**Введение**

Учебная практика проходила в организации «СофтСервис», которая специализируется на разработке программных решений на базе платформы 1С. Основными направлениями деятельности компании являются автоматизация управленческого и бухгалтерского учета, разработка CRM-систем, построение систем управления предприятием (ERP), а также создание специализированных решений для различных отраслей бизнеса.

На учебную практику по программированию была поставлена задача разработка консольного приложения «Гостиница», позволяющего хранить, редактировать и удалять информацию о клиентах и номерах в гостинице, истории бронирования и отслеживать финансы.

Ниже приведено краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ предметной области и формулировка требований к программе». Он посвящён изучению предметной области и организационно-экономической сущности поставленной задачи. В подразделе «Инструменты разработки» будет рассмотрена среда, в которой создаётся данный проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачи.

В разделе «Проектирование» будут рассмотрены основные аспекты разработки консольного приложения. Здесь можно ознакомиться с диаграммой использования и диаграммой деятельности. Будет представлено описание тестов.

«Построение программы» – это третий раздел отчёта по практике, в котором описываются классы и методы программы.

Четвёртый раздел – «Тестирование». В нем будет описано полное и функциональное тестирование данной программы. Будут смоделированы все возможные действия пользователя при работе с приложением.

В разделе «Применение» будет описано назначение и условия применения программного продукта. А также руководство по использованию.

«Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, степени автоматизации процессов на различных этапах разработки.

В разделе «Список использованных источников» будет приведён список используемых при разработке источников.

В приложении А будет предоставлен листинг программы.

**1** **Анализ предметной области и формулировка требований к программе**

**1.1 Исследование предметной области**

Наименование задачи: приложение «Гостиница».

Целью разработки данного программного продукта является создание системы с понятным и эффективным интерфейсом. Приложение должно обеспечивать возможность добавления, редактирования, поиска, просмотра и удаления информации о клиентах, номерах и бронировании в гостинице, а также просмотра отчёта о выручке за период времени.

Создаваемая программа будет рассчитана на работников гостиницы.

Периодичность использования: по мере необходимости.

**1.2 Инструменты разработки**

Инструменты, используемые при разработке приложения и написании документации:

* Microsoft Visual Studio – это редактор исходного кода, в котором было написано приложение;
* Microsoft Office Word – для написания документации к программному продукту;
* Web-ресурс Draw.io – сайт, позволяющий создавать и редактировать схематичные объекты (схемы, макеты, диаграммы), который будет использоваться для создания графической части и разработки UML-диаграмм.

Разработка проекта будет происходить на компьютере со следующими параметрами:

* + процессор Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz;
  + видеокарта NVIDIA GeForce MX130;
  + объем оперативной памяти 8.00 GB;
  + объем места на жестких дисках 237 GB;
  + ОС Windows 11 Pro.

**2. Проектирование задачи**

**2.1 Диаграмма вариантов использования**

Диаграмма вариантов использования (Use Case Diagram) - это графическое представление функциональности системы, которое показывает, как различные типы пользователей взаимодействуют с системой для выполнения определенных целей (вариантов использования). Эта диаграмма помогает понять, как система будет использоваться из-за взаимодействия между пользователями и системой.

Основные элементы диаграммы вариантов использования включают:

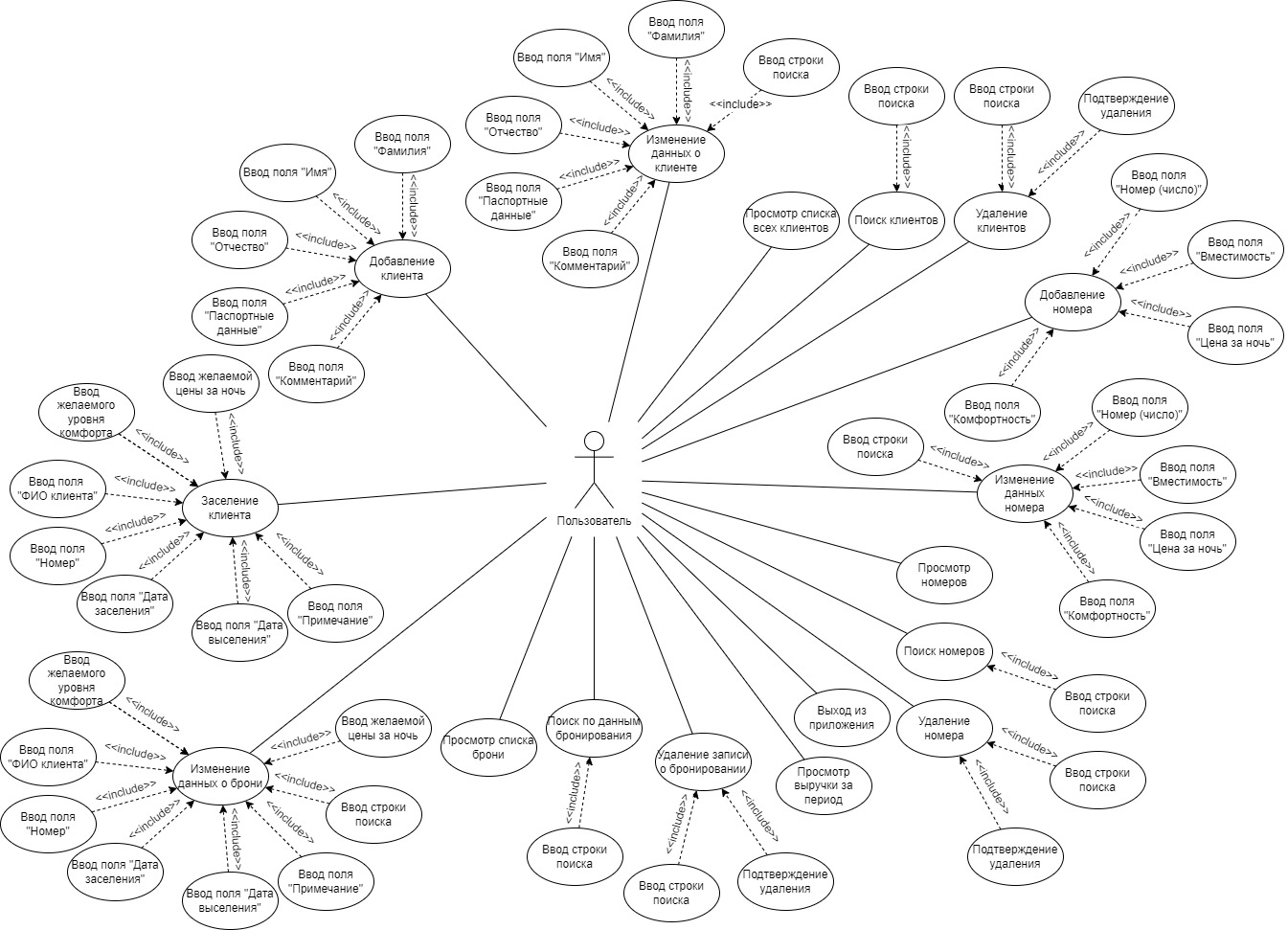
Актеры (Actors): Это пользователи или внешние системы, взаимодействующие с системой. Актеры обычно представлены в виде стикеров с именами рядом с системой.

