|  |
| --- |
| **Министерство образования и науки Российской Федерации**  Федеральное государственное автономное образовательное  учреждение высшего образования  **«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  **ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»** |

Подразделение: Инженерная школа энергетики

Направление подготовки: 09.04.03 – Прикладная информатика

Отделение: Электроэнергетики и электротехники

**Проектная документация**

**Отчёт по лабораторной работе №5**

по дисциплине: «Основы объектно-ориентированного программирования»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент гр. О-5КМ91 | | | \_\_\_\_\_\_\_\_\_\_ | |  | Васильев В. В. |
|  | |  |  | |  |  |
|  | |  |  | \_\_\_ \_\_\_\_\_\_ 2021 г. | | |
|  | |  |  | |  |  |
| Отчёт принял доцент, к.т.н. | | | \_\_\_\_\_\_\_\_\_\_ | |  | Калентьев А. А. |
|  |  |  |  | |  |  |
|  |  |  |  | \_\_\_ \_\_\_\_\_\_ 2021 г. | | |

Томск – 2021

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc89853961)

[1 Основная часть 4](#_Toc89853962)

[1.1 UML диаграмма вариантов использования 4](#_Toc89853963)

[1.2 UML диаграмма классов 5](#_Toc89853964)

[1.3 Описание классов, образующих связь типа «общее-частное» 6](#_Toc89853965)

[1.4 Дерево ветвлений Git 10](#_Toc89853966)

[1.5 Тестирование программы 11](#_Toc89853967)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17](#_Toc89853968)

[ПРИЛОЖЕНИЕ А 18](#_Toc89853969)

# ВВЕДЕНИЕ

Корректная и полная документация сопровождает разработку программного обеспечения (далее – ПО) от появления идеи до выпуска конечного продукта. Написание документации является обязательным критерием разработки и последующей поддержки проекта [1].

Целью данной лабораторной работы является разработка проектной документации на созданный программный продукт.

Для достижения поставленной цели должны быть выполнены следующие задачи:

* Составление технического задания (далее – ТЗ) на разработанную программу (Приложение А);
* Составление UML диаграммы вариантов использования для разработанной программы;
* Составление UML диаграммы классов;
* Описание классов, образующих связь типа «общее-частное»;
* Привести дерево ветвлений Git;
* Провести тестирование программы.

# Основная часть

## 1.1 UML диаграмма вариантов использования

Вариант использования (use case) — это описание множества последовательных действий (включая вариации), которые выполняются некоторым субъектом с целью получения результата, значимого для некоторого действующего лица [1]. ВИ предполагает взаимодействие действующих лиц и системы или другого объекта. Действующее лицо представляет собой логически связанное множество ролей, которые играют пользователи системы во время взаимодействия с ней.

Диаграмма вариантов использования для разработанного ПО приведена на рисунке 1.

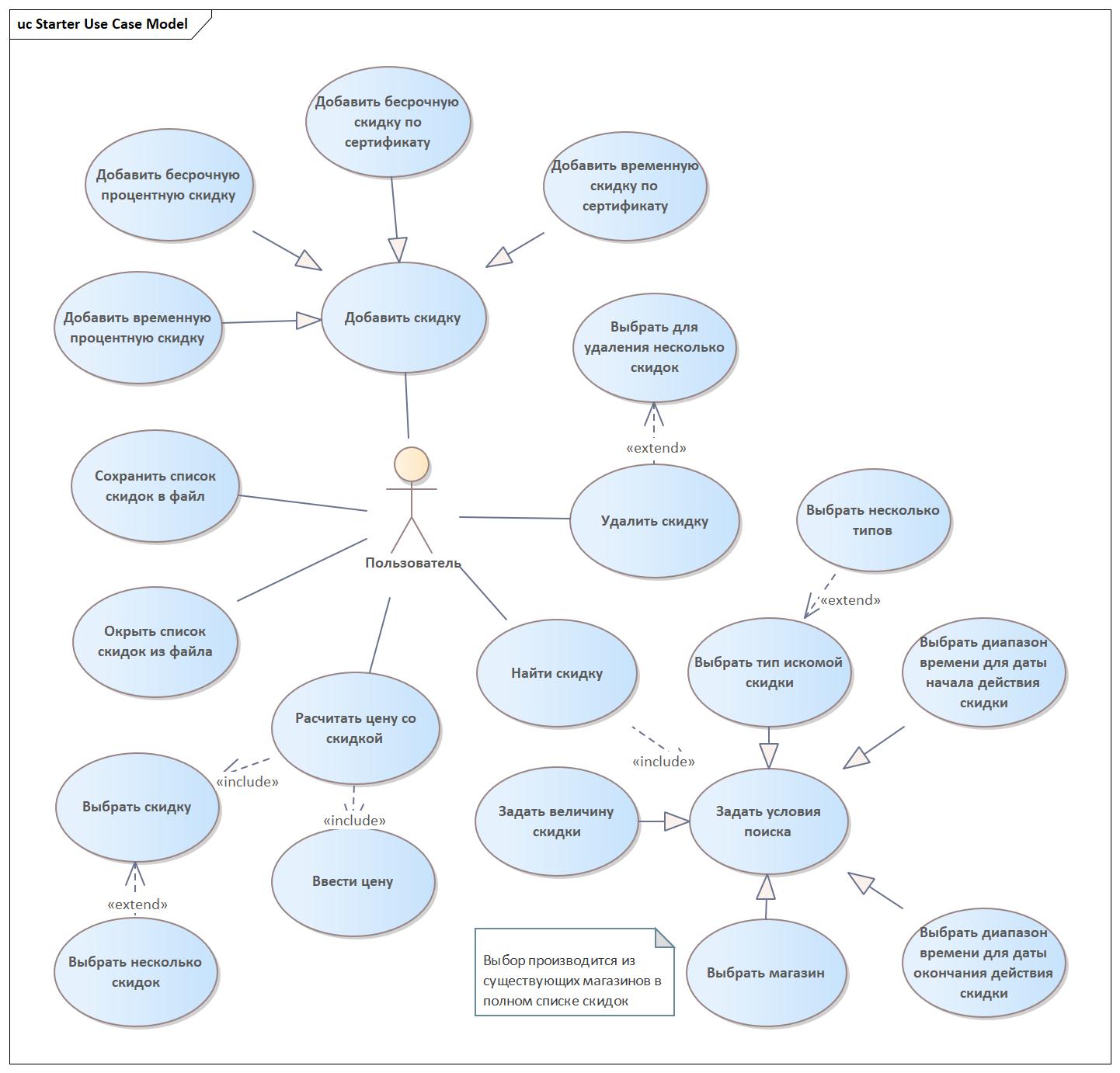


Рисунок 1 – Диаграмма вариантов использования

## 1.2 UML диаграмма классов

Диаграмма классов — это центральная методика моделирования, которая используется практически во всех объектно-ориентированных методах. Эта диаграмма описывает типы объектов в системе и различные виды статических отношений, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области (в процессе анализа) или элементы программной системы (в процессах проектирования и реализации).

