Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение

«Политехнический колледж городского хозяйства»

**ОТЧЕТ**

**Учебная практика**

по профессиональному модулю

**ПМ.02 Осуществление интеграции программных модулей**

|  |  |
| --- | --- |
| Руководитель практики: Солопова Елена Владиславовна  ФИО | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |
| Отчет выполнил студент группы: ИП-21-3 | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Пономарев Андрей Викторович |
|  | ФИО |

2025

**Содержание**

*Раздел 2.2:*

* *представить алгоритмы подсистемы для работы с объектами системы*
* *описать поэтапную технологию создания объектов приложения: окон, формы со списком и других форм, таблиц и т.п. (с приведением кодов), включая интерфейс программного модуля для работы с клиентами/товарами/материалами (функции добавления/редактирования)*

На диске **Х:** (своей учетной записи) создать папку **УП.02\_Фамилия**

В папку **УП.02\_Фамилия** поместить все **результаты выполнения заданий практики:**

1. **файл** итогового отчета по практике **ОтчётУП.02\_Фамилия** (включить ссылку на **Gogs)**
2. **папку с выполненным решением задания 1 по ДЭМО 2025 года, включая** скрипт БД с данными
3. **папку с выполненным решением задания 2:** скрипт БД с данными (сформировать из MS SQL), папку с приведенными данными для импорта (например, папка *ГотовыеДанныеДляИмпорта* и скопировать в неё приведенные готовые файлы *.xlsx* для импорта), папку с решением своего проекта приложения, руководство пользователя, Test-case.

**1. Введение**

ООО «Игора» - Прокат спортивного инвентаря.

В рамках выполнения задания необходимо разработать основные модули информационной системы для точки проката горнолыжного курорта «Игора» в Ленинградской области:

* неавторизованный клиент и авторизованный клиент может просматривать предоставляемые услуги и формировать заказы;
* менеджер может просматривать услуги, добавлять и редактировать клиентов, подтверждать и редактировать заказы;
* администратор может просматривать/добавлять/редактировать/удалять услуги.

Кроме того, разрабатываемая информационная система предполагает установку на терминалах в точках прокатах. На терминале клиент (авторизованный и неавторизованный) может просмотреть услуги и сформировать заказ.

**Общая часть**

**Разработка базы данных, объектов БД и импорт данных**

СУБД для создания базы данных: SQL Server Management studio.

Логин для входа: 975-21;

Пароль для входа: 7snhf12K;

Пошаговая инструкция запуска и просмотра БД:

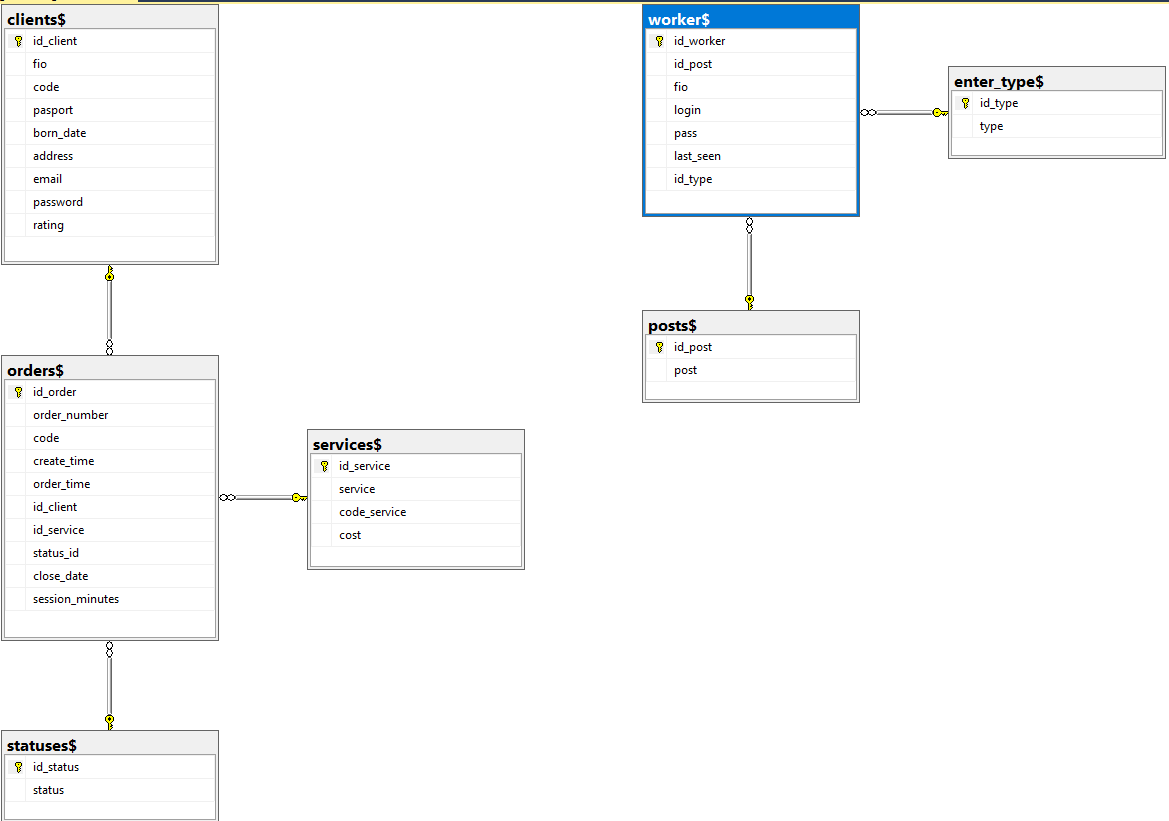
1. Запустить SSMS;
2. Войти используя данные, указанные выше;
3. Раскрыть пункт «gogs.wsr.ru»;
4. Раскрыть пункт «Базы данных»;
5. Найти базу данных с названием IgoraDemo и раскрыть;
6. Раскрыть пункт «Таблицы»;
7. Выбрать интересующую таблицу, нажать ПКМ по ней и выбрать пункт «Выбрать первые 1000 строк».

Пошаговая инструкция импорта данных БД:

Пошаговая инструкция экспорта данных БД:

1. Выполнить выход, найти нужную БД;
2. Нажать ПКМ по названию БД;
3. Нажать «Задачи»;
4. Нажать «Сгенерировать скрипты»;
5. Нажать «Далее», затем «Далее»;
6. В настройках выбрать «Сохранить как скрипт»;
7. Нажать «Продвинутые», там найти пункт «Типы данных скрипт» - выбрать «Данные и таблицы», нажать «ОК»;
8. Выбрать место куда сохранится скрипт;
9. Завершить импорт.

**Схема базы данных**



**Разработка модулей программного обеспечения для компьютерных систем**

Среда разработки: Microsoft Visual Studio, C# .NET Framework 4.8, WinForms;

Данные для входа как клиент: логин – client; пароль – client.

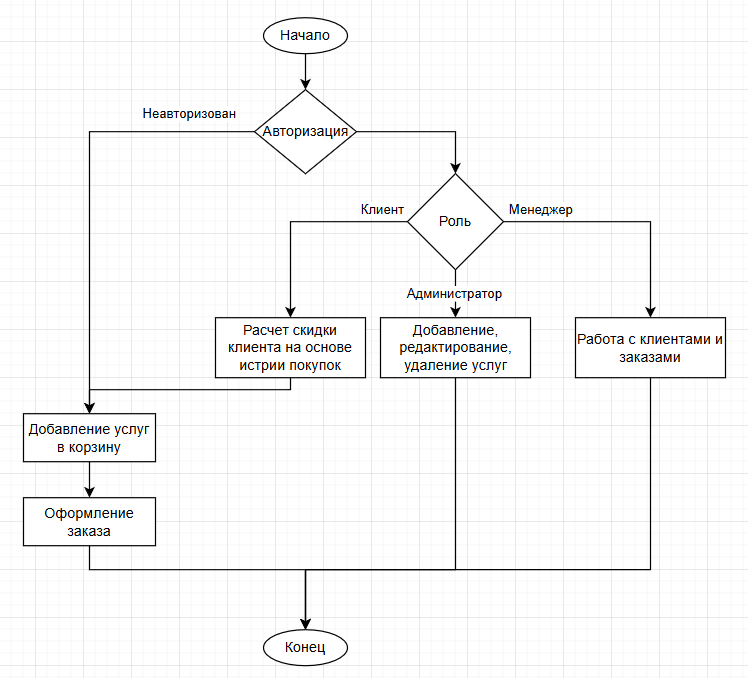
Данные для входа как менеджер: логин – manager; пароль – manager;