Варианты использования (Use Cases): Это конкретные сценарии использования системы. Каждый вариант использования описывает одну функциональность, которую система предоставляет актеру. Варианты использования представлены овалами с названиями внутри.

Отношения между актерами и вариантами использования: Стрелки, показывающие, какие варианты использования доступны для каждого актера.

Диаграммы вариантов использования помогают команде проектирования системы понять, как пользователи будут взаимодействовать с системой, и определить функциональные требования для разработки. Они также могут использоваться для визуализации и обсуждения функциональности системы с заинтересованными сторонами.

Диаграмма вариантов использования, разработанная для данного проекта, представлена на рисунке 1.

Рисунок 1 – Диаграмма вариантов использования

**2.2 Диаграмма деятельности**

Диаграмма деятельности (Activity Diagram) – это графическое представление процесса или алгоритма, показывающее последовательность действий или действий и решений в системе или программе. Она широко используется для моделирования бизнес-процессов, описания работы программного обеспечения или даже поведения пользователей в системе.

Диаграмма деятельности состоит из следующих элементов:

Действия (Actions): Основные шаги или действия, выполняемые в процессе. Они представлены прямоугольниками с названиями действий.

Старт (Start): Начальная точка процесса, обозначающая его начало. Представляется окружностью с надписью "Start".

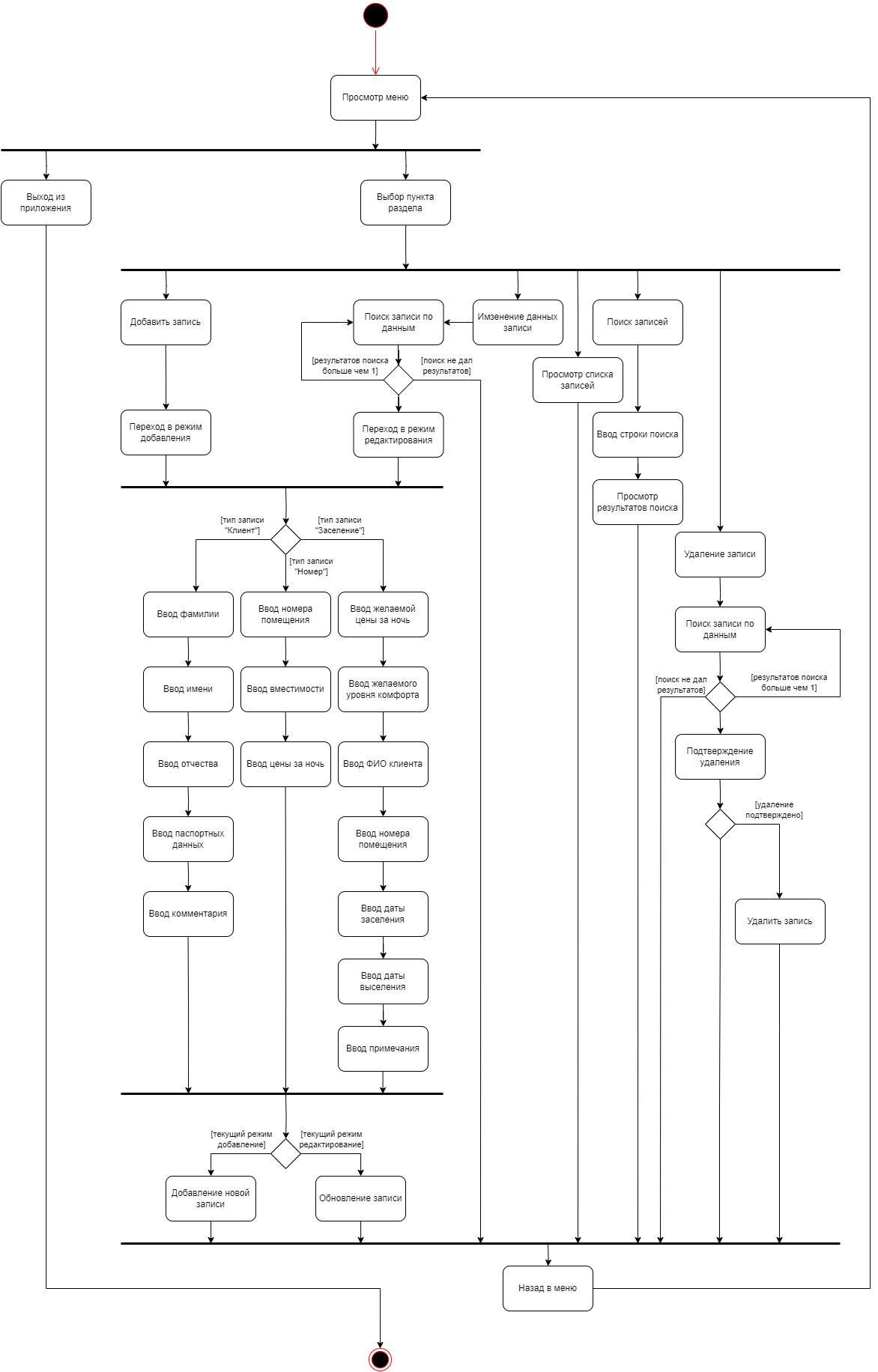
Остановка (End): Конечная точка процесса, обозначающая его завершение. Представляется окружностью с надписью "End".