Диаграмма классов приведена на рисунке 2.

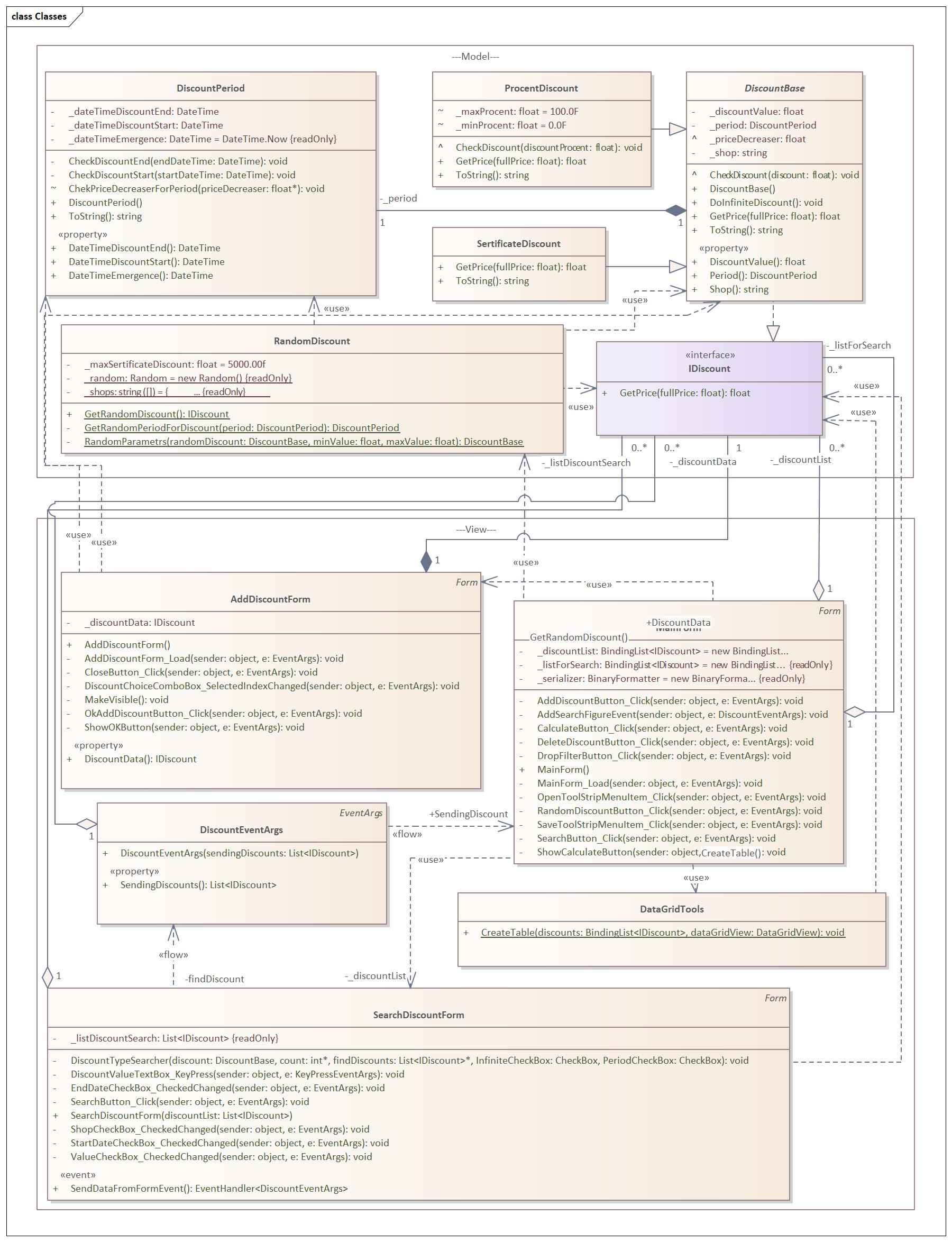


Рисунок 2 – UML диаграмма классов

## 1.3 Описание классов, образующих связь типа «общее-частное»

В таблице 1 приведено описание интерфейса IDiscount с его свойствами и методами.

Таблица 1 – Описание интерфейса IDiscount

|  |  |  |
| --- | --- | --- |
| Описание класса | | |
| Класс IDiscount– интерфейс для скидок | | |
| Методы | | |
| Название | Тип | Описание |
| GetPrice(fullPrice : float) | float | Метод для расчета цены товара со скидкой.  fullPrice – исходная цена товара. |

В таблице 2 приведено описание абстрактного класса DiscountBase с его полями, свойствами и методами, который реализуется интерфейсом IDiscount.

Таблица 2 – Описание класса DiscountBase

|  |  |  |  |
| --- | --- | --- | --- |
| Описание класса | | | |
| Класс *DiscountBase* – базовый абстрактный класс для скидок. | | | |
| Поля | | | |
| Название | Тип | Доступ | Описание |
| \_discountValue | float | Private | Величина скидки. |
| \_shop | string | Private | Информация о скидке. |
| \_priceDecreaser | float | Protected Private | Уменьшитель цены, используемый для расчета цены со скидкой. |
| \_period | Discount Period | Private | Период действия скидки. |
| Свойства | | | |
| DiscountValue | float | Public | Величина скидки. |
| Shop | string | Public | Информация о скидке. |
| Period | Discount Period | Public | Период действия скидки. |
| Методы | | | |
| CheckDiscount (discount : float) | void | Private Protected | Проверка корректности ввода данных.  discount – присваиваемая величина скидки.  Абстрактное свойство, переопределяется в производных классах. |
| DoInfiniteDiscount() | void | Public | Делает бесконечный период действия скидки |
| ToString() | string | Public | Вывод информации о скидке.  Абстрактное свойство, переопределяется в производных классах. |
| GetPrice (fullPrice : float) | float | Public | Вычисляет цену товара со скидкой.  fullPrice – исходная цена товара. |

В таблицах 3-4 приведено описания классов SertificateDiscount и ProcentDiscount, которые наследуются от DiscountBase.

