

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА  
ПОЯСНЮВАЛЬНА ЗАПИСКА  
з дисципліні “ Об’єктно -орієнтоване програмування”

Керівник , Професор кафедри програмної інженерії      Бондарєв В.М.  
Студент гр. ПЗПІ-22-1      Ханіна Д.Д.

Комісія:

Проф. Бондарев В.М.,  
Ст. викл. Черепанова Ю.Ю.,  
Ст. викл. Ляпота В.М.

Харків 2023

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
РАДІОЕЛЕКТРОНІКИ

Кафедра *програмної інженерії*

Рівень вищої освіти *перший (бакалаврський)*

Дисципліна *Об'єктно-орієнтоване програмування*

Спеціальність *121 Інженерія програмного  
забезпечення*

Освітня програма: *Програмна інженерія*

Курс 1.

Група ПЗПІ-22 -1.

Семестр 2

**ЗАВДАННЯ**

***на курсовий проект студента***

*Ханіна Дарина Денисівна*

- 1 Тема проекту: Адміністратор Готеля
- 2 Термін здачі студентом закінченого проекту: ***“16” - червня - 2023 р.***
- 3 Вихідні дані до проекту:  
*Специфікація програми, методичні вказівки до виконання курсової роботи*
- 4 Зміст розрахунково-пояснювальної записки:  
*Вступ.*  
*опис вимог, проектування програми, інструкція користувача, висновки*

## КАЛЕНДАРНИЙ ПЛАН

	<i>Назва етапу</i>	<i>Термін виконання</i>
	Видача теми, узгодження і затвердження теми	13.02.2023 - 14.03.2023 р.
	Формулювання вимог до програми	29.03.2023 –30.03.2023 р.
	Розробка підсистеми зберігання та пошуку даних.	03.04.2023 –10.04.2023 р.
	Розробка функцій .....	11.04.2023 –20.04.2023 р.
	Розробка функцій зберігання та завантаження даних	21.04.2023 –01.05.2023 р.
	Тестування і доопрацювання розробленої програмної системи.	2.05.2023 – 14.05.2023 р.
	Оформлення пояснювальної записки, додатків, графічного матеріалу	20.05.2023 – 09.06.2023 р.
	Захист	05.06.2023 – 16.06.2023 р.

Студент

Ханіна Дарина Денисівна

Керівник

Бондарев Володимир Михайлович

« 13 » лютого 2023 р.

## РЕФЕРАТ

Пояснювальна записка до курсової роботи: 30 с., 22 рис., 3 табл., 2 джерела.

КЛАС, КОЛЕКЦІЯ, МЕТОД, МОВА ПРОГРАМУВАННЯ C#, ГОТЕЛЬ, НОМЕРИ, АДМІНІСТРАТОР, ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПРОГРАМА.

Метою роботи є розробка програми «Адміністратор готелю», яка буде надавати адміністратору готеля функції для роботи з гостями та номерами.

В результаті отримана програма, що дозволяє зберігати список гостей, характеристики кожного гостя, такі як: ім'я, прізвище, номер паспорту, номер готеля, в'їзд, виїзд. Є можливість утворювати нові списки гостей, додавати, видаляти та отримувати списки гостей. Також, є функція формування чеку.

В процесі розробки використано середовища Rider, платформи .NET 6.0, мова програмування C#.