Решения (Decisions): Условия или решения, ведущие к различным ветвям процесса. Представлены ромбами с условиями или названиями выбора.

Соединители (Transitions): Стрелки, обозначающие переход от одного действия к другому или от решения к следующему шагу в процессе.

Диаграммы деятельности могут быть использованы для описания как простых последовательных процессов, так и сложных ветвящихся алгоритмов. Они помогают визуализировать и анализировать процессы, что полезно при проектировании и оптимизации систем и программного обеспечения.

Диаграмма деятельности, разработанная для данного проекта, представлена на рисунке 2.

Рисунок 2 – Диаграмма деятельности

**2.3 Описание тестов**

Тест-кейсы – это документированные шаги или инструкции, которые определяют, как проводить тестирование программного обеспечения для проверки его соответствия определенным требованиям. Они используются для обеспечения того, что каждый аспект программы тестируется и поведение программы соответствует ожиданиям.

Для данного проекта были составлены тест-кейсы, представленные в таблице 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 1 – Тест-кейсы | | | |
| Идентификатор теста | Название теста | Шаги тест-кейса | Ожидаемый результат |
| 1 | 2 | 3 | 4 |
| Т1 | Проверка действия «Выйти из приложения» | 1. Открыть приложение  2. Ввести «17» | Сохранение данных и завершение работы приложения |
| Т2 | Попытка ввода в строковую переменную (типа string) числовую переменную (типа int) | 1. Открыть приложение  2. Ввести «1» | Обработка исключения и повторный ввод |
| Т3 | Попытка ввода в переменную типа DateTime несуществующую дату | 1. Открыть приложение  2. Ввести «11» | Сообщение о неверном формате даты и повторный ввод |
| Т4 | Попытка ввода недоступного значения комфортности номера | 1. Открыть приложение  2. Ввести «6» | Сообщение о недоступном значении |
| Т5 | Попытка ввода занятого номера в гостинице | 1. Открыть приложение  2. Выбрать пункт меню «Поселение» - «Добавить» | Сообщение «Введите номер свободного помещения» |
| Т6 | Проверка наличия ожидания нажатия «Esc» после каждой операции | 1. Открыть приложение  2. Ввести номер операции  3. Завершить операцию  4. Повторить 2-3 действия со всеми операциями | Ожидание нажатия «Esc» |
| Т7 | Возврат в меню после нажатия «Esc» после выполнения операций | 1. Открыть приложение  2. Ввести номер операции  3. Завершить операцию  4. Нажать «Esc»  5. Повторить 2-4 действия со всеми операциями | Возврат в меню |
|  | | | |
| Продолжение таблицы 1 | | | |
| 1 | 2 | 3 | 4 |
| Т8 | Проверка действия «Добавить» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Добавить»  3. Заполнить данные  4. Завершить операцию  5. Повторить 2-4 действия со всеми сущностями | Добавление сущностей в данные |
| Т9 | Проверка действия «Просмотр» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Просмотр»  3. Повторить предыдущее действие со всеми сущностями | Вывод списка сущностей |
| Т10 | Проверка действия «Поиск» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Поиск»  3. Ввести строку поиска  4. Повторить 2-3 действия со всеми сущностями | Вывод результата поиска |
| Т11 | Проверка действия «Удалить» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Удалить»  3. Произвести поиск любой существующей сущности  4. Подтвердить удаление  5. Повторить 2-4 действия со всеми сущностями | Удаление сущности |
| Т12 | Проверка поиска сущностей при действии «Изменить» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Изменить»  3. Повторить 2 действие со всеми сущностями | Результат поиска соответствует введённой строке поиска |
| Т13 | Проверка поиска сущностей при действии «Удалить» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Удалить»  3. Повторить 2 действие со всеми сущностями | Результат поиска соответствует введённой строке поиска |
| Т14 | Проверка вывода текущих данных после добавления, редактирования и\или удаления объектов | 1. Открыть приложение  2. С помощью пунктов меню добавить, изменить и удалить некоторые сущности  3. Повторить 2 действие со всеми сущностями | Вывод объектов без удаленных, с редактированными и добавлением новых |

**3 Построение программы**

Данный программный продукт представляет собой консольное приложение на языке программирования C#, созданное для управления данными о клиентах, номерах и бронировании номеров. Оно позволяет пользователю выполнять различные операции, такие как добавление, вывод на экран, удаление, поиск и редактирование записей.

Для реализации программного продукта были определены следующие классы:

Класс «Клиент» (Client):

* «Фамилия» (тип string): Surname;
* «Имя» (тип string): Name;
* «Отчество» (тип string): Patronymic;
* «Паспортные данные» (тип string): PassportDetails;
* «Комментарий» (тип string): Comment.

Класс «Номер» (HotelRoom):

* «Номер» (тип int): Number;
* «Количество человек» (тип int): PersonsNumber;
* «Комфортность» (тип string, люкс, полулюкс или обычный): ComfortLevel;
* «Цена» (тип decimal): Price.

Класс «Поселение» (Accommodation):

* «Клиент» (тип string): ClientInitials;
* «Номер» (тип string): HotelRoomNumber;
* «Дата поселения» (тип DateTime): CheckInDate;
* «Дата освобождения» (тип DateTime): CheckOutDate;
* «Примечание» (тип string): Note.