Таблица 3 – Описание класса SertificateDiscount

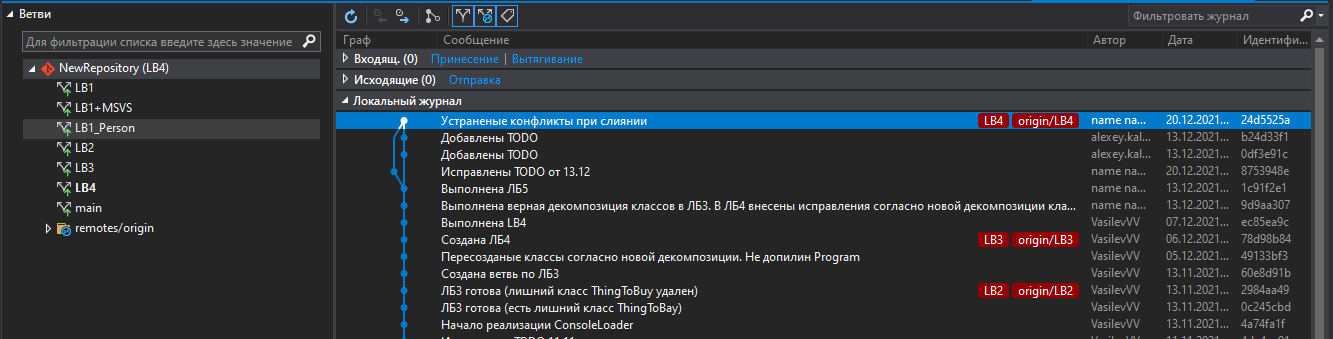
|  |  |  |  |
| --- | --- | --- | --- |
| Описание класса | | | |
| Класс SertificateDiscount– класс для скидок по сертификату. | | | |
| Поля | | | |
| Название | Тип | Доступ | Описание |
| \_discountValue | float | Private | Величина скидки. |
| \_shop | string | Private | Информация о скидке. |
| \_priceDecreaser | float | Protected Private | Уменьшитель цены, используемый для расчета цены со скидкой. |
| \_period | Discount Period | Private | Период действия скидки. |
| Свойства | | | |
| DiscountValue | float | Public | Величина скидки. |
| Shop | string | Public | Информация о скидке. |
| Period | Discount Period | Public | Период действия скидки. |
| Методы | | | |
| CheckDiscount (discount : float) | void | Private Protected | Проверка корректности ввода данных.  discount – присваиваемая величина скидки |
| DoInfiniteDiscount() | void | Public | Делает бесконечный период действия скидки |
| GetPrice (fullPrice : float) | float | Public | Вычисляет цену товара со скидкой.  fullPrice – исходная цена товара. |
| ToString() | string | Public | Вывод информации о скидке |

Таблица 4 – Описание класса ProcentDiscount

|  |  |  |  |
| --- | --- | --- | --- |
| Описание класса | | | |
| Класс ProcentDiscount – класс для процентных скидок. | | | |
| Поля | | | |
| Название | Тип | Доступ | Описание |
| \_discountValue | float | Private | Величина скидки. |
| \_shop | string | Private | Информация о скидке. |
| \_priceDecreaser | float | Protected Private | Уменьшитель цены, используемый для расчета цены со скидкой. |
| \_period | Discount Period | Private | Период действия скидки. |
| \_maxProcent | float | Private | Максимальная величина процентной скидки. Используется для валидации данных. |
| \_minProcent | float | Private | Минимальная величина процентной скидки. Используется для валидации данных. |
| Свойства | | | |
| DiscountValue | float | Public | Величина скидки. |
| Shop | string | Public | Информация о скидке. |
| Period | Discount Period | Public | Период действия скидки. |
| Методы | | | |
| CheckDiscount (discount : float) | void | Private Protected | Проверка корректности ввода данных.  discount – присваиваемая величина скидки |
| DoInfiniteDiscount() | void | Public | Делает бесконечный период действия скидки |
| GetPrice (fullPrice : float) | float | Public | Вычисляет цену товара со скидкой.  fullPrice – исходная цена товара. |
| ToString() | string | Public | Вывод информации о скидке |

## 1.4 Дерево ветвлений Git

На рисунке 3 представлено дерево ветвлений Git, полученное по окончании работы с проектом.



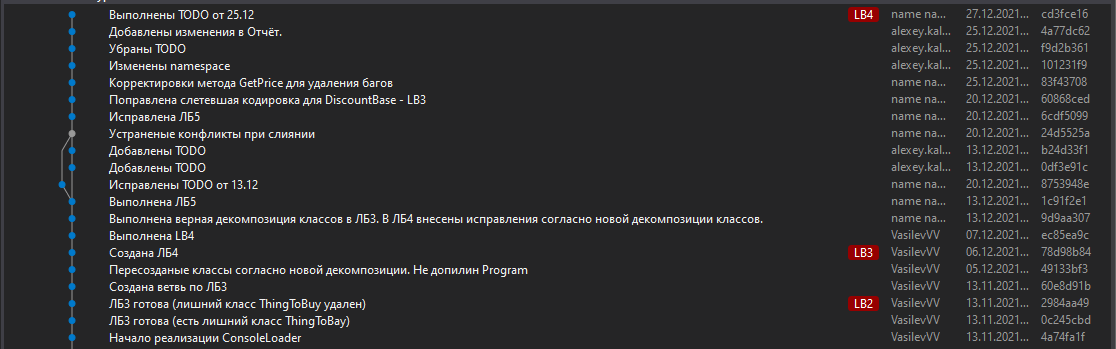


Рисунок 3 – Дерево ветвлений и коммиты в Git

## 1.5 Тестирование программы

Далее приводится процесс функционального тестирования программы. Графический интерфейс пользователя представлен на рисунке 4.

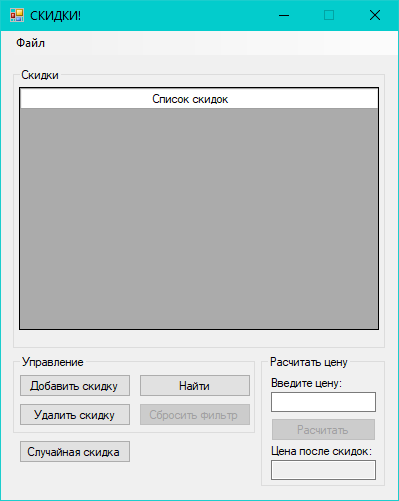


Рисунок 4 – Графический интерфейс пользователя

**1.5.1 Тестовый случай «Добавить скидку»**

Для добавления элемента необходимо вызвать соответствующую форму путём нажатия кнопки «Добавить скидку» (рисунок 5).