Данные для входа как администратор: логин – admin; пароль – admin;

Запуск приложения:

1. Зайти в папку с проектом.
2. Открыть папку IgoraDemo.
3. Открыть папку bin.
4. Открыть папку Debug.
5. Запустить приложение «IgoraDemo.exe»

**Алгоритм решения задачи**

****

Функция расчета индивидуальной скидки:

public static int calculateDiscount(int sum)

{

if (sum < 50\_000)

{

return 0;

}

else if (sum >= 50\_000 && sum < 100\_000)

{

return 1;

}

else if (sum >= 100\_000 && sum < 300\_000)

{

return 3;

}

else

{

return 5;

}

}

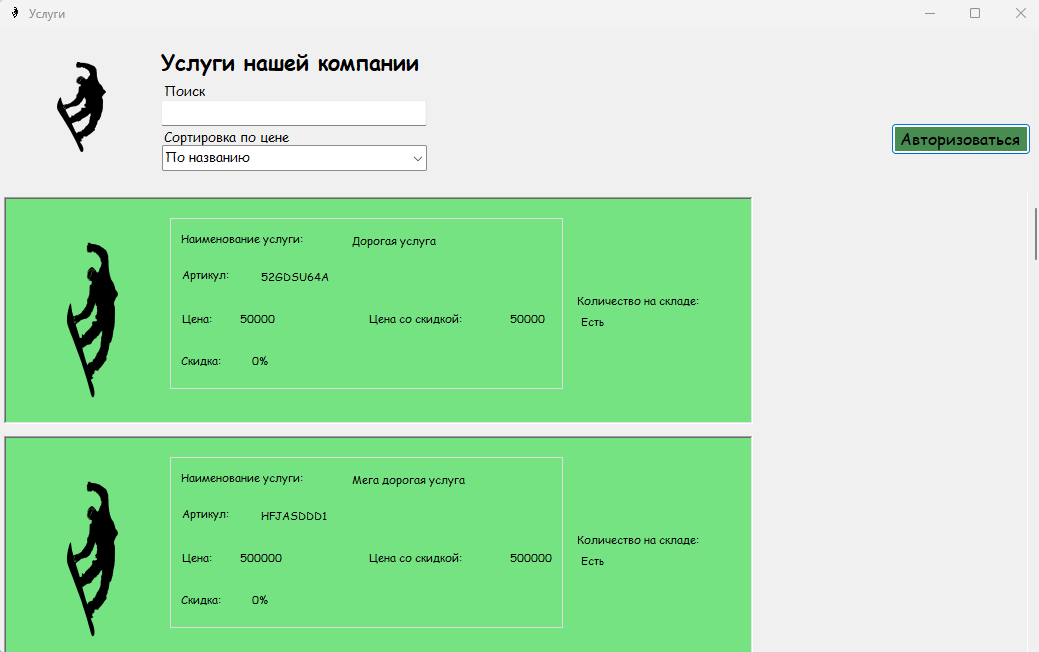
На вход подается количество потраченных денег клиентом, возвращается скидка величиной от 0% до 5% в виде целочисленного значения.

