Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

(ТУСУР) Кафедра компьютерных систем в управлении и проектировании (КСУП)

**ПРОЕКТНАЯ ДОКУМЕНТАЦИЯ**

Лабораторная работа № 6

по дисциплине «Новые технологии в программировании»

Вариант № 5

Студент гр. 586-2

\_\_\_\_\_\_\_\_\_ Рогозин Р. А.

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

Руководитель

к.т.н., доц. каф. КСУП

\_\_\_\_\_\_\_\_\_ Горяинов А. Е.

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

Томск 2018

**Оглавление**

[1 Введение 3](#_Toc517087502)

[2 Техническое задание 4](#_Toc517087503)

[2.1 Исходная проблема 4](#_Toc517087504)

[2.2 Цель 4](#_Toc517087505)

[2.3 Задачи 4](#_Toc517087506)

[2.4 Конечный пользователь и контекст использования 5](#_Toc517087507)

[2.5 Критерии качества 5](#_Toc517087508)

[2.6 Дополнительные требования 5](#_Toc517087509)

[2.7 Функциональные возможности 6](#_Toc517087510)

[2.7.1 Работа с БД 6](#_Toc517087511)

[2.7.2 Работа с файлами БД 8](#_Toc517087512)

[2.8 План разработки 9](#_Toc517087513)

[3 Описание программной системы 9](#_Toc517087514)

[3.1 Описание вариантов использования 9](#_Toc517087515)

[3.2 Описание классов программы 10](#_Toc517087516)

[3.3 Разработка программы 14](#_Toc517087517)

[3.4 Система контроля версий 14](#_Toc517087518)

[3.5 Тестирование программы 15](#_Toc517087519)

[3.5.1 Модульное тестирование 15](#_Toc517087520)

[3.5.2 Функциональное тестирование 16](#_Toc517087521)

[3.6 Сборка установщика 17](#_Toc517087522)

[4 Заключение 18](#_Toc517087524)

[Список использованных источников 19](#_Toc517087525)

# **Введение**

Ведение проектной документации является неотъемлемой частью разработки любого хоть сколько-нибудь сложного и востребованного программного продукта. Основным проектным документом является техническое задание (ТЗ) — документ, описывающий требования заказчика к разрабатываемому проекту. Главное назначение ТЗ — обеспечение единого понимания всех аспектов проекта у разработчика и заказчика, что позволяет снизить число конфликтов и взаимных недопониманий в процессе разработки. В конечном счёте, грамотно составленное ТЗ сокращает сроки и расходы на реализацию проекта.

Помимо ТЗ, в состав проектной документации входят и другие документы, уточняющие или дополняющие ТЗ, например, описание архитектуры разрабатываемой программы в виде UML-диаграмм.

Данный отчёт содержит в себе ТЗ и описание программы, разработанной по нему. На рисунке 1.1 изображён графический интерфейс полученной программы.

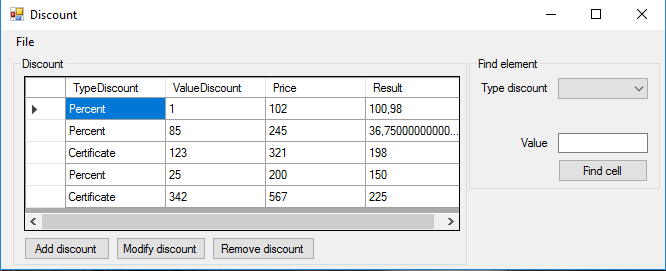


Рисунок 1.1 – Пользовательский интерфейс

# **Техническое задание**

## **Исходная проблема**

У студента, обучающегося по направлениям программирования, мало опыта в программировании.

## **Цель**

Цель данного проекта — разработать небольшую программу, рассчитывающую скидку товара различными способами, для получения навыков в программировании.

## **Задачи**

Для выполнения поставленной цели необходимо решить следующие задачи:

* Программа должна рассчитывать разную по типу скидку;
* Программа должна сохранять, считывать и редактировать данные;
* Программа должна представлять графический интерфейс взаимодействия с пользователем;
* Исходный код программы должен быть оформлен по соглашению RSDN.