Каждый класс имеет следующие статические методы для работы с массивами объектов, реализующие логику взаимодействия с записями каждого класса:

* «Add»: добавление элемента;
* «Edit»: редактирование элемента;
* «View»: просмотр списка элементов;
* «Search»: поиск элементов;
* «Delete»: удаление элемента.

Дополнительный метод «CalculateRevenue» реализует подсчёт дохода от раннее забронированных номеров.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4 Тестирование**  В рамках проекта было проведено функциональное тестирование данной программы и смоделированы все возможные действия пользователя при работе с приложением.  Отчёт о результатах тестирования продемонстрирован в таблице 2.  Таблица 2 – Отчёт о результатах тестирования | | | | |
| Тест | Шаги теста | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | 2 | 3 | 4 | 5 |
| Проверка действия «Выйти из приложения» | 1. Открыть приложение  2. Ввести «17» | Сохранение данных и завершение работы приложения | Сохранение данных и завершение работы приложения | Выполнено |
| Попытка ввода в строковую переменную (типа string) числовую переменную (типа int) | 1. Открыть приложение  2. Ввести «1» | Обработка исключения и повторный ввод | Обработка исключения и повторный ввод | Выполнено |
| Попытка ввода в переменную типа DateTime несуществующую дату | 1. Открыть приложение  2. Ввести «11» | Сообщение о неверном формате даты и повторный ввод | Сообщение о неверном формате даты и повторный ввод | Выполнено |
| Попытка ввода недоступного значения комфортности номера | 1. Открыть приложение  2. Ввести «6» | Сообщение о недоступном значении | Сообщение о недоступном значении | Выполнено |
| Попытка ввода занятого номера в гостинице | 1. Открыть приложение  2. Выбрать пункт меню «Поселение» - «Добавить» | Сообщение «Введите номер свободного помещения» | Сообщение «Введите номер свободного помещения» | Выполнено |
| Проверка наличия ожидания нажатия «Esc» после каждой операции | 1. Открыть приложение  2. Ввести номер операции  3. Завершить операцию  4. Повторить 2-3 действия со всеми операциями | Ожидание нажатия «Esc» | Ожидание нажатия «Esc» | Выполнено |
|  | | | | |
| Продолжение таблицы 2 | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Возврат в меню после нажатия «Esc» после выполнения операций | 1. Открыть приложение  2. Ввести номер операции  3. Завершить операцию  4. Нажать «Esc»  5. Повторить 2-4 действия со всеми операциями | Возврат в меню | Возврат в меню | Выполнено |
| Проверка действия «Добавить» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Добавить»  3. Заполнить данные  4. Завершить операцию  5. Повторить 2-4 действия со всеми сущностями | Добавление сущностей в базу данных | Добавление сущностей в базу данных | Выполнено |
| Проверка действия «Просмотр» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Просмотр»  3. Повторить предыдущее действие со всеми сущностями | Вывод списка сущностей | Вывод списка сущностей | Выполнено |
| Проверка действия «Поиск» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Поиск»  3. Ввести строку поиска  4. Повторить 2-3 действия со всеми сущностями | Вывод результата поиска | Вывод результата поиска | Выполнено |
| Проверка действия «Удалить» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Удалить»  3. Произвести поиск любой существующей сущности  4. Подтвердить удаление  5. Повторить 2-4 действия со всеми сущностями | Удаление сущности | Удаление сущности | Выполнено |
| Проверка поиска сущностей при действии «Изменить» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Изменить»  3. Повторить 2 действие со всеми сущностями | Результат поиска соответствует введённой строке поиска | Результат поиска соответствует введённой строке поиска | Выполнено |
| Продолжение таблицы 2 | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Проверка поиска сущностей при действии «Удалить» | 1. Открыть приложение  2. В группе операций сущности выбрать пункт «Удалить»  3. Повторить 2 действие со всеми сущностями | Результат поиска соответствует введённой строке поиска | Результат поиска соответствует введённой строке поиска | Выполнено |
| Проверка вывода текущих данных после добавления, редактирования и\или удаления объектов | 1. Открыть приложение  2. С помощью пунктов меню добавить, изменить и удалить некоторые сущности  3. Повторить 2 действие со всеми сущностями | Вывод объектов без удаленных, с редактированными и добавлением новых | Вывод объектов без удаленных, с редактированными и добавлением новых | Выполнено |

В ходе разработки данного программного продукта все обнаруженные проблемы и недочеты устранялись на этапе внедрения проекта. По завершению написания программы проводилось тщательное функциональное тестирование, направленное на обеспечение исправной работы всех компонентов программы в изолированном режиме.

**5** **Использование**

**5.1 Назначение и условия применения программы**

Разработанная программа является консольным приложением «Гостиницы».

Данная программа предназначена для работников гостиницы.

Операционная система: Windows 11 Pro, версия 23H2 или более поздней версии.

Процессор: с тактовой частотой не ниже 1,6 ГГц.

Оперативная память (ОЗУ): требуется 1 ГБ ОЗУ (1,5 ГБ при запуске на виртуальной машине).

Жесткий диск: требуется 200 МБ доступного пространства.

Разрешение экрана: не ниже 1024 на 768.

Среда разработки: Visual Studio 2022 версии 17.0 или более поздней с необходимыми рабочими нагрузками и компонентами.

**5.2 Выполнение программы**

При запуске приложения открывается главное меню (рисунок 3).