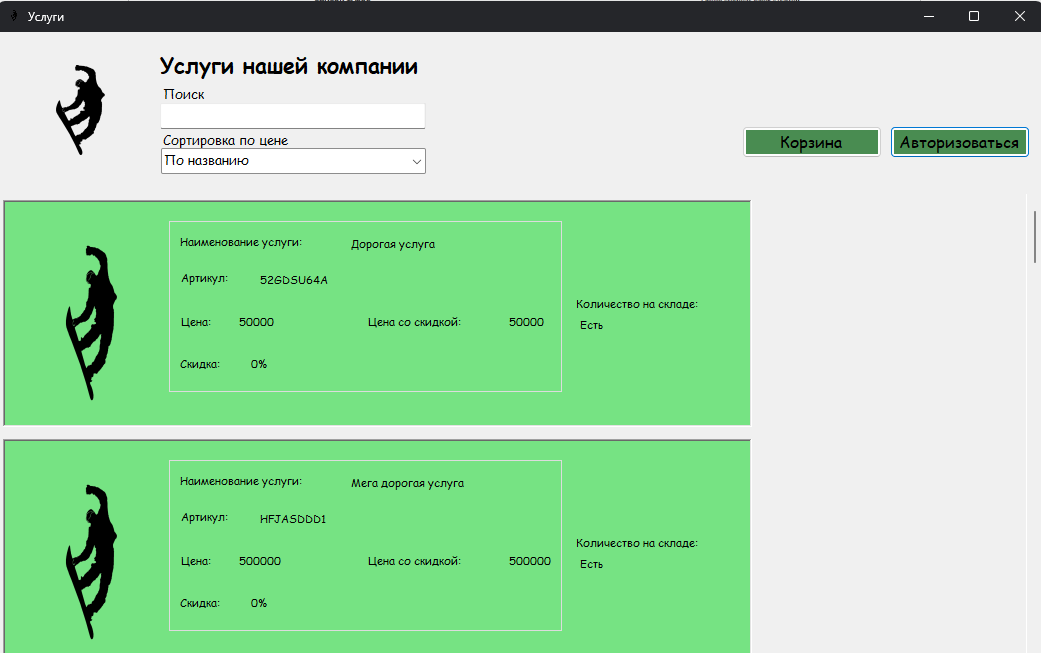
Скидка вычисляется в зависимости от потраченных денег: до 50000 – 0%, от 50000 – до 100000 – 1%, от 100000 – до 300000 – 3%, более 300000 – 5%.

**Разработка desktop-приложения**

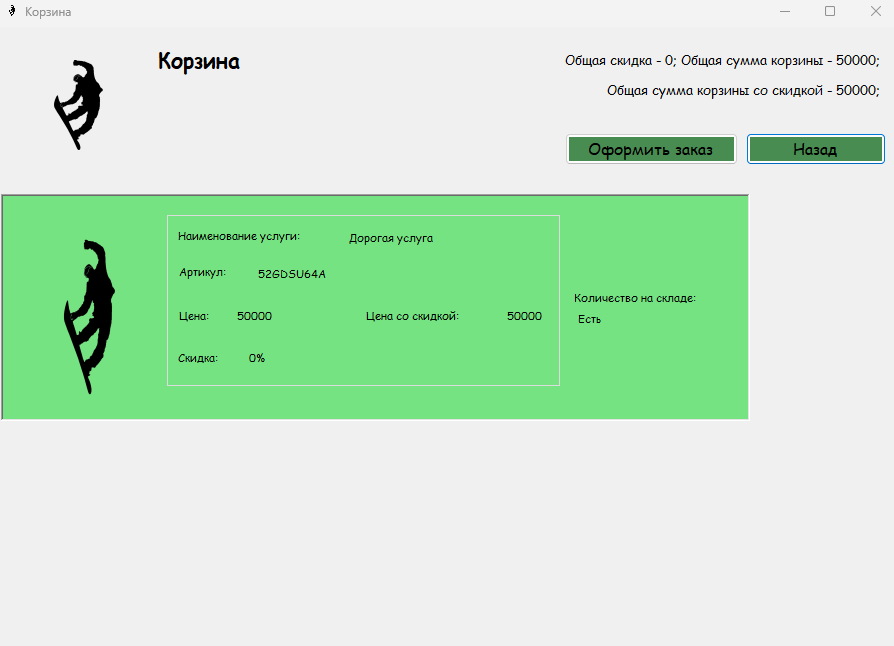
Ссылка на репозиторий - .