## **2.4 Конечный пользователь и контекст использования**

Конечными пользователями являются люди, которые смогут оценить данный программный продукт и дать рекомендации по улучшению кода.

Критерии качества, описанные далее в п. 2.5 — небольшое потребление памяти и возможность работы на самой распространённой операционной системе (ОС) [1] — отражают то, что целевая аудитория располагает только базовыми вычислительными средствами.

## **2.5 Критерии качества**

Разработанная программа должна отвечать следующим критериям качества:

* Должна присутствовать проверка вводимых пользователем данных на корректность;
* Программа должна работать на ОС Microsoft Windows 10 Pro 64 bit;
* Время выполнения операций над БД, содержащей 50 записей, не должно превышать 3 секунд.

## **2.6 Дополнительные требования**

К разрабатываемой программе также предъявляются следующие требования:

* Разработка должна вестись в IDE Visual Studio 2017 Community на языке C#, используемая версия .NET Framework — 4.6.1;
* Исходные тексты программы должны храниться с использованием системы контроля версий Git;
* Логика программы должна быть подвергнута модульному тестированию с помощью библиотеки NUnit;
* Программа должна распространяться в виде установочного файла.

## **2.7 Функциональные возможности**

Функциональные возможности программы можно разделить на два высокоуровневых блока — собственно работа с БД и операции над файлами, хранящими в себе БД. Данные блоки можно рассматривать отдельно друг от друга: при реализации работы с файлами БД знания о структуре БД используются только при сериализации или десериализации отдельно взятой записи. Аналогично, реализация работы с записями не затрагивает работу с файлами.

## **2.7.1 Работа с БД**

Записи в БД — данные о скидке различные по двум типам: скидка по процентам, скидка по сертификату. Интерфейс программы (рисунок 1.1) состоит из нескольких окон: «главная форма», «добавить», «изменить», «поиск», «сохранить» и «открыть». При нажатии на кнопку «добавить» открывается форма для добавления (рисунок 2.1), которая имеет переключатель между типами скидки, а также поля инициализации с проверкой вводимых данных. На рисунке 2.2 изображено окно «изменить», которое позволяет изменять открытые данные выбранного элемента. Окно «поиск» изображено на рисунке 1.1 и позволяет находить данные в таблице. Окна «открыть» и «сохранить» изображены на рисунке 1.1 и представляют удобный интерфейс для перехода между файлами на компьютере. В случае отсутствия элемента в каком-нибудь окне, выводится сообщение пользователю об отсутствии данных.

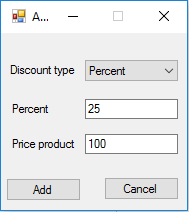


Рисунок 2.1 – Форма «добавить»

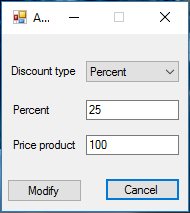


Рисунок 2.2 – Окно «изменить»

## **2.7.2 Работа с файлами БД**

Запись и открытие данных представлены с помощью диалоговых окон: «Open File Dialog» и «Safe File Dialog». В случае если открыто какое-либо диалоговое окно и необходимо отменить запись данных или открыть данные на компьютере, то можно нажать на кнопку «отменить».

## **2.8 План разработки**

Разработку программы можно разделить на этапы, соответствующие лабораторным работам курса:

– разработка бизнес-логики;

– разработка графического интерфейса и средств работы с БД;

– создание модульных тестов для бизнес-логики;

– наращивание функциональности графического интерфейса, рефакторинг исходного кода и сборка установочного файла;

– оформление документации к программе.

Время, отведённое на выполнение каждого из этапов — две недели. Всего на разработку программы отведено два с половиной месяца.

# **3 Описание программной системы**

В данном разделе приведено краткое описание проекта программы, отвечающей требованиям поставленного ТЗ, и процесса её создания.

## **3.1 Описание вариантов использования**

Диаграмма вариантов использования (use case diagram), являющаяся частью стандарта UML [2], служит для описания взаимодействия системы с действующими лицами [3]. Варианты использования специфицируют ожидаемое поведение системы (её части), описывая последовательности действий, включая их варианты, которые осуществляются системой для достижения действующим лицом определенного результата. При этом конкретная реализация функций вариантов использования не специфицируется. Диаграмма вариантов использования программы приведена на рисунке 3.1.