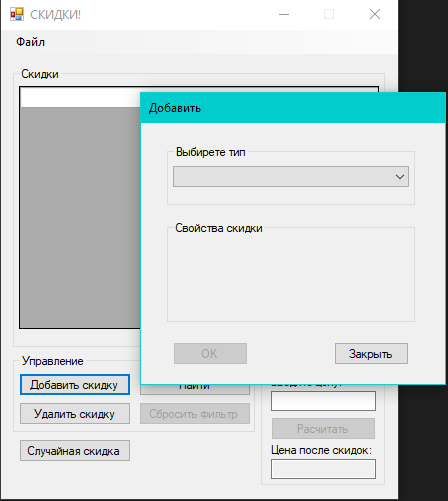


Рисунок 5 – Форма для добавления элемента

Параметры скидки можно ввести, выбрав соответствующий тип в выпадающем меню (бессрочная скидка по сертификату, временная скидка по сертификату, бессрочная процентная скидка, временная процентная скидка).

После ввода данных необходимо нажать кнопку «ОК», элемент появится в таблице главной формы (рисунки 6 и 7).

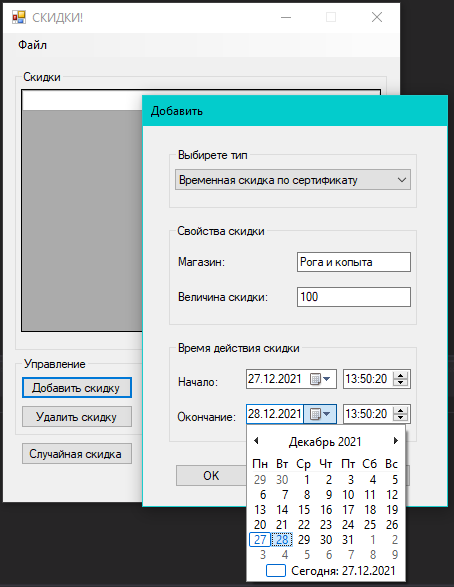


Рисунок 6 – Заполнение полей

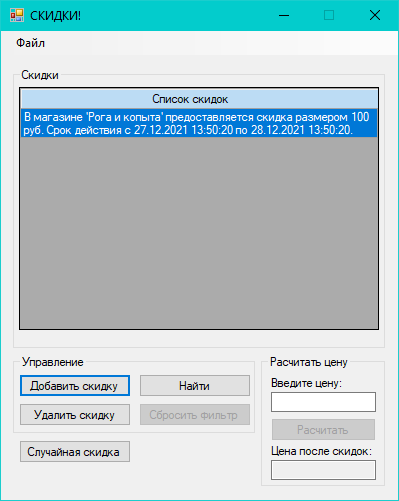


Рисунок 7 – Успешное добавление нового элемента

В программе предусмотрена система обработки некорректного ввода данных пользователем. При попытке ввести неправильную величину скидки (отрицательное число или не число) блокируется кнопка «ОК». Для того чтобы пользователь понимал, что именно требуется ввести, при наведении курсора на поле ввода выпадает ToolTip с сообщением «Величина скидки может быть числом или десятичной дробью (через ',')». Аналогичным образом обрабатываются и пустые значения в TextBox для ввода.

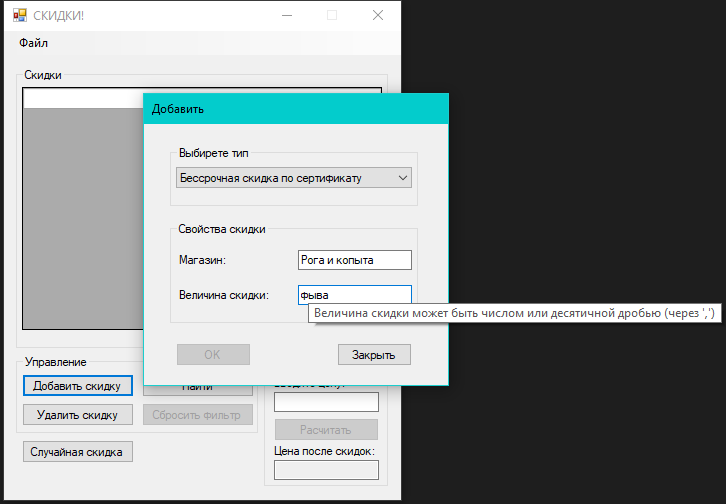


Рисунок 8 – Некорректный ввод

При неправильном вводе периода действия скидки появится сообщение об ошибке.

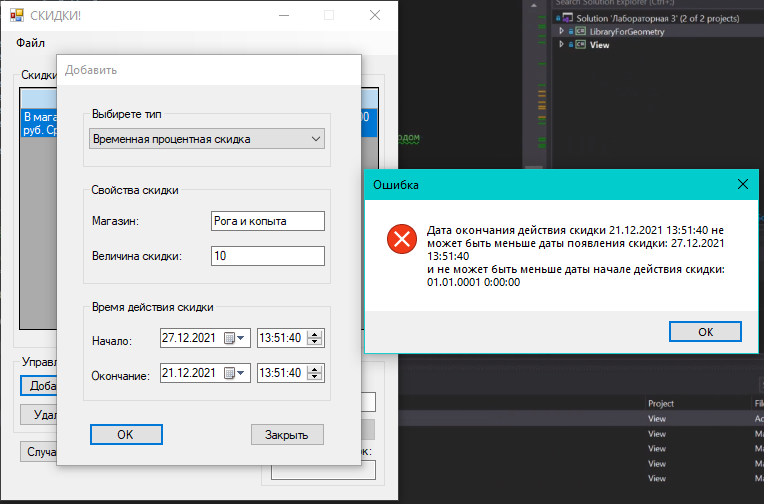


Рисунок 9 – Некорректный ввод периода времени

**1.5.2 Тестовый случай «Удалить элемент»**

Для удаления одного или нескольких элементов необходимо выбрать их в таблице и нажать на кнопку «Удалить» (рисунки 11-14).