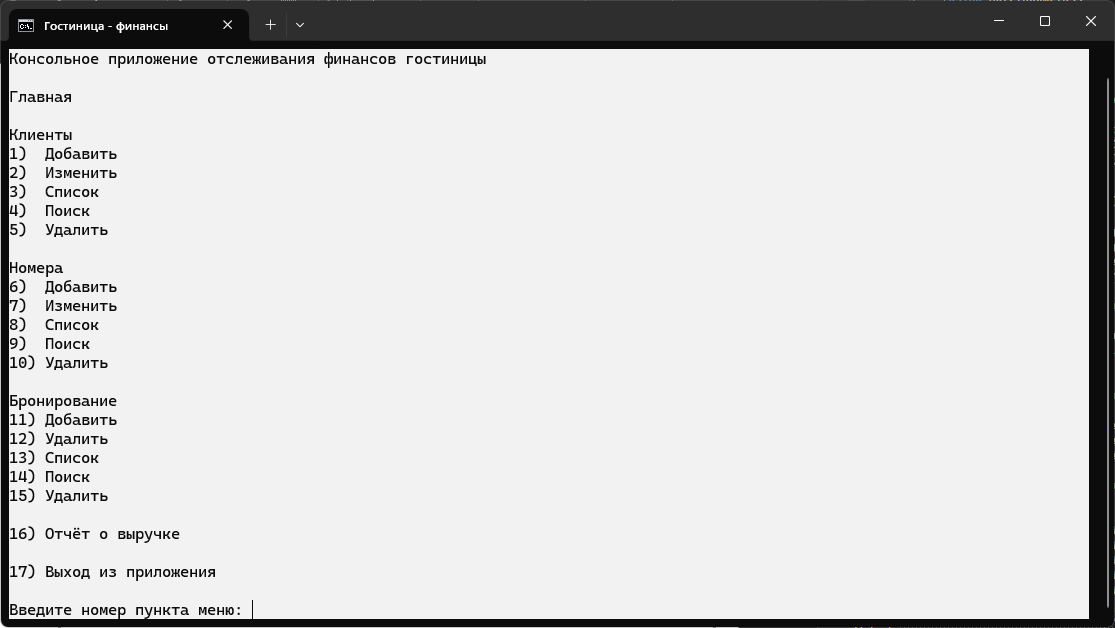
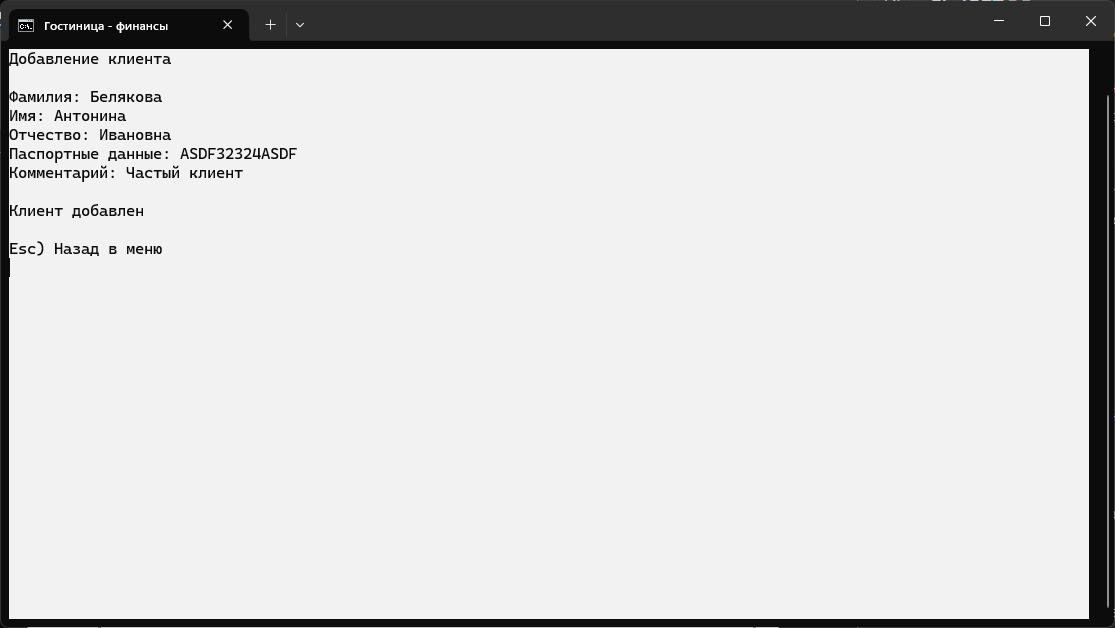


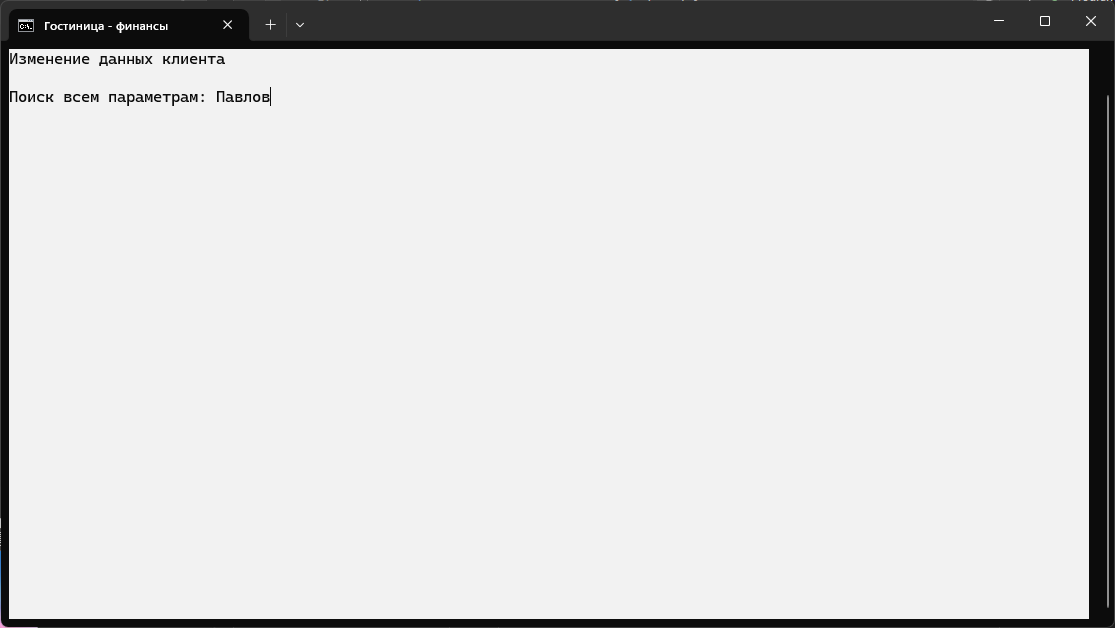
Рисунок 3 - Главное меню

При вводе «1» появляется форма заполнения данных о новом клиенте (рисунок 4).