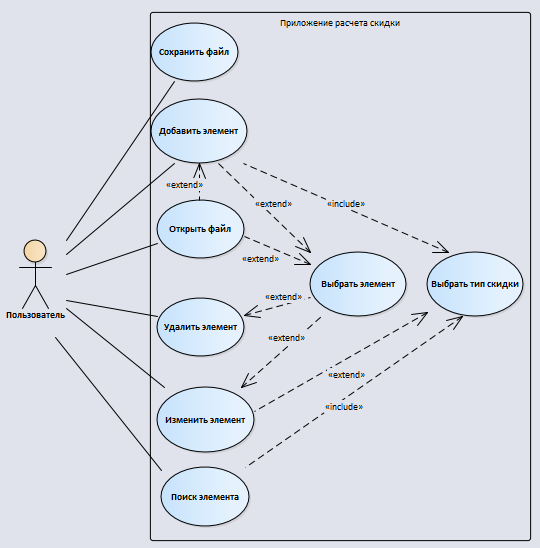


Рисунок 3.1 – Диаграмма вариантов использования программы

## **3.2 Описание классов программы**

Разработка программы велась с использованием объектно-ориентированного программирования (ООП) — подхода к программированию, основной целью которого является повторное использование существующего программного обеспечения [4] (в частности, весь графический интерфейс программы состоит из стандартных переиспользуемых компонентов). Средством реализации основной цели ООП являются абстрактные типы данных с поддержкой наследования, называемые классами.

Диаграмма классов программы служит для описания классов, входящих в систему, их статической структуры и типы взаимосвязей между классами. Множество элементов диаграммы в совокупности отражает декларативные знания о структуре системы. Высокоуровневая UML-диаграмма, отображающая связи между всеми классами программы и принадлежность их к пакетам, без указания полных списков полей и методов классов, приведена на рисунке 3.2.

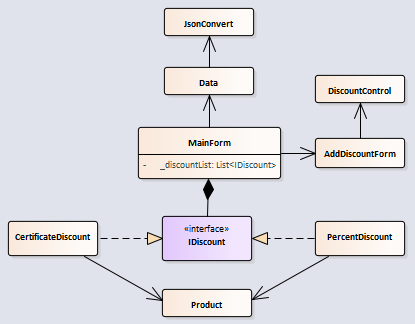


Рисунок 3.2 – UML диаграмма классов программы

Из рисунка 3.2 видно, что программа состоит из двух пакетов. Пакет View содержит в себе классы, отвечающие за графический интерфейс программы. Пакет Model содержит логику взаимодействия нескольких типов скидок, такие как «скидка по сертификату» и «скидка по процентам». На рисунке 3.3 приведена UML-диаграмма уровня реализации данного пакета, то есть с указанием всех имеющихся полей и методов классов в дополнение к их взаимоотношениям.

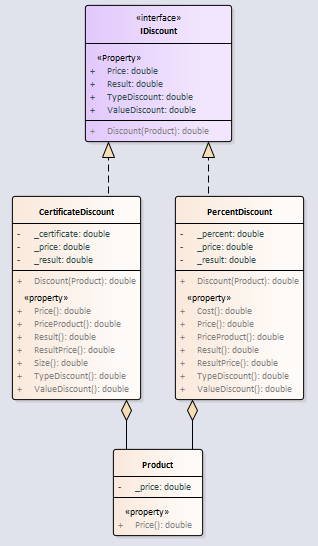


Рисунок 3.3 – Классы пакета Model

UML-диаграммы не являются единственным способом описания структуры программы. Более подробным в плане представления внутренней структуры классов является их табличное описание, где помимо имён и типов свойств, полей и методов представлены их краткие характеристики. В таблице 3.1 приведено описание интерфейса IDiscount

Таблица 3.1 – Описание интерфейса IDiscount

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание интерфейса | | |
| Интерфейс IDiscount – сущность, описывающая базовую логику производных классов. | | |

|  |  |  |
| --- | --- | --- |
| Свойства | | |
| + TypeDiscount | double | Тип скидки |
| + ValueDiscount | double | Значение скидки |
| + Price | double | Цена товара |
| + Result | double | Цена товара с учетом скидки |
| Методы | | |
| + Discount(Product) | double | Вычисление цены товара с учетом скидки |

Окончание таблицы 3.1

Каждый из классов, реализующих данный интерфейс, для своей работы требует использования дополнительных свойств, полей и методов, не специфицируемых интерфейсом. Описание класса CertificateDiscount приведено в таблице 3.2.