## ЗМІСТ

<b>ВСТУП</b>	<b>6</b>
<b>1 СПЕЦИФІКАЦІЯ ПРОГРАМИ</b>	<b>7</b>
1.1 Основні функції програми	7
1.2 Опис функцій програми	7
1.2.1 Функції для роботи зі сторінкою «Дія над гостями»	8
1.2.1.1 Функція «Додавання гостя»	8
1.2.1.2 Функція «Пошук гостей за ознакою»	9
1.2.1.2.1 Функція «Список усіх гостей»	9
1.2.1.2.2 Функція «Пошук за номером паспорту»	10
1.2.1.2.3 Функція «Пошук за датою виїзду»	10
1.2.1.3 Функція «Виселення гостя»	11
1.2.2 Функції для роботи зі сторінкою «Дія над номерами»	11
1.2.2.1 Функція «Додавання кімнати»	12
1.2.2.2 Функція «Список усіх кімнат»	12
<b>2 ПРОЕКТУВАННЯ ПРОГРАМИ</b>	<b>13</b>
2.1 Об'єктна структура програми	13
2.1 Докладний опис класів та їх взаємодія	14
2.1.1 Клас BaseEntity	15
2.1.2 Клас HotelRoom	15
2.1.3 Клас Guest	16
2.1.4 Інтерфейс IRepository	17
2.1.5 Інтерфейс IGenericService	18
2.1.6 Інтерфейс IGuestService	19
2.1.7 Інтерфейс IHotelRoomService	20
2.1.8 Класи DependencyRegistration	20
2.1 Зберігання та використання даних	21
<b>3 ІНСТРУКЦІЯ КОРИСТУВАЧА</b>	<b>24</b>
3.1 Встановлення програми	24
3.2 Функціонал програми та інструкція до неї	24
3.2.1 Робота зі сторінкою «Дія над гостями»	25
3.2.1.1 Робота зі формою «Додати гостя»	25
3.2.1.2 Робота зі формою «Пошук гостей за ознакою»	26
3.2.1.3 Робота зі формою «Виселення гостя»	27
3.2.2 Робота зі сторінкою «Дія над номерами»	28

	5
3.2.2.1 Робота зі формою «Додати кімнату у базу»	28
3.2.2.2 Меню з усіма номерами	28
<b>ВИСНОВОК</b>	<b>29</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ</b>	<b>30</b>

## ВСТУП

Метою моєї курсової роботи є створення консольного застосунку адміністратор готеля. Для цього були реалізовані наступні задачі:

- Додавання гостей у базу;
- Пошук усіх гостей;
- Пошук гостей за номером паспорту;
- Пошук гостей за датою виїзду;
- Виселення гостей;
- Додавання кімнат у базу;
- Пошук усіх номерів з повною інформацією про них;

В процесі розробки використано середовища Rider, платформи .NET 6.0, мова програмування C#

## 1 СПЕЦИФІКАЦІЯ ПРОГРАМИ

### 1.1 Основні функції програми

Завдяки програмі «Адміністратор Готеля», адміністратор готеля має змогу додавати гостей в обраний номер із п'яті, вже доданих у програму на старті, номерів. Також він може виселяти їх і видавати чек за кількість днів, які гість повинен був прожити у своєму номері. Крім того, адміністратор може переглядати список усіх гостей та список усіх номерів, з повною інформацією про них.

Для здійснення програми, вона повинна мати в собі наступні функції:

- Додавання гостей у базу;
- Пошук усіх гостей;
- Пошук гостей за номером паспорту;
- Пошук гостей за датою виїзду;
- Виселення гостей;
- Додавання кімнат у базу;
- Пошук усіх номерів з повною інформацією про них;

### 1.2 Опис функцій програми

При запуску програми у адміністратора з'являється головне меню з трьома варіантами вибору дій, на які можна перейти («Дія над гостями (1)», «Дія над номерами (2)») та кнопка виходу («Вийти з програми»)



### 1.2.1 Функції для роботи зі сторінкою «Дія над гостями»

Коли адміністратор потрапляє на сторінку «Дії над гостями», йому доступні декілька функцій для роботи з гостями.

Адміністратор має можливість:

- Додати гостя (1. Додати гостя)
- Провести пошук гостя за ознакою (2. Пошук гостей за ознакою)
- Виселити гостя (3. Выселення гостя)
- Повернутися до головного меню (4. Вийти)

#### 1.2.1.1 Функція «Додавання гостя»

Ця функція доступна адміністратору після натискання кнопки додавання. Відкривається форма, у яку потрібно ввести дані гостя (Ім'я, Прізвище, Номер Паспорту, Дату Від'їзду (День, Місяць, Рік)) та обрати номер, в якій відбувається заселення (попередньо заданий). Якщо номер паспорту, введений адміністратором вже є у базі, з'являється повідомлення, про те, що ця людина вже є у базі.

Від адміністратора потрібно лише натиснути кнопку додати та увести дані правильно, якщо якісь дані введені некоректно програма повідомляє про це адміністратору.

Після успішного додавання гостя, адміністратор має можливість повернутися на сторінку «Дії над гостями».

### 1.2.1.2 Функція «Пошук гостей за ознакою»

Ця функція доступна адміністратору після натискання кнопки «Пошук Гостей».