Главная форма открывается при запуске приложения. На ней пользователь может просмотреть услуги и сформировать заказ. Также возможен поиск по названию услуги и сортировка по цене. После добавления в корзину появляется кнопка «Корзина».

****

****

В корзине отображаются все услуги, которые были добавлены в корзину, а также общая скидка и сумма заказа. По нажатии кнопки «Оформить заказ» можно оформить заказ и данные о нем занесутся в базу данных

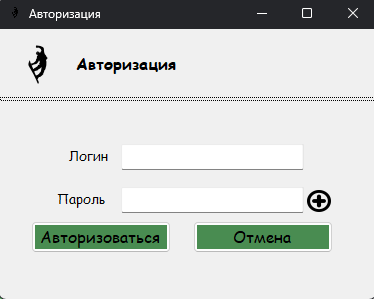
****

Окно авторизации позволяет заполнить данные для входа.

Данные для входа клиента: логин - client; пароль - client;

Данные для входа менеджера: логин - manager; пароль - manager;

Данные для входа администратора: логин – admin; пароль – admin;



**Тестирование приложения**

Тестирование валидации данных создания нового клиента:

**Тест №1:** Добавление пользователя проходит успешно, без потери данных и с валидацией пользовательского ввода.

Входные данные:

* ФИО – qwe qwe qwe
* Код - 12312312
* Паспорт – 1212121212
* Дата рождения – текущая дата
* Адрес – йцу
* Почта – qwe
* Пароль – qwe123QWE123
* Рейтинг - 5

Ожидаемый результат: сообщение о неверном формате введённой почты без аварийного завершения.

Фактический результат: сообщение о неверном формате введённой почты без аварийного завершения.

Результат: Тест пройден.

**Тест №2:** Добавление пользователя проходит успешно, без потери данных и с валидацией пользовательского ввода.

Входные данные:

* ФИО – qwe
* Код - 12312312
* Паспорт – 1212121212
* Дата рождения – текущая дата
* Адрес – йцу
* Почта – qwe@bk.ru
* Пароль – qwe123QWE123
* Рейтинг - 5

Ожидаемый результат: сообщение о неверном формате введённого ФИО без аварийного завершения.

Фактический результат: сообщение о неверном формате введённого ФИО без аварийного завершения.

Результат: Тест пройден.

**Тест №3:** Добавление пользователя не проходит если в БД уже есть такой клиент.

Входные данные:

* ФИО – qwe qwe qwe
* Код - 12312312
* Паспорт – 1212121212
* Дата рождения – текущая дата
* Адрес – йцу
* Почта – qwe@bk.ru
* Пароль – qwe123QWE123
* Рейтинг – 5

Примечание: в базе данных уже должна быть запись с кодом 12312312 и паспортом 1212121212.

Ожидаемый результат: сообщение о невозможности повтора одинаковых клиентов.

Фактический результат: сообщение о невозможности повтора одинаковых клиентов.

Результат: Тест пройден.

**Тест №4:** Добавление услуги не проходит, если указанный артикул существует в БД.

Входные данные:

* Название – Прокат красного сноуборда
* Артикул – SHDDT123A
* Цена - 300

Примечание: в базе данных уже должна быть запись с артикулом SHDDT123A.

Ожидаемый результат: сообщение о невозможности повтора одинаковых артикулов.

Фактический результат: сообщение о невозможности повтора одинаковых артикулов.

Результат: Тест пройден.

Тестирование подсистемы расчета скидки:

**Тест №1:** Скидка для клиента с суммой выкупа 40000 рублей

Входные данные: Сумма выкупа 40000.

Ожидаемый результат: функция вернёт 0.

Фактический результат: функция вернула 0.

Результат: Тест пройден.

**Тест №2:** Скидка для клиента с суммой выкупа 51000 рублей

Входные данные: Сумма выкупа 51000.

Ожидаемый результат: функция вернёт 1.