Рисунок 4 – Список взятых книг

Рисунок 4 – Добавление клиента

По нажатию кнопки «Esc» на клавиатуре, происходит возврат в меню. При вводе «2» из главного меню появляется экран поиска клиента (рисунок 5).

Рисунок 5 – Поиск клиента

При вводе «Павлов» отображается список клиентов, ранее добавленных в хранилище и имеющих в своих данных строку введённую строку (рисунок 6).

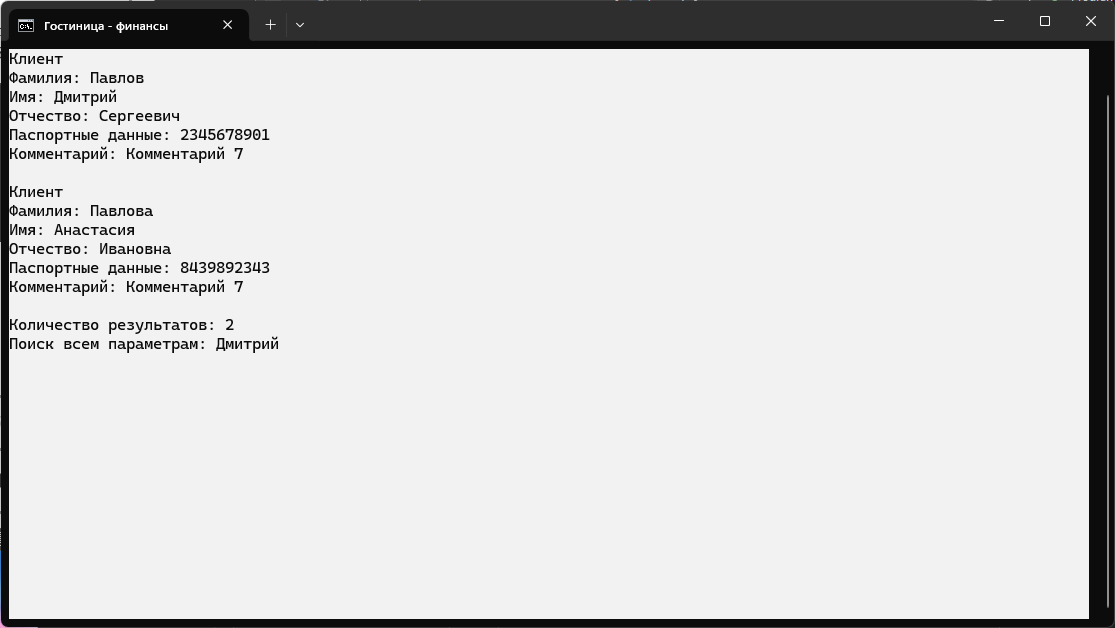


Рисунок 6 – Результат поиска

При вводе «Дмитрий» результат уточнится и останется единственная запись и появится форма изменения данных пользователя. По завершению ввода новых данных появится сообщение о возможности возврата в меню (рисунок 7).

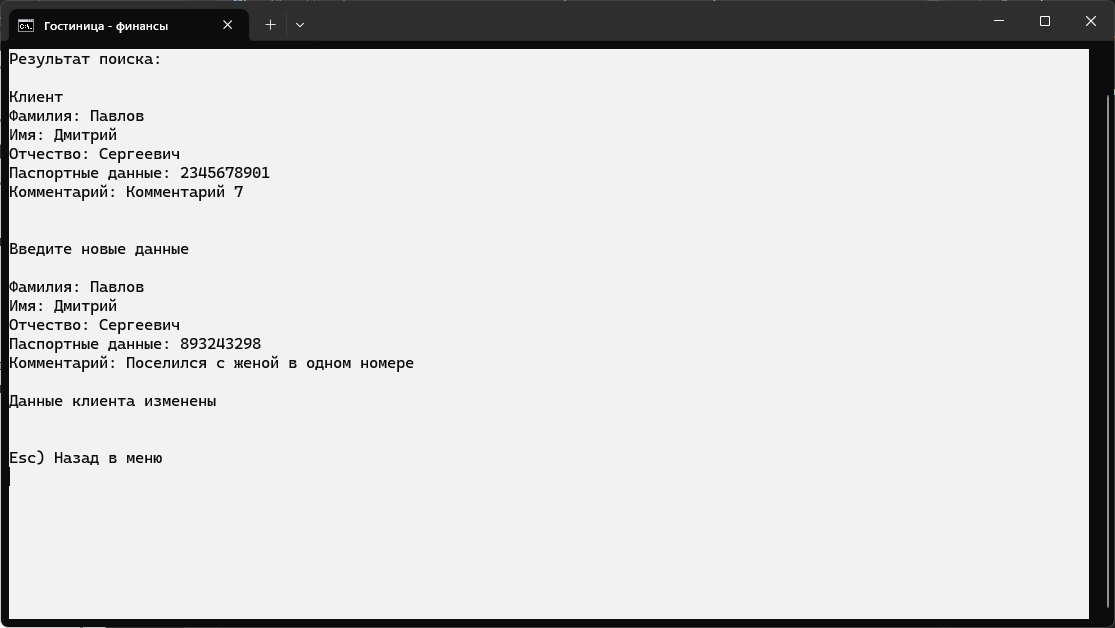
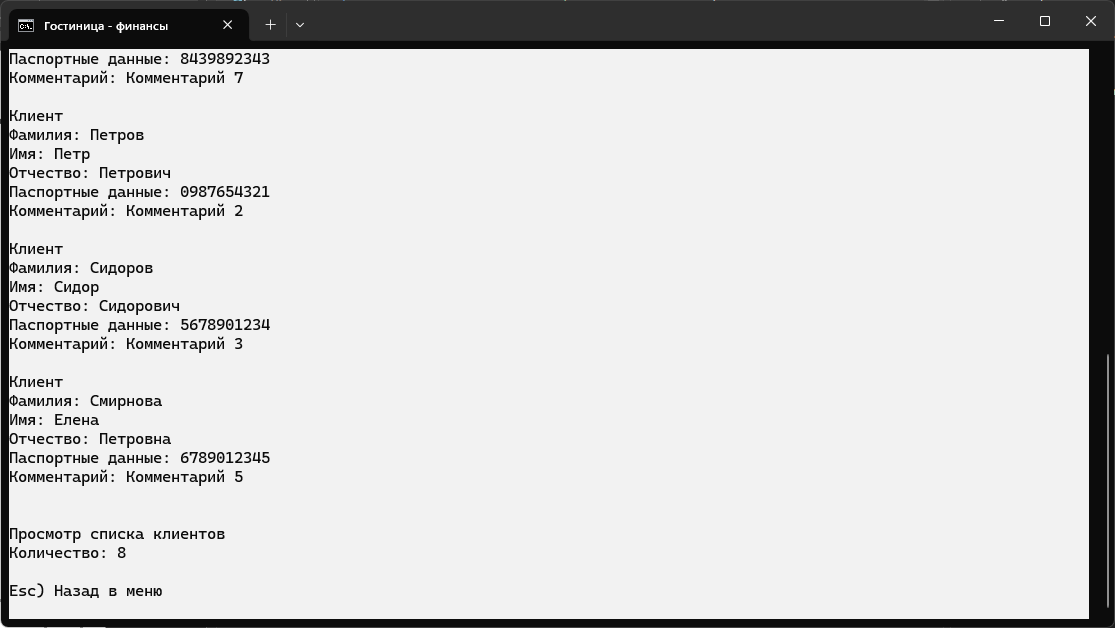