Адміністратор має можливість:

- Показати список усіх гостей (1. Список усіх гостей)
- Провести пошук гостя за номером паспорту (2. Пошук гостя за номером паспорту)
- Провести пошук гостя за датою виїзду (3. Пошук гостей за датою виїзду)
- Повернутися на сторінку «Дії над гостями» (4. Вийти)

#### 1.2.1.2.1 Функція «Список усіх гостей»

Ця функція доступна адміністратору після натискання кнопки «Список усіх гостей».

Адміністратор має можливість побачити повний перелік усіх даних про гостей, введених раніше (Ім'я, Прізвище, Номер Паспорту, Дату В'їзду та Виїзду (День, Місяць, Рік, Номер гостя))

Після появи списку гостей, адміністратор має можливість повернутися на сторінку «Дії над гостями», або вийти до головного меню

#### 1.2.1.2.2 Функція «Пошук за номером паспорту»

Ця функція доступна адміністратору після натискання кнопки «Пошук за номером паспорту». Відкривається форма, у яку потрібно ввести номер паспорту гостя.

Якщо адміністратор вводить правильний номер паспорту, з'явиться інформація про цього гостя ((Ім'я, Прізвище, Номер Паспорту, Дату В'їзду та Виїзду (День, Місяць, Рік, Номер гостя)).

Якщо введеного номеру паспорта немає у базі, про це з'явиться повідомлення

Після появи інформації про гостя, адміністратор має можливість повернутися на сторінку «Дії над гостями», або вийти до головного меню.

#### 1.2.1.2.3 Функція «Пошук за датою виїзду»

Ця функція доступна адміністратору після натискання кнопки «Пошук за датою виїзду». Відкривається форма, у яку потрібно ввести дату (День, Місяць, Рік).

Якщо адміністратор вводить існуючу у базі дату виїзду, з'явиться інформація про цього гостя ((Ім'я, Прізвище, Номер Паспорту, Дату В'їзду та Виїзду (День, Місяць, Рік, Номер гостя)).

Якщо введеної дати немає у базі, про це з'явиться повідомлення.

Після появи інформації про гостя, адміністратор має можливість повернутися на сторінку «Дії над гостями», або вийти до головного меню.

### 1.2.1.3 Функція «Виселення гостя»

Ця функція доступна адміністратору після натискання кнопки «Виселення гостя». Відкривається форма, у якій є перелік усіх зареєстрованих гостей. З цієї форми потрібно обрати гостя, якого адміністратор бажає виселити.

Після появи інформації про те, що гость виселений, з'явиться інформація про його чек (сумму яку гість, повинен заплатити за перебування в готелі). Адміністратор має можливість повернутися на сторінку «Дії над гостями», або вийти до головного меню.

При виселенні гостя, який зареєстрований у номері з іншими гостями, усі гості з цього номеру будуть автоматично виселені.

### 1.2.2 Функції для роботи зі сторінкою «Дія над номерами»

Коли адміністратор потрапляє на сторінку «Дії над номерами», йому доступні декілька функцій для роботи з номерами.

Адміністратор має можливість:

- Додати кімнату (1. Додати кімнату у базу );
- Показати список усіх номерів та гостей, зареєстрованих у них (2. Пошук усіх номерів );
- Повернутися до головного меню (3. Вийти );

Якщо кімната, яку адміністратор бажає додати вже є в базі, про це з'явиться повідомлення.

#### 1.2.2.1 Функція «Додавання кімнати»

Ця функція доступна адміністратору після натискання кнопки «Додати кімнату у базу». Відкривається форма, у якій є перелик усіх доступних кімнат, які можна додати. З цієї форми потрібно обрати кімнату, яку адміністратор бажає додати. Від самого початку в програму вже додані усі номери.

Після завершення функції, адміністратор має можливість повернутися на сторінку «Дії над номерами», або вийти до головного меню

#### 1.2.2.2 Функція «Список усіх кімнат»

Ця функція доступна адміністратору після натискання кнопки «Пошук усіх номерів». Відкривається форма, у якій інформація про всі кімнати у готелі (Назва, Кількість усіх місць, Кількість вільних місць, Ціна) та інформація про усіх гостей у них ((Ім'я, Прізвище, Номер Паспорту, Дату В'їзду та Виїзду (День, Місяць, Рік,)).