Фактический результат: функция вернула 1.

Результат: Тест пройден.

**Тест №3:** Скидка для клиента с суммой выкупа 150000 рублей

Входные данные: Сумма выкупа 150000.

Ожидаемый результат: функция вернёт 3.

Фактический результат: функция вернула 3.

Результат: Тест пройден.

**Тест №4:** Скидка для клиента с суммой выкупа 200000 рублей

Входные данные: Сумма выкупа 200000.

Ожидаемый результат: функция вернёт 5.

Фактический результат: функция вернула 5.

Результат: Тест пройден.

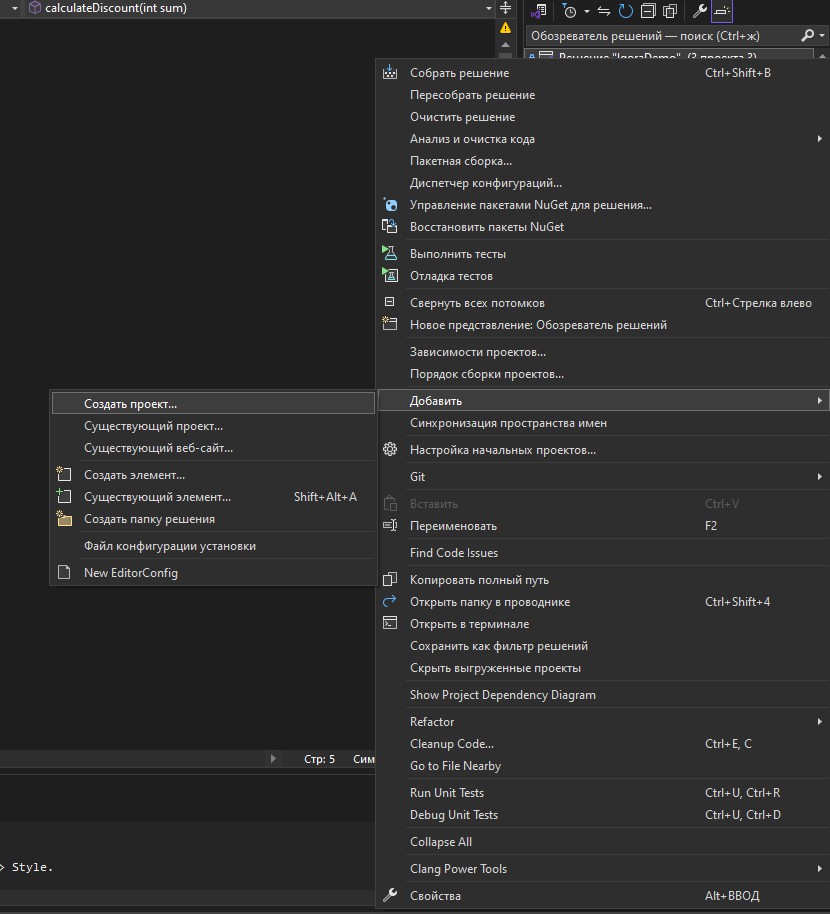
**Сопровождение и обслуживание программного обеспечения компьютерных систем**

