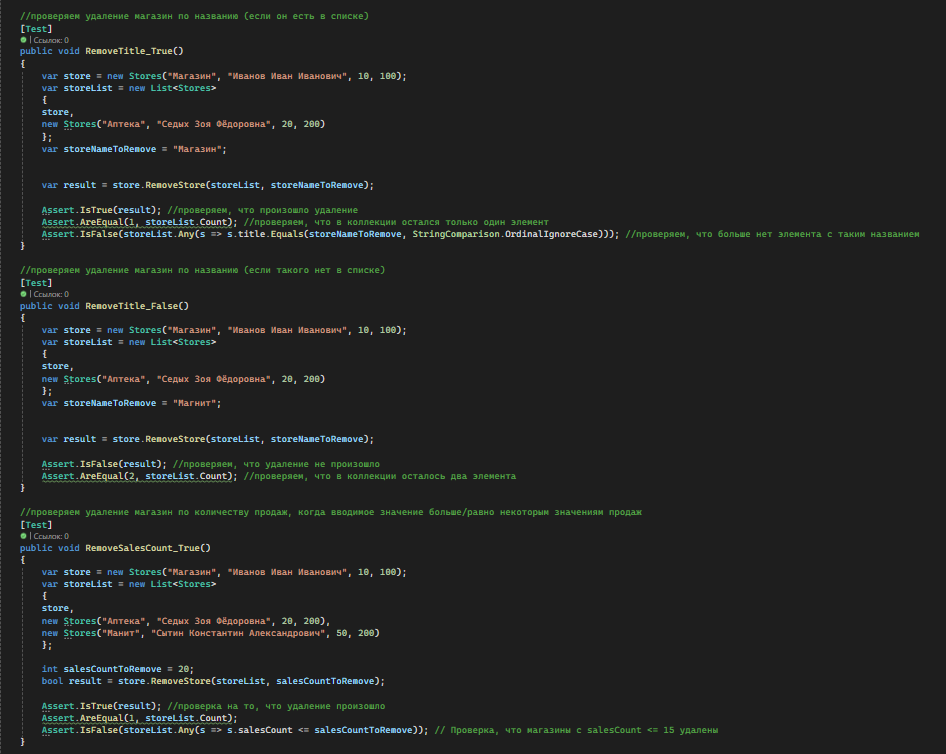
Если пользователь введёт корректное число то, те значения, которые будут меньше или равны введённому будут удалены:

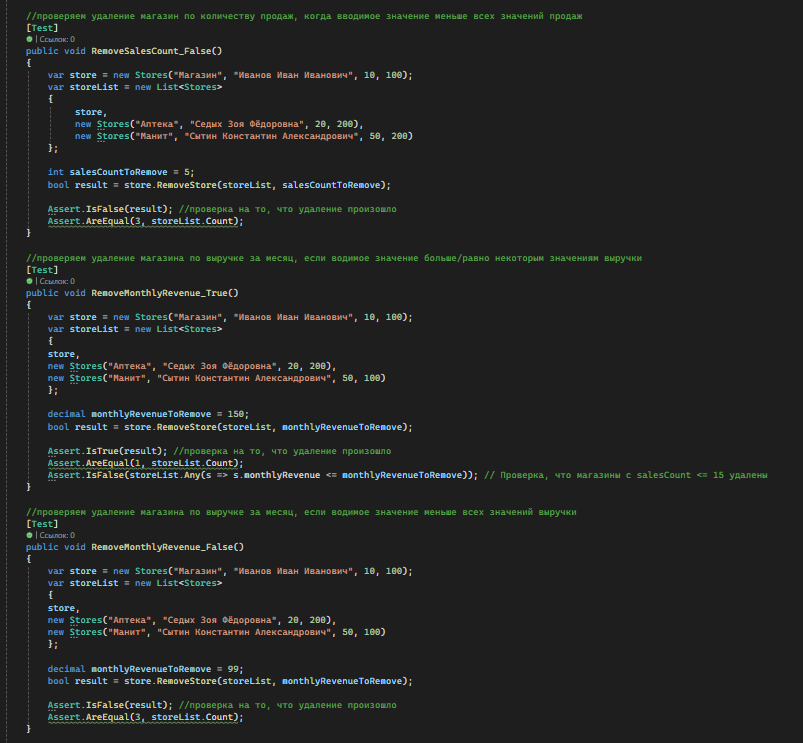


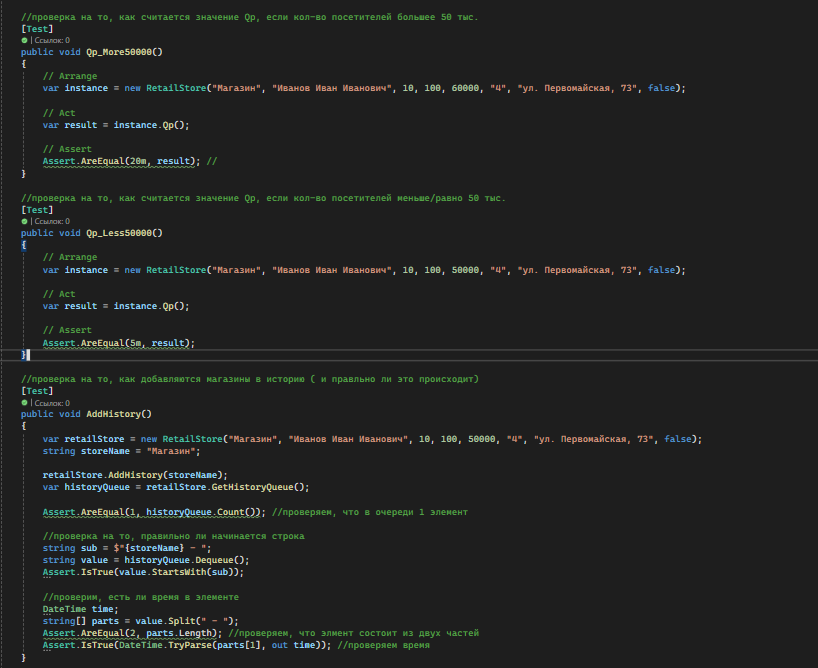
**Модульные тесты:**

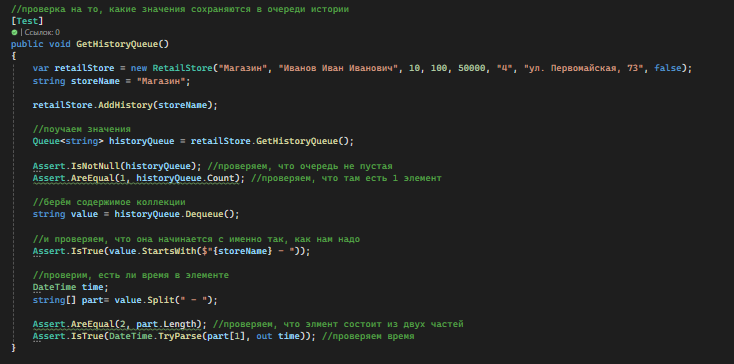
*Листинг тестов:*

**

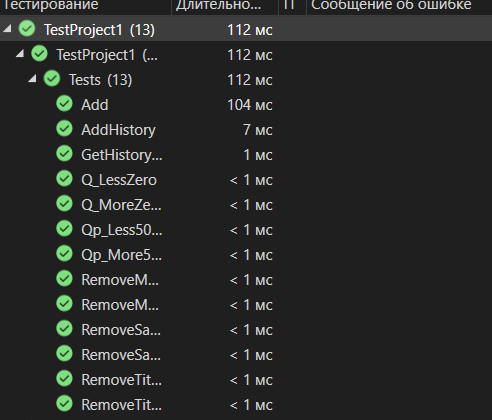
**

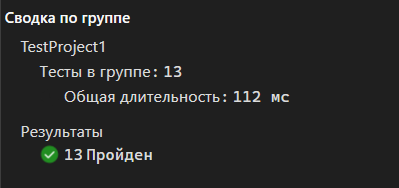
**

**

**

*Результаты прохождения тестов:*

**

****

**Используемые библиотеки:**

|  |  |
| --- | --- |
| System | Содержит основные классы и базовые классы, определяющие часто используемые значения и ссылочные типы данных, события и обработчики событий, интерфейсы, атрибуты и исключения обработки. |
| System.Collections.Generic | Содержит интерфейсы и классы, определяющие универсальные коллекции, которые позволяют пользователям создавать строго типизированные коллекции, обеспечивающие лучшую безопасность типов и производительность, чем не универсальные строго типизированные коллекции. |
| System.Text.RegularExpressions | Предоставляет классы для работы с регулярными выражениями, которые являются мощными инструментами для сопоставления шаблонов в строках. |
| System.Windows.Forms | Используется для создания многофункциональных настольных приложений с графическими пользовательскими интерфейсами (GUI) в Windows. Оно включает в себя широкий спектр элементов управления (например, кнопки, текстовые поля и таблицы данных) и функций для обработки пользовательского ввода. |

**Описание разработанных функций и методов:**

Методы базового класса:

|  |  |
| --- | --- |
| AddStore | Предназначен для добавления нового магазина в коллекцию. Метод получает коллекцию и объект класса, который нужно занести в неё, и затем добавляет его в List |
| Show | Предназначен для отображения магазинов, занесённых в коллекцию. Метод получает DataFridView и заносит в его строки поля класса. |

Функции базового класса:

|  |  |
| --- | --- |
| Q | Функция, которая определяет качество объекта Q по заданной формуле. Возвращает значение decimal. |
| RemoveStore | Функция, состоящая из 3-х перегрузок, предназначенных для удаления объектов по критериям : по названию, по количеству продаж, по выручке. Функция принимает коллекцию и параметр для удаления. Функция возвращает значения true/false. |

Методы класса-потомка:

|  |  |
| --- | --- |
| Show | Предназначен для отображения магазинов, занесённых в коллекцию. Метод получает DataFridView и заносит в его строки поля класса. |
| AddHistory | Метод принимает строку с названием магазина и добавляет её в коллекцию Queue. |
| ShowHistory | Метод принимает ListBox и с помощью цикла foreach берёт элементы из коллекции Queue и выводит их. |

Функции базового класса:

|  |  |
| --- | --- |
| decimal Qp | Функция, которая определяет «качество» объекта класса потомка Qp, которая перекрывает функцию качества базового класса,  выполняя вычисления по новой формуле. |
| Queue<string> GetHistoryQueue | Функция доступа, которая возвращает текущее состояние очереди, её текущий элемент. |

**Используемые инструменты:** WindowsForms C#.