Після завершення функції, адміністратор має можливість повернутися на сторінку «Дії над номерами», або вийти до головного меню

## 2 ПРОЕКТУВАННЯ ПРОГРАМИ

### 2.1 Об'єктна структура програми

Програма була написана з дотриманням правил триступеневої архітектури. Який включає в себе поділ даних програми, на три основні компоненти: UI (User Interface), BLL (Business Logic Layer), DAL (Data Access Layer). Рівень доступу до даних залежить від інших рівнів, рівень бізнес-логіки залежить від рівня доступу до даних, а рівень уявлення - від рівня бізнес-логіки.

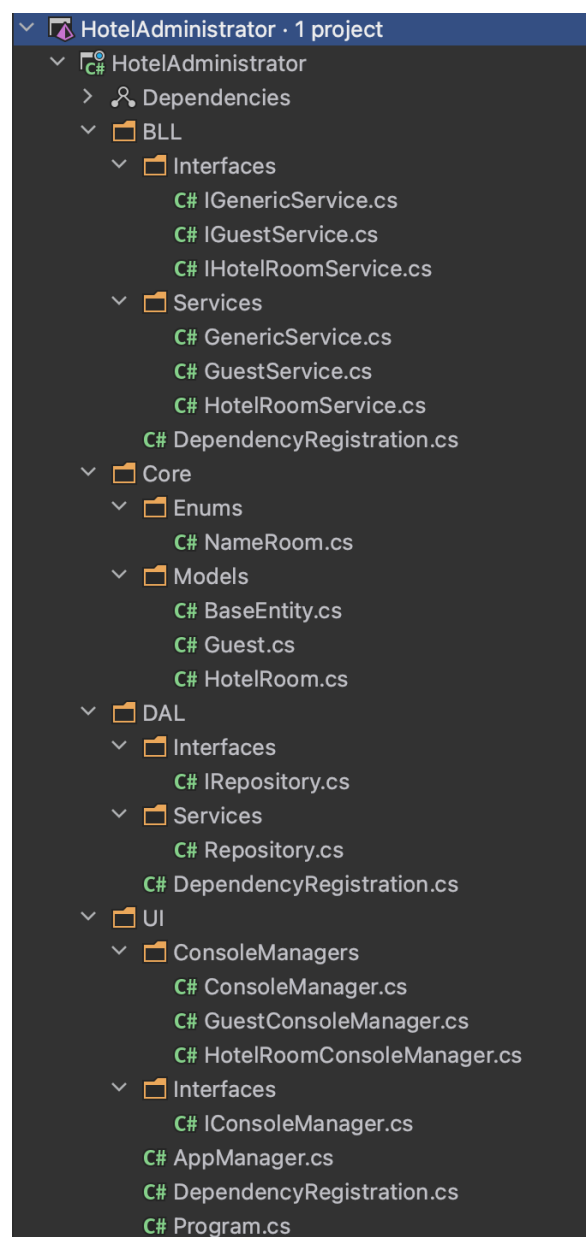


Рис. 2.1 – Структура проекту

Клас BaseEntity, який позначає кожній моделі айді та є абстрактним, та класи Guest, HotelRoom, створюються задля того, щоб використовуватися у всьому проекті. NameRoom є набір постійних назв для HotelRoom. Це все знаходиться в Core і може бути використано в будь якій частині програми.

В DAL храниться робота з файлами типу json завдяки інтерфейсу та сервісу IRepository і Repository.

В BLL знаходиться основна логіка програми і такі сервіси, як IGenericService, який впроваджує методи для роботи з моделями в файлах, IGuestService і IHotelRoomService, які містять методи для роботи із їх сутностями. Та класи GenericService, GuestService, HotelRoomService в який є реалізація.

В UI бачимо роботу з консоллю, перевірку деяких даних та взагалі все, що бачить адміністратор завдяки AppManager та ConsoleManagers.

Клас DependencyRegistration включає в себе процес вказання, які класи або компоненти залежать від інших класів або компонентів, а також як їх створити та внедрити.