Таблица 3.2 – Описание класса CertificateDiscount

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс CertificateDiscount – сущность, описывающая скидку по сертификату. | | |
| Методы | | |
| + Discount(Product) | double | Вычисление цены товара с учетом сертификата |
| Свойства | | |
| + Size | double | Значение сертификата |
| + PriceProduct | double | Цена товара |
| + ResultPrice | double | Цена товара с учетом скидки |
| + TypeDiscount | double | Тип скидки |
| + ValueDiscount | double | Вернуть сертификат |

Окончание таблицы 3.2

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| + Price | double | Вернуть цену товара |
| + Result | double | Вернуть цену товара с учетом скидки |

Описание класса PercentDiscount приведено в таблице 3.3.

Таблица 3.3 – Описание класса PercentDiscount

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс PercentDiscount – сущность, описывающая скидку по процентам. | | |
| Методы | | |
| + Discount(Product) | double | Вычисление цены товара с учетом процентной скидки |
| Свойства | | |
| + Cost | double | Значение скидки |
| + PriceProduct | double | Цена товара |
| + ResultPrice | double | Цена товара с учетом скидки |
| + TypeDiscount | double | Тип скидки |
| + ValueDiscount | double | Вернуть сертификат |
| + Price | double | Вернуть цену товара |
| + Result | double | Вернуть цену товара с учетом скидки |

## **3.3 Разработка программы**

Разработка программы велась с использованием следующего стека технологий:

– язык C# 7.0, .NET Framework 4.6.1;

– IDE MS Visual Studio 2017 Community;

– система контроля версий git 2.5.2;

– библиотека модульного тестирования NUnit 3.10.1.

Для каждого свойства, поля и метода каждого класса в коде имеется XMLкомментарий, кратко характеризующий его назначение, что облегчает дальнейшее переиспользование кода программы.

## **3.4 Система контроля версий**

Для хранения исходного кода программы на сервисе GitHub был создан git репозиторий NTVP\_lab\_1, расположенный по адресу <https://github.com/Rogozilio/NTVP_lab_1>.

На рисунке 3.4 приведены все одиннадцать коммитов в ветке develop репозитория. Данное представление получено с помощью дополнения Git for Visual Studio, входящее в диспетчер пакетов NuGet в VS.

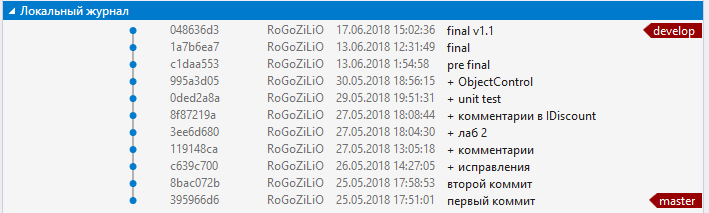


Рисунок 3.4 – Все коммиты в репозитории.

## **3.5 Тестирование программы**

Тестирование программы позволяет убедиться, что реальное поведение программы соответствует ожидаемому, описанному в документации. Своевременное тестирование даёт возможность выявлять ошибки на ранних стадиях разработки, что значительно снижает затраты на их устранение и в целом упрощает разработку и дальнейшую поддержку программы.

## **3.5.1 Модульное тестирование**

Модульное тестирование — процесс изолированного тестирования отдельных модулей программ для подтверждения корректности их работы. В данной программе для классов пакета Model — CertificateDiscount, PercentDiscount и Product — с использованием библиотеки модульного тестирования NUnit были написаны тесты для всех публичных свойств и методов. Были использованы как позитивные тесты, проверяющие правильность работы классов на корректных данных, так и негативные, проверяющие правильность типа выдаваемых классом исключений при подаче на вход некорректных данных. Общее количество написанных тестовых случаев — 17. На рисунке 3.5 приведена иерархия тестов, отображаемая в обозревателе тестов VS.