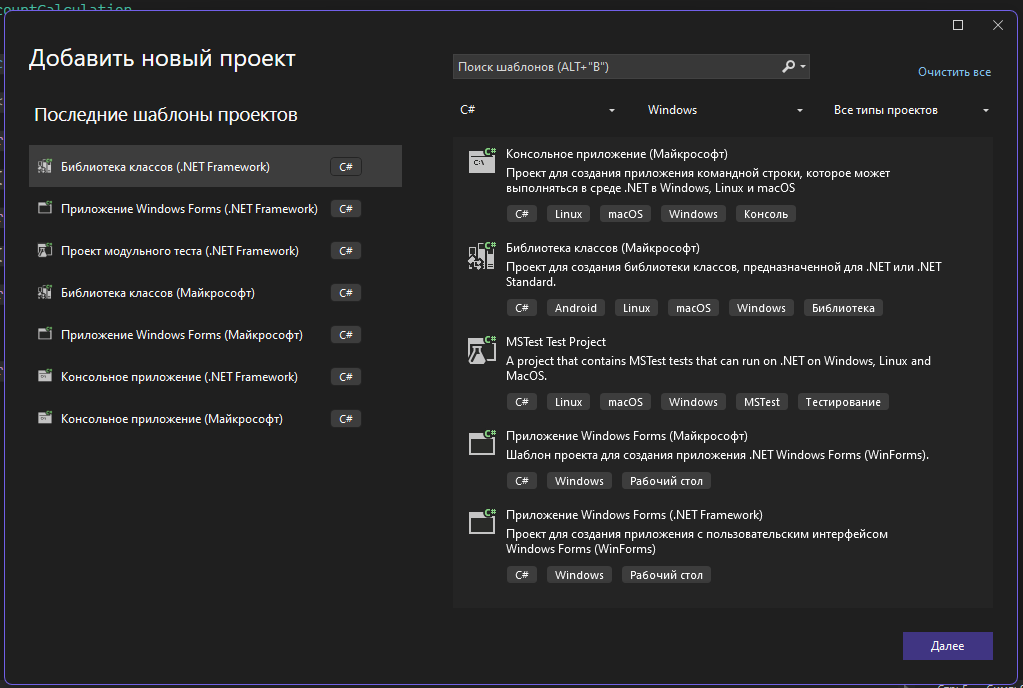
**Программный модуль для работы с заказами**

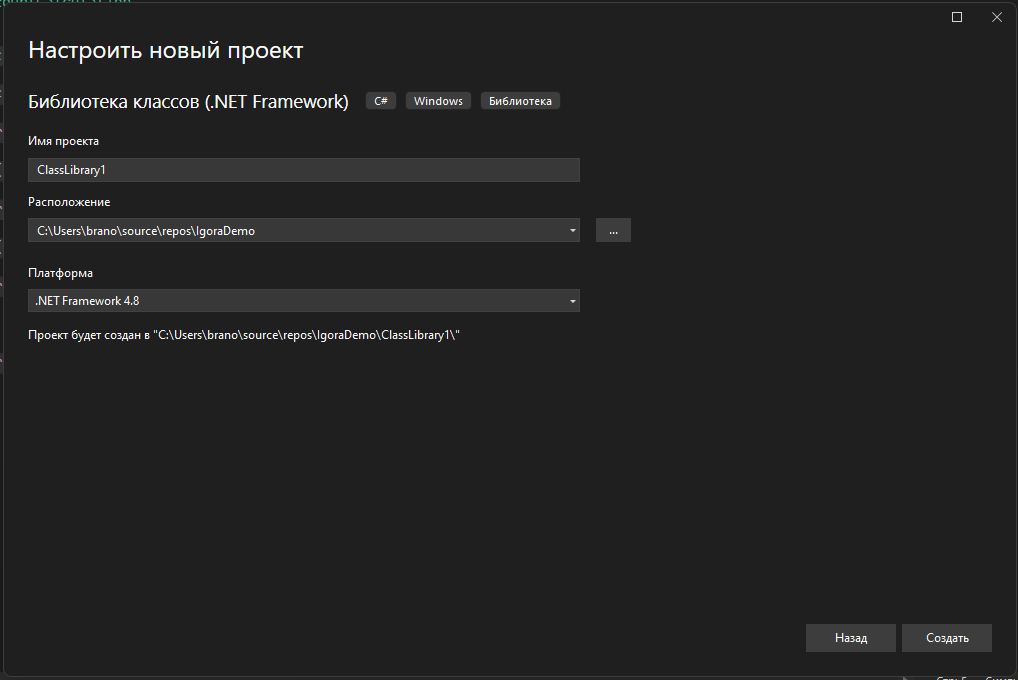
**Осуществление интеграции программных модулей**

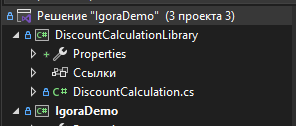
**Разработка библиотеки и подключение её к проекту**

Для создания библиотеки необходимо нажать ПКМ по решению -> выбрать «Добавить» -> «Создать проект» -> «Библиотека классов (.NET Framework)» -> Выбрать название и создать проект.