## 2.1 Докладний опис класів та їх взаємодія

Програма написана за допомогою 3 класів в Core:

- BaseEntity;
- HotelRoom;
- Guest;

1 інтерфейс в DAL:

- IRepository

3 інтерфейси в BLL:

- IGenericService

- IHotelRoomService
- IGuestService

Та класи з реєстрацією залежностей та класи з реалізацією.

### 2.1.1 Клас BaseEntity

Цей клас представляє собою 1 айді для наступних моделей та є абстрактним.

```
public abstract class BaseEntity
{
    10 usages
    public Guid Id { get; set; }
}
```

Рис. 2.2 – Клас BaseEntity

Він складається з:

- Відкритого свойства Id;

Цей клас потрібен для надання моделям айдішника, для більш легкої роботи над ними.

### 2.1.2 Клас HotelRoom

Цей клас представляє собою 1 номер та його характеристики.



```

public class HotelRoom : BaseEntity
{
    10 usages
    public NameRoom NameRoom { get; set; }
    6 usages
    public List<Guest> Guests { get; set; }
    2 usages
    public int Seats { get; set; }
    6 usages
    public int AvailableSeats { get; set; }
    4 usages
    public decimal PriceForOneDay { get; set; }
}

```

Рис. 2.3 – Клас HotelRoom

Він складається з:

- Відкритого свойства Id;
- Відкритого свойства Guests
- Відкритого свойства Seats
- Відкритого свойства AvailableSeats
- Відкритого свойства PriceForOneDay

Цей клас потрібен для більш легкої реалізації списку номерів.

### 2.1.3 Клас Guest

Цей клас представляє собою 1 гостя та його характеристики.

```

public class Guest : BaseEntity
{
    5 usages
    public string Name { get; set; }
    4 usages
    public string SurName { get; set; }
    6 usages
    public string NumPassport { get; set; }
    6 usages
    public DateTime CheckIn { get; set; }
    7 usages
    public DateTime CheckOut { get; set; }
    7 usages
    public HotelRoom HotelRoom { get; set; }
}

```

Рис. 2.4 – Клас Guest

Він складається з:

- Відкритого свойства Name;
- Відкритого свойства SurName
- Відкритого свойства NumPassport
- Відкритого свойства CheckIn
- Відкритого свойства CheckOut
- Відкритого свойства HotelRoom

Цей клас потрібен для більш легкої реалізації списку гостей та їх пошуку.

#### 2.1.4 Інтерфейс IRepository

```
public interface IRepository<T> where T : BaseEntity
{
    [1 usage] [1 implementation] darhan1168
    List<T> GetAll(int pageNumber = 1, int pageSize = 10);

    [1 usage] [1 implementation] darhan1168
    T GetById(Guid id);

    [1 usage] [1 implementation] darhan1168
    T GetByPredicate(Func<T, bool> predicate);

    [1 usage] [1 implementation] darhan1168
    void Add(T obj);

    [1 usage] [1 implementation] darhan1168
    void Update(Guid id, T updateObj);

    [1 usage] [1 implementation] darhan1168
    void Delete(Guid id);
}
```

Рис. 2.5 – Інтерфейс IRepository в DAL

Він складається з:

- Нереалізованого метода GetAll;
- Нереалізованого метода GetById;
- Нереалізованого метода GetByPredicate;
- Нереалізованого метода Add;

- Нереалізованого метода Update;
- Нереалізованого метода Delete;

Цей інтерфейс потрібен для реалізації методів та їх взаємодії подалі

### 2.1.5 Інтерфейс IGenericService

```
public interface IGenericService<T> where T : BaseEntity
{
    [ 1 usage 1 implementation darhan1168 ]
    void Add(T obj);

    [ 1 usage 1 implementation darhan1168 ]
    void Delete(Guid id);

    [ 1 usage 1 implementation darhan1168 ]
    T GetById(Guid id);

    [ 6 usages 1 implementation darhan1168 ]
    List<T> GetAll();

    [ 1 usage 1 implementation darhan1168 ]
    T GetByPredicate(Func<T, bool> predicate);

    [ 3 usages 1 implementation darhan1168 ]
    void Update(Guid id, T obj);
}
```

Рис. 2.6 – Інтерфейс IGenericService в BLL

Він складається з:

- Нереалізованого метода GetAll;
- Нереалізованого метода GetById;
- Нереалізованого метода GetByPredicate;
- Нереалізованого метода Add;
- Нереалізованого метода Update;
- Нереалізованого метода Delete;