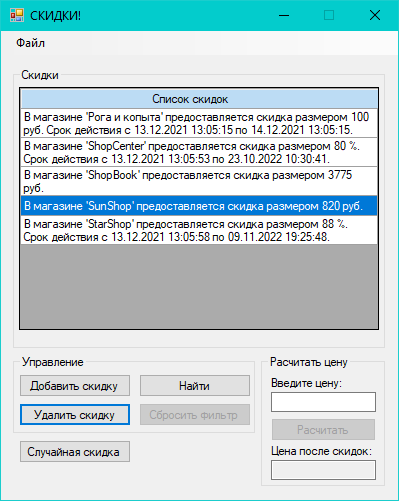
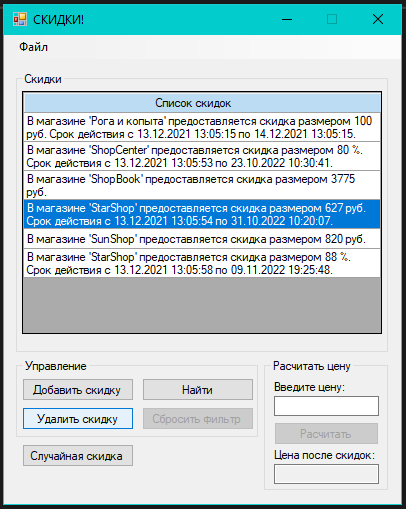


Рисунок 12 – Удаление одного элемента

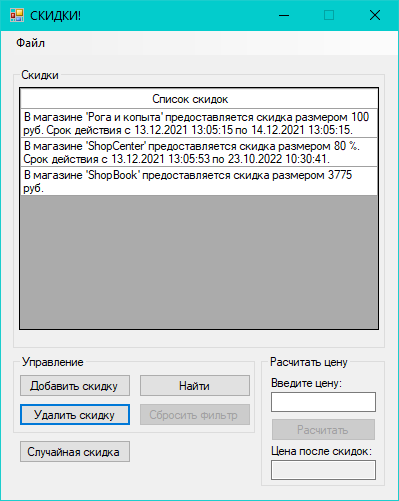
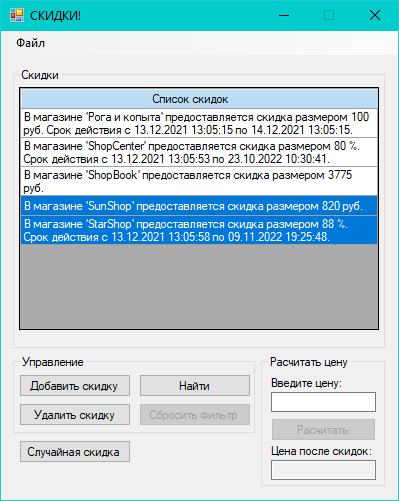


Рисунок 13 – Удаление нескольких элементов

**1.5.3 Тестовый случай «Найти элемент»**

Для поиска элементов в списке необходимо нажать кнопку «Найти». Откроется соответствующая форма для поиска (рисунок 15).

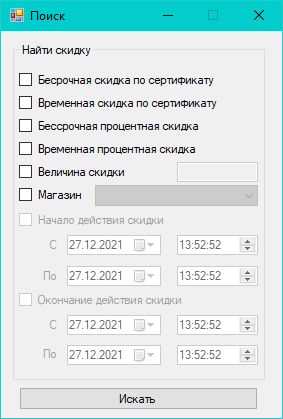
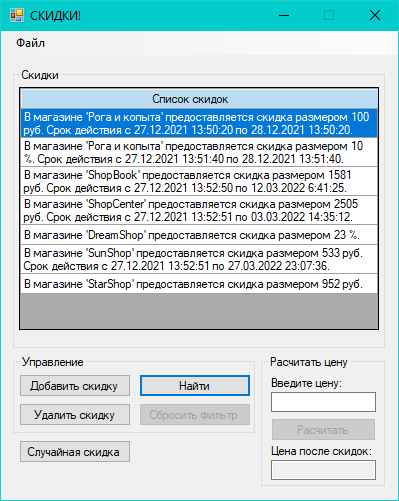
 

Рисунок 15 – Форма для поиска

Далее пользователь выбирает параметры, по которым требуется найти элемент, вводит значения этих параметров и нажимает кнопку «Искать» (рисунки 16 и 17). Для поиска скидок по магазину предусмотрен выпадающий список всех существующих магазинов из главной формы (рисунок 18).

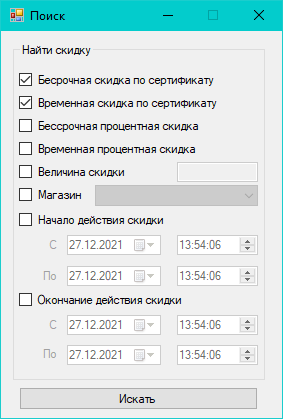
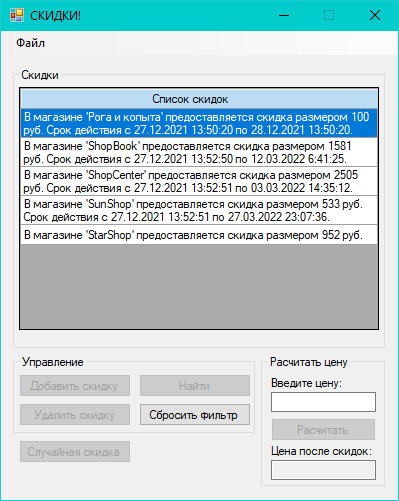
 

Рисунок 16 – Поиск элемента по типу

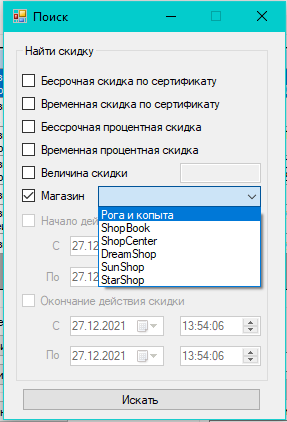
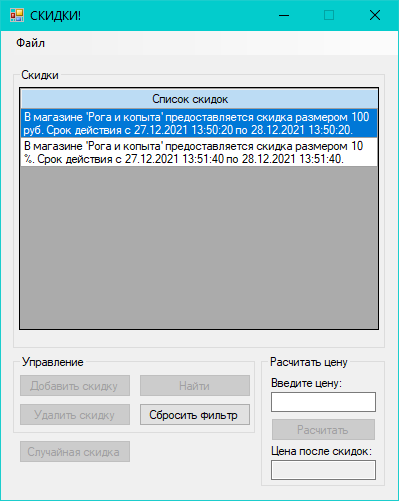
 

Рисунок 18 – Поиск по существующим магазинам

Для того, чтобы сбросить фильтр, предусмотрена соответствующая кнопка «Сбросить фильтр» (рисунок 19).