Рисунок 7 – Форма изменения данных клиента

При вводе «3» появляется отсортированный список клиентов и их количество (рисунок 8).

Рисунок 8 – Список клиентов

При выборе пункта меню «Клиенты» – «Поиск» появляется ввод строки поиска (рисунок 9).

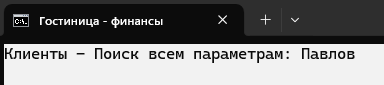
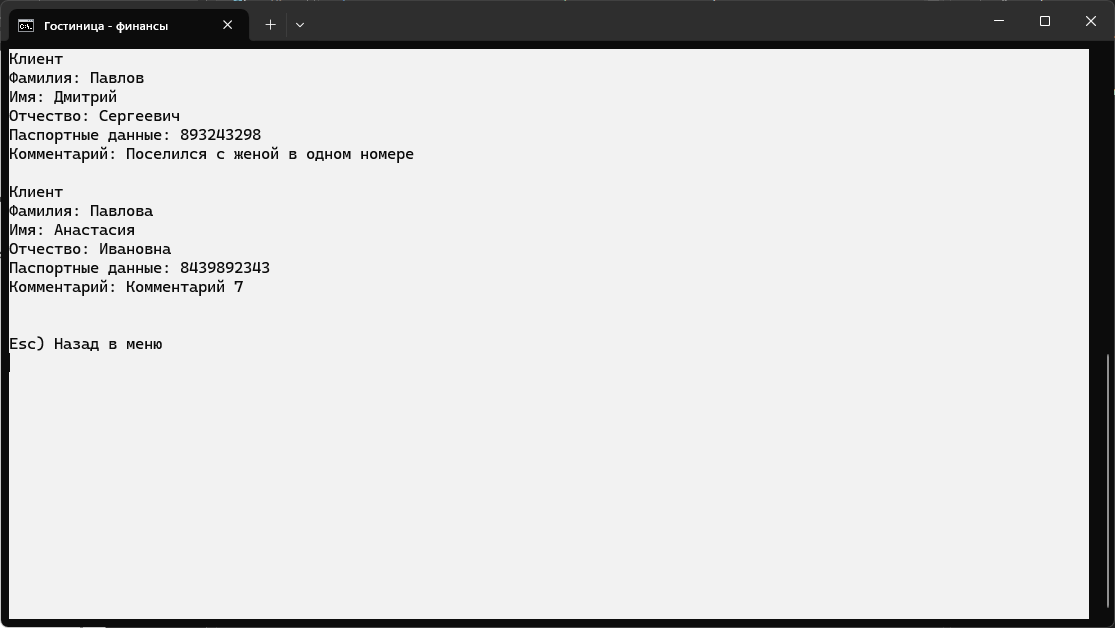


Рисунок 9 – Ввод строки поиска

После ввода отображается результат поиска (рисунок 10).

Рисунок 10 – Результат поиска клиентов

При выборе пункта меню «Клиенты» – «Удалить», отображается такой же поиск, аналогичный поиску при редактировании. По завершению поиска и вводу «1» пользователь подтверждает удаление клиента (рисунок 11).



Рисунок 11 – Удаление клиента завершено

Взаимодействие с разделами «Номера» (6-10 пункты меню) и «Поселение» (11-15) аналогично разделу «Клиента» (1-5) (рисунки 12-23).