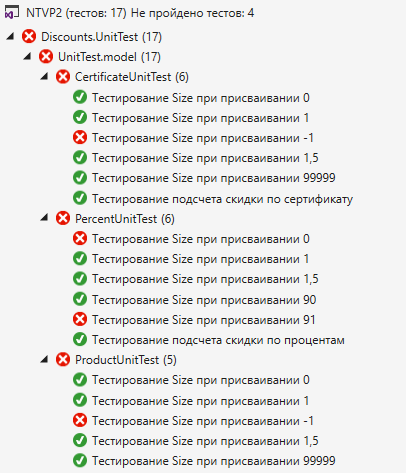


Рисунок 3.5 – Иерархия модульных тестов.

## **3.5.2 Функциональное тестирование**

Функциональное тестирование предназначено для проверки реализуемости в программе функциональных требований к ней, заданных ТЗ. Проверка вводимых пользователем данных о разных по типам зарплат не даёт создавать экземпляры зарплат с некорректными данными — при нажатии на кнопку «добавить» появляется окно с описанием ошибки. Примеры ввода некорректных данных приведены на рисунке 3.6.

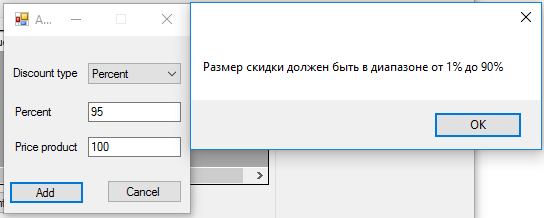


Рисунок 3.6 – Ввод некорректных данных.

## **3.6 Сборка установщика**

Для сборки установщика программы была использована свободно распространяемая утилита с открытым исходным кодом InnoSetup [5]. С её помощью был создан установочный файл, размещающей в заданном каталоге. По желанию пользователя также возможно создание ярлыка для исполняемого файла программы на рабочем столе.

# **4 Заключение**

В ходе лабораторных работ курса «Новые технологии в проектировании» была создана программа, полностью отвечающая требованиям, описанным в ТЗ на неё: все поставленные для достижения цели задачи выполнены. Программа полностью соответствует заданным критериям качества и прочим требованиям и, соответственно, подходит для решения исходной проблемы из ТЗ. Написанная программа также сопровождается документацией. Проведённое тестирование — модульное и функциональное — подтверждает правильность поведения программы в том числе в тех случаях, когда в интерфейс программы вводятся заведомо некорректные данные. Так как в проектной документации присутствует описание архитектуры программы, а весь её исходный текст прокомментирован, её расширение или повторное использование классов в других проектах не вызовет больших сложностей. В процессе разработки большое количество времени ушло на изучение работы с библиотеками, в частности с NUnit и Newtonsoft.Json. Также было изучен элемент UserControl.

# **Список использованных источников**

1. Desktop Windows Version Market Share Worldwide | StatCounter Global Stats. [Электронный ресурс]. — Режим доступа: http://gs.statcounter.com/os-version-marketshare/windows/desktop/worldwide (дата обращения: 15.06.2018).

2. About the Unified Modeling Language Specification Version 2.5. [Электронный ресурс]. — Режим доступа: http://www.omg.org/spec/UML/About-UML/ (дата обращения 15.06.2018).

3. Новые технологии в программировании : учебное пособие / Д. В. Гарайс, А. Е. Горяинов, А. А. Калентьев. — Томск : Эль Контент, 2014. — 176 с.

4. Полное руководство по языку программирования С# 7.0 и платформе .NET 4.7. [Электронный ресурс]. — Режим доступа: https://metanit.com/sharp/tutorial/ (дата обращения 15.06.2018).

5. Inno Setup. [Электронный ресурс]. — Режим доступа: http://www.jrsoftware.org/isinfo.php (дата обращения 15.06.2018).