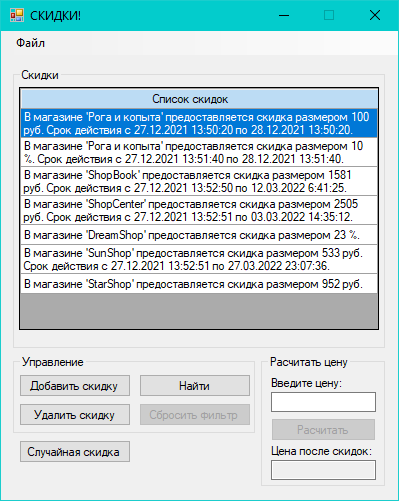


Рисунок 19 – Сброс условий поиска

**1.5.4 Тестовый случай «Рассчитать цену»**

Для расчета цены товара после применения скидки предусмотрено отдельные управляющие компоненты: «Рассчитать цену».

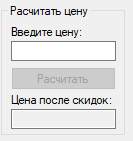


Рисунок 18 – Расчет цены товара

Пользователю необходимо ввести стоимость какого-либо товара в виде положительного числа (в том числе десятичные дроби), после этого становится доступна кнопка «Рассчитать». Расчет будет производиться для одной или нескольких выбранных скидок в таблице.

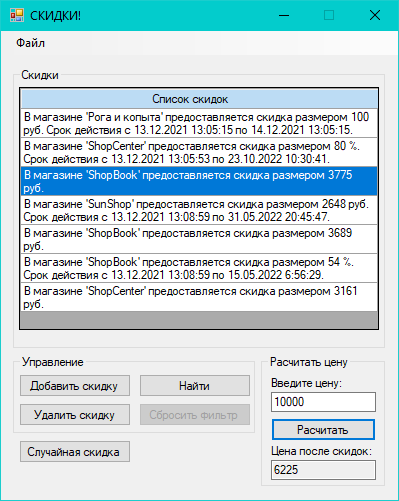


Рисунок 19 – Расчет цены одной скидки

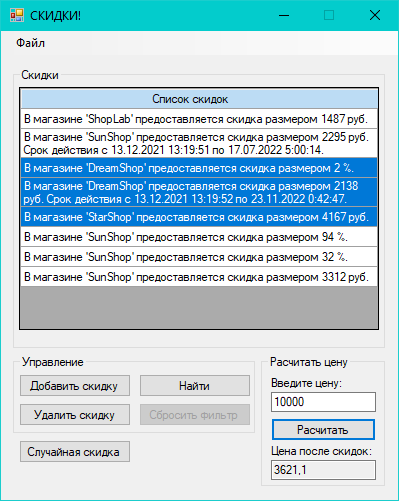


Рисунок 19 – Расчет цены нескольких скидок

**1.5.5 Тестовый случай «Сохранить данные»**

Для сохранения данных в таблице необходимо нажать на панели «Файл\Сохранить» (рисунок 20). Откроется системный диалог сохранения файла, где пользователь выбирает директорию и указывает имя файла (рисунок 21).

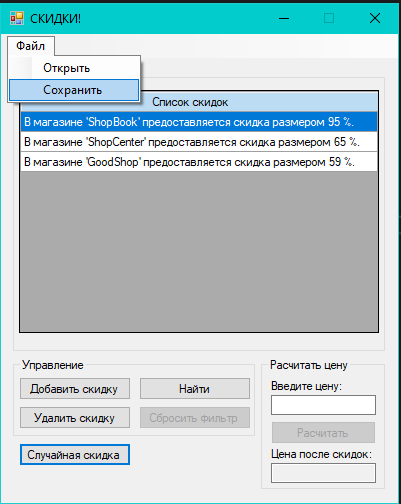


Рисунок 20 – Сохранение данных

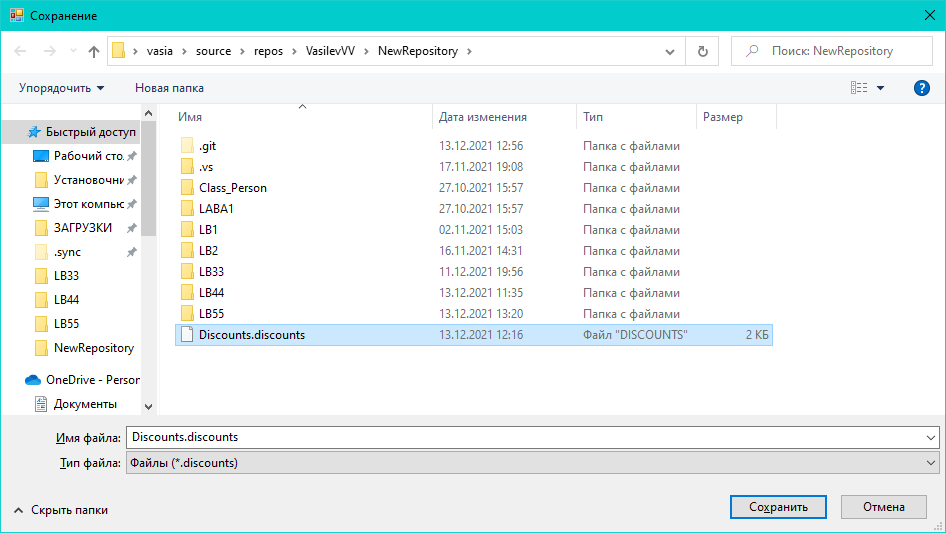


Рисунок 21 – Сохранение файла

После сохранения данных в файл появится соответствующее сообщение (рисунки 22 и 23).