Цей інтерфейс потрібен для реалізації методів та їх взаємодії подалі в BLL

### 2.1.6 Інтерфейс IGuestService

```
public interface IGuestService : IGenericService<Guest>
{
    [1 usage] [1 implementation] [darhan1168]
    void RegisterGuest(Guest guest);

    [1 usage] [1 implementation] [darhan1168]
    void DeleteGuest(Guid guestId);

    [1 implementation] [darhan1168]
    List<Guest> GetAllGuest();

    [1 implementation] [darhan1168]
    List<Guest> GetGuestByHotelRoom(HotelRoom hotelRoom);

    [1 usage] [1 implementation] [darhan1168]
    List<Guest> GetGuestByCheckOut(DateTime checkOut);

    [1 usage] [1 implementation] [darhan1168]
    Guest GetGuestByNumPassport(string numPassport);

    [1 usage] [1 implementation] [darhan1168]
    decimal GetCheck(Guid guestId);
}
```

Рис. 2.7 –Інтерфейс IGuestService в BLL

Він складається з:

- Нереалізованого метода RegisterGuest;
- Нереалізованого метода DeleteGuest;
- Нереалізованого метода GetAllGuest;
- Нереалізованого метода GetGuestByHotelRoom;
- Нереалізованого метода GetGuestByCheckOut;
- Нереалізованого метода GetGusetByNumPassport;
- Нереалізованого метода GetCheck

Цей інтерфейс потрібен для реалізації методів для дій над гостями та їх взаємодії подалі в BLL

### 2.1.7 Інтерфейс IHotelRoomService

```
public interface IHotelRoomService : IGenericService<HotelRoom>
{
    [1 usage] [1 implementation] darhan1168
    void CreateHotelRoom(HotelRoom hotelRoom);

    [1 usage] [1 implementation] darhan1168
    List<HotelRoom> GetAllRooms();
}
```

Рис. 2.8 – Інтерфейс IHotelRoomService в BLL

Він складається з:

- Нереалізованого метода CreateHotelRoom;
- Нереалізованого метода GetAllRooms;

Цей інтерфейс потрібен для реалізації методів для дій над номерами та їх взаємодії подалі в BLL

### 2.1.8 Класи DependencyRegistration

Визначаємо процеси визначення та документування залежностей між різними елементами системи або проекту.

```
public class DependencyRegistration
{
    [1 usage] darhan1168
    public static void RegisterRepositories(IServiceCollection services)
    {
        services.AddScoped(serviceType: typeof(IRepository<>), implementationType: typeof(Repository<>));
    }
}
```

Рис. 2.9 – Клас DependencyRegistration в DAL

```

public class DependencyRegistration
{
    [1 usage] [darhan1168]
    public static void RegisterServices(IServiceCollection services)
    {
        services.AddScoped<IGuestService, GuestService>();
        services.AddScoped<IHotelRoomService, HotelRoomService>();

        DAL.DependencyRegistration.RegisterRepositories(services);
    }
}

```

Рис. 2.10 – Клас DependencyRegistration в BLL

```

public class DependencyRegistration
{
    [1 usage] [darhan1168]
    public static IServiceProvider Register()
    {
        var services = new ServiceCollection();
        services.AddScoped<AppManager>();
        services.AddScoped<GuestConsoleManager>();
        services.AddScoped<HotelRoomConsoleManager>();

        foreach (var type in typeof(IConsoleManager)
            .Where(t:Type => t.IsClass &&
                .Any(i:Type => i.IsGenericType))
        {
            var interfaceType = type.GetInterfaces();
            services.AddScoped(serviceType: interfaceType);
        }

        BLL.DependencyRegistration.RegisterServices(services);

        return services.BuildServiceProvider();
    }
}