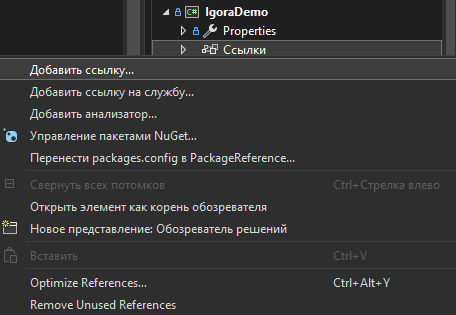


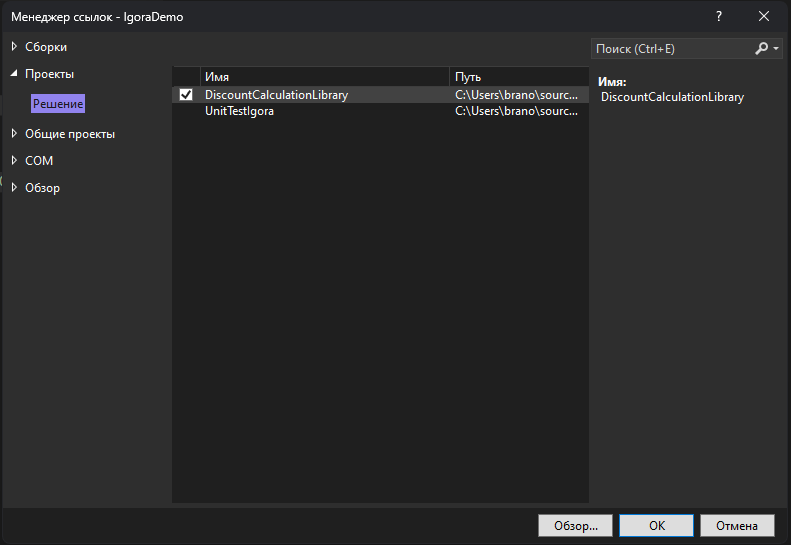


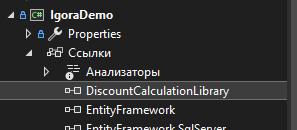




Для создания ссылки на проект (связывания двух проектов) необходимо в другом проекте нажать ПКМ по кнопку «Ссылки» -> «Добавить ссылку» -> «Проекты» -> Пометить галочкой нужный проект для создания ссылки. Далее можно будет использовать библиотеку так, словно она часть проекта.







**Представление результатов работы**

**Заключение**

Разработка информационной системы для точки проката спортивного инвентаря ООО «Игора» позволит оптимизировать процессы взаимодействия с клиентами и повысить эффективность работы персонала. Система обеспечит удобный доступ к информации о предоставляемых услугах как для авторизованных, так и для неавторизованных пользователей, а также упростит процесс формирования заказов через веб-интерфейс и терминалы на точках проката.

Для сотрудников компании система предоставит необходимые инструменты для управления клиентской базой, обработки заказов и контроля услуг. Возможность администрирования контента позволит гибко адаптировать ассортимент и актуализировать данные о доступных услугах.

Внедрение данной информационной системы станет важным шагом в цифровизации деятельности ООО «Игора», что обеспечит повышение качества обслуживания клиентов, сокращение времени на выполнение рутинных операций и создаст дополнительные конкурентные преимущества для горнолыжного курорта.

**Список использованных источников**

**Приложение А**

**Код библиотеки:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace IgoraDemo.Services

{

internal class DiscountCalculation

{

public static int calculateDiscount(int sum)

{

if (sum < 50\_000)

{

return 0;

}

else if (sum >= 50\_000 && sum < 100\_000)

{

return 1;

}

else if (sum >= 100\_000 && sum < 300\_000)

{

return 3;

}

else

{

return 5;

}

}

}

}