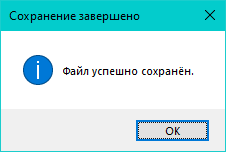


Рисунок 22 – Сообщение о сохранении файла

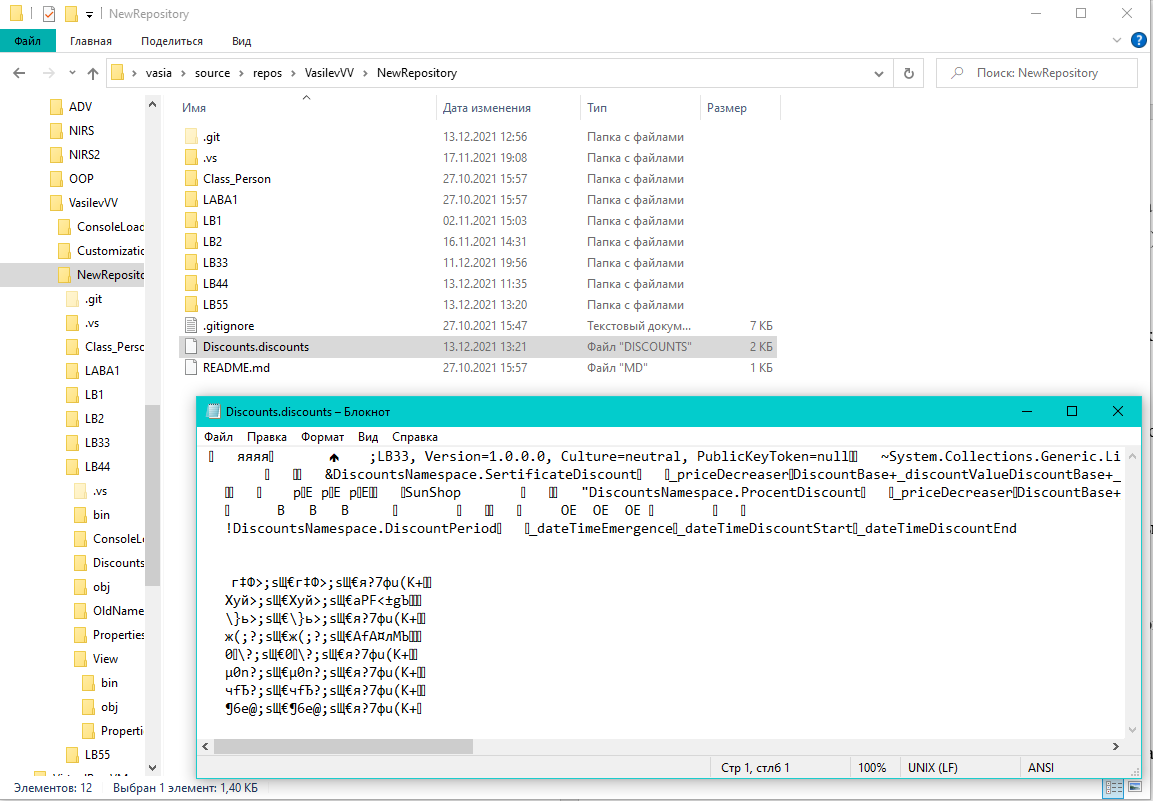


Рисунок 23 – Результат сохранения файла

В случае, если таблица пуста, сохранение не производится (рисунок 24).

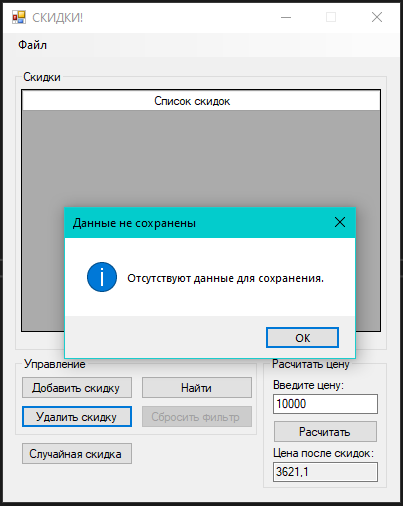


Рисунок 24 – Результат нажатия на кнопку «Сохранить данные» при пустой таблице

**1.5.6 Тестовый случай «Открыть данные»**

Для загрузки данных в таблицу необходимо нажать на соответствующую кнопку (рисунок 25).

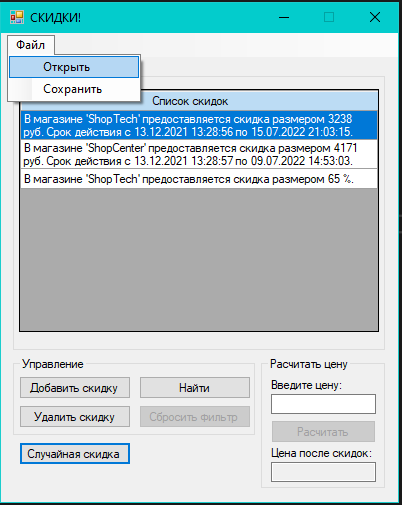


Рисунок 25 – Загрузить данные в таблицу

Далее откроется системный диалог загрузки файла (рисунок 26). После успешной загрузки появится сообщение (рисунки 27 и 28).