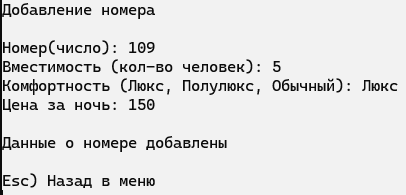


Рисунок 12 – Добавление номера

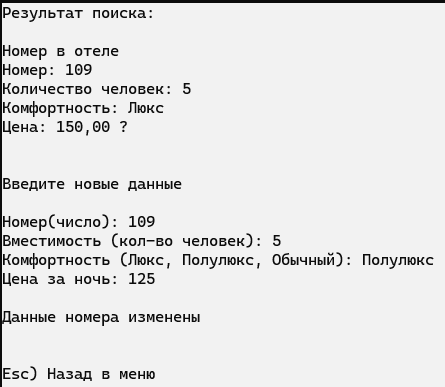


Рисунок 13 – Изменение номера

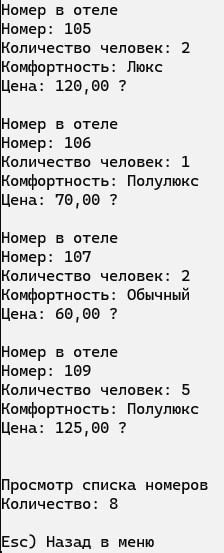


Рисунок 14 – Просмотр списка номеров

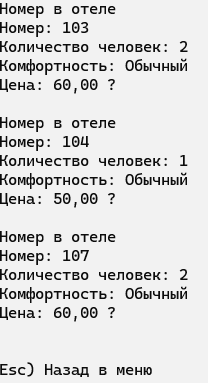
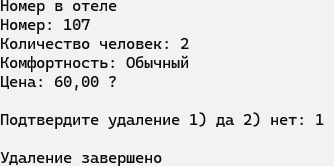
Рисунок 15 – Поиск по комфортности «Обычный»

Рисунок 16 – Поиск и удаление

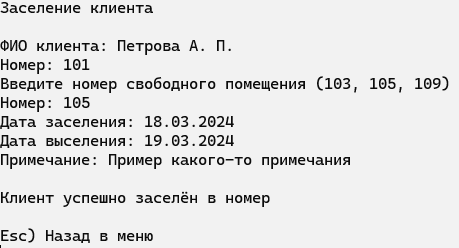
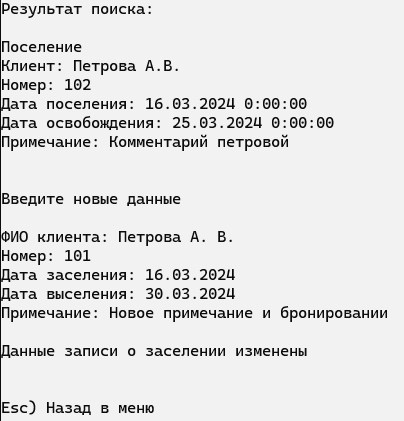


Рисунок 17 – Добавления поселения

Рисунок 18 – Изменение данных о заселение

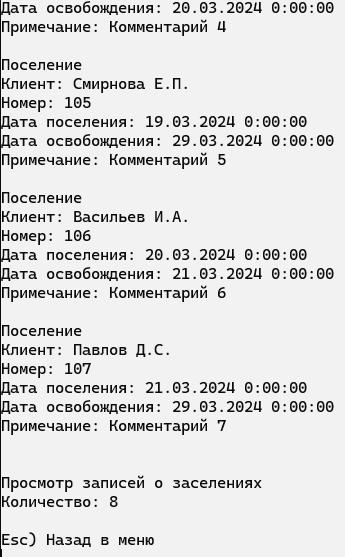
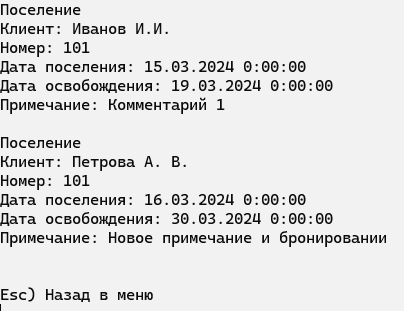


Рисунок 19 – Просмотр списка поселений

Рисунок 20 – Поиск записей поселений

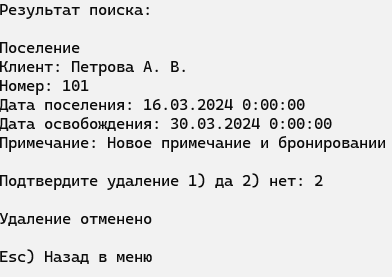


Рисунок 21 – Поиск по строке «16.03» и отмена удаления

При выборе пункта «Отчёт о выручке» появляется строка ввода текущей даты, после чего отображается количество денежных единиц, полученных после освобождения номеров до введённой даты (рисунок 22).

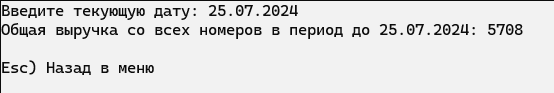


Рисунок 22 – Отчёт о выручке

При выборе пункта «Выход из приложения» происходит сохранение данных и завершение работы приложения (рисунок 23).

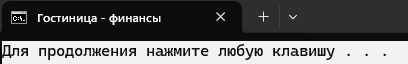


Рисунок 23 – Завершение работы приложения

**Заключение**

Во время практики было разработано консольное приложение «Гостиница» для отслеживания финансовой стороны работы гостиницы. Проект позволяет вести учет клиентов, комнат и бронирований, а также предоставляет функционал поиска свободных комнат и создания новых бронирований.

За время работы над проектом были изучены основы разработки консольных приложений на языке программирования C# и применены важнейшие принципы объектно-ориентированного программирования. Приложение было успешно протестировано, все компоненты проекта проверены, и исключительные ситуации обработаны.

Результаты разработки соответствуют поставленным задачам, пользовательский интерфейс реализован интуитивно понятным и удобным. Можно сделать вывод, что проект выполнен успешно и готов к использованию в реальной работе.

**Список использованных источников**

1. Мюллер, Д. Книга: C# для чайников (2019) / Д. Мюллер, Б. Семпф, Ч. Сфер. – Москва: 2019. — 608 с.
2. Документация по языку программирования C# [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp/ – Дата доступа: 12.02.2024.
3. Руководство по C# [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/tutorial/1.1.php – Дата доступа: 13.02.2024.
4. Head First. Изучаем C#(2022) [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/top-10-knig-po-c-v-2023> ¬ Дата доступа: 03.03.2024.