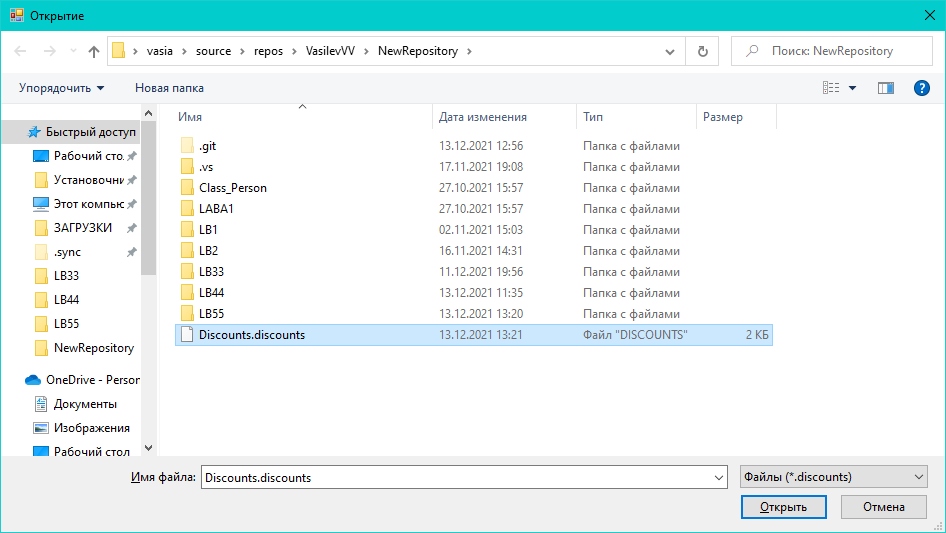


Рисунок 26 – Выбор файла для загрузки

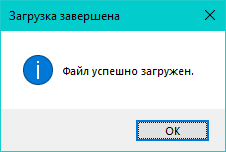


Рисунок 27 – Сообщение, информирующее об успешной загрузке файла

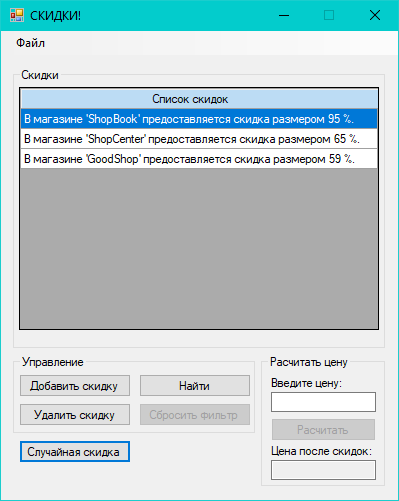


Рисунок 28 – Результат загрузки данных

В случае, если файл не соответствует установленному формату, появится соответствующее сообщение (рисунок 29).

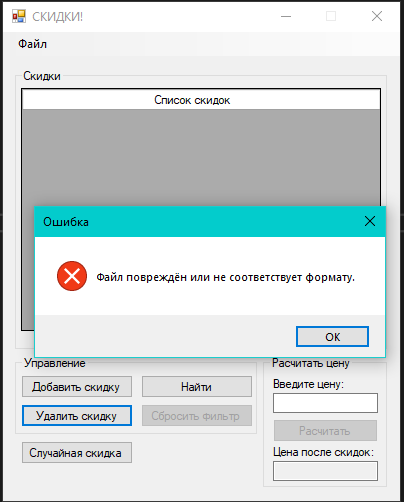


Рисунок 28 – Загрузка повреждённого файла

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Калентьев, А. А. Новые технологии в программировании : учебное пособие / А. А. Калентьев, Д. В. Гарайс, А. Е. Гориянов. – Томск : Эль Контент, 2014. – 176 с. – ISBN 978-5-4332-0185-9.

# ПРИЛОЖЕНИЕ А

**Техническое задание на создание автоматизированной системы**

Программа для расчёта скидок

Разработчик: студент гр. О-5КМ91 НИ ТПУ Васильев В. В.

Заказчик: Канд. техн. наук, доцент каф. КСУП ТУСУР Калентьев А. А.

Томск – 2021

**1 Общие сведения**

**1.1 Полное наименование системы и её условное обозначение**

Полное наименование: «Программа для расчёта скидок».

Условное обозначение: Система.

**1.2 Наименование предприятий разработчика и заказчика системы**

Заказчик: Канд. техн. наук, доцент каф. КСУП ТУСУР Калентьев А. А.

Разработчик: Студент гр. О-5КМ91 НИ ТПУ Васильев В. В.

**1.3 Плановые сроки начала и окончания работы по созданию системы**

Начало работ: 01 ноября 2021 г.

Окончание работ: 15 декабря 2021 г.

**2 Назначение и цели создания системы**

**2.1 Назначение системы**

Система предназначена для расчёта цены после применения скидок: бессрочная по сертификату, временная по сертификату, бессрочная процентная, временная процентная.

**2.2 Цели создания системы**

Система создаётся в целях сокращения трудозатрат кассиров при расчётах цены товара после применения скидки.

**3 Характеристика объектов автоматизации**

Вычисление цены товара со скидкой может пригодиться в любой сфере жизнедеятельности человека. Например, при работе в торговом центре.

Поскольку такие расчёты выполняются продавцами при выполнении продажи ежедневно, представляется целесообразным автоматизировать этот процесс.

**4 Требования к системе**

Таблица 4.1 – Префиксы мнемонических идентификаторов требований и их расшифровка

|  |  |
| --- | --- |
| Префикс | Тип требования |
| A | Архитектурное требование |
| С | Требование к программной или аппаратной совместимости |
| D | Требование к структуре данных |
| F | Функциональное требование |
| U | Требование к пользовательскому интерфейсу |

**4.1 Требования к архитектуре**

**А01** Система должна быть реализована в виде настольного приложения.

**4.2 Требования к структуре данных**

**D01** Данные о скидках должны храниться в бинарном файле с расширением \*.discounts.

**4.3 Функциональные требования**

**F01** Система должна рассчитывать объём для следующих фигур:

* бессрочная скидка по сертификату,
* временная скидка по сертификату,
* бессрочная процентная скидка,
* временная процентная скидка.

**F01.01** Временные скидки должны иметь возможность уменьшения цены только в заданный период времени:

* начало действия скидки
* окончание действия скидки

**F01.02** Цена по скидке по сертификату определяется по выражению:



где Price – цена со скидкой;

FullPrice – начальная цена;

DiscountValue – размер скидки.

**F01.03** Цена по процентной скидке определяется по выражению:



где Price – цена со скидкой;

FullPrice – начальная цена;

DiscountValue – размер скидки в процентах.

**F02** В системе должен быть реализован список заданных скидок.

**F02.01** Для каждой скидки в списке должны быть отображена информация о ней:

* тип скидки,
* величина скидки,
* название магазина, предоставляющего скидку,
* дата начала действия скидки (для временных скидок)
* дата окончания действия скидки (для временных скидок)

**F03** В системе должна присутствовать функция добавления элементов в список.

**F04** В системе должна присутствовать функция удаления элементов из списка.

**F05** В системе должна присутствовать функция поиска элементов по параметрам, указанным в **F02.01**

**F06** В системе должна присутствовать функция сохранения списка элементов в файл (**D01**).

**F07** В системе должна присутствовать функция загрузки списка элементов из файла (**D01**).

**4.4 Требования к пользовательскому интерфейсу**

**U01** Система должна иметь графический интерфейс пользователя.

**U02** Данные должны быть представлены в табличном виде.

**U03** В системе должна быть реализована система обработки ошибок.

**4.5 Требования к программному обеспечению**

**C01** Система должна работать на операционной системе Windows 10 Pro 21H1. Работоспособность на других выпусках и версиях не гарантируется.

**C02** На рабочей станции должен быть установлен .NET Framework версии 4.7.2.

**4.6 Требования к аппаратному обеспечению**

**C03** Процессор – не менее 1 ГГц.

**C04** ОЗУ – не менее 2 ГБ.