```

Рис. 2.11 – Клас DependencyRegistration в UI

## 2.1 Зберігання та використання даних

Дані о колекції номерів зберігаються у 1 файлі HotelRoom.json. Дані о колекції гостей зберігаються у 1 файлі Guest.json. Якщо ці файли відсутні, то програма їх сама створить.

```
[
  {
    "NameRoom": 0,
    "Guests": [],
    "Seats": 1,
    "AvailableSeats": 1,
    "PriceForOneDay": 300.0,
    "Id": "8d8f24c1-cd75-4eba-b30e-cd6aa0fe9417"
  },
  {
    "NameRoom": 1,
    "Guests": [],
    "Seats": 2,
    "AvailableSeats": 2,
    "PriceForOneDay": 500.0,
    "Id": "820e034c-8160-4beb-acdb-5a9c41b7c2cc"
  },
  {
    "NameRoom": 2,
    "Guests": [],
    "Seats": 3,
    "AvailableSeats": 3,
    "PriceForOneDay": 1000.0,
    "Id": "acbbe7ad-0b02-4cb5-9047-b7f50beab383"
  },
  {
    "NameRoom": 3,
    "Guests": [],
    "Seats": 4,
    "AvailableSeats": 4,
    "PriceForOneDay": 2000.0,
    "Id": "a764c252-94ec-4137-aca2-da4de7f75c66"
  },
  {
    "NameRoom": 4,
    "Guests": [],
    "Seats": 5,
    "AvailableSeats": 5,
    "PriceForOneDay": 5000.0,
    "Id": "8eeeba45-769e-49b2-8a42-b22dbac2eb6e"
  }
]
```

Рис. 2.12 Приклад змісту файла HotelRoom.json

```
[
  {
    "Name": "Дарина ",
    "SurName": "Ханіна",
    "NumPassport": "A12H8D",
    "CheckIn": "2023-05-23T00:00:00+02:00",
    "CheckOut": "2023-05-26T00:00:00",
    "HotelRoom": {
      "NameRoom": 0,
      "Guests": [],
      "Seats": 1,
      "AvailableSeats": 1,
      "PriceForOneDay": 300.0,
      "Id": "8d8f24c1-cd75-4eba-b30e-cd6aa0fe9417"
    },
    "Id": "817c0cca-f540-419b-9843-3e0583cc94be"
  },
  {
    "Name": "Олена",
    "SurName": "Зозуля",
    "NumPassport": "G678KS",
    "CheckIn": "2023-05-23T00:00:00+02:00",
    "CheckOut": "2023-05-30T00:00:00",
    "HotelRoom": {
      "NameRoom": 2,
      "Guests": [],
      "Seats": 3,
      "AvailableSeats": 3,
      "PriceForOneDay": 1000.0,
      "Id": "acbbe7ad-0b02-4cb5-9047-b7f50beab383"
    },
    "Id": "16f92964-059d-4d9c-8bc3-754988723233"
  }
]
```

Рис. 2.13 Приклад змісту файла Guest.json



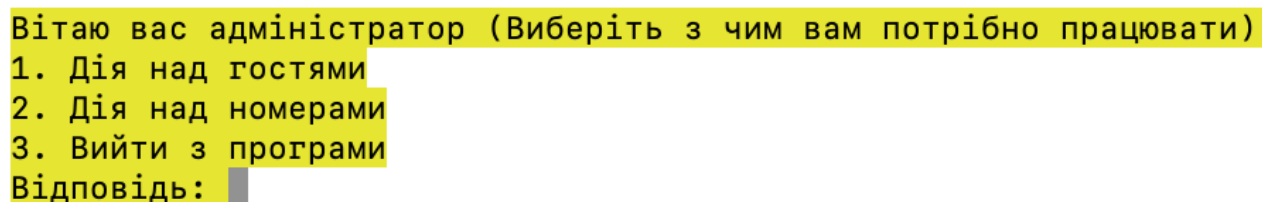
### 3 ІНСТРУКЦІЯ КОРИСТУВАЧА

#### 3.1 Встановлення програми

Для запуску програма потребується знайти консольний застосунок з назвою HotelAdministrator, який має знаходитися у папці `bin > debug > net6.0 >` Консольний застосунок. Усі інші файли та папки програма створить сама. Але ці файли будуть без даних.

#### 3.2 Функціонал програми та інструкція до неї

Як тільки запустився консольний застосунок, можна побачити сторінку із 3 діями: «Дія над гостями», Дія над номерами, «Вийти з програми».



```
Вітаю вас адміністратор (Виберіть з чим вам потрібно працювати)
1. Дія над гостями
2. Дія над номерами
3. Вийти з програми
Відповідь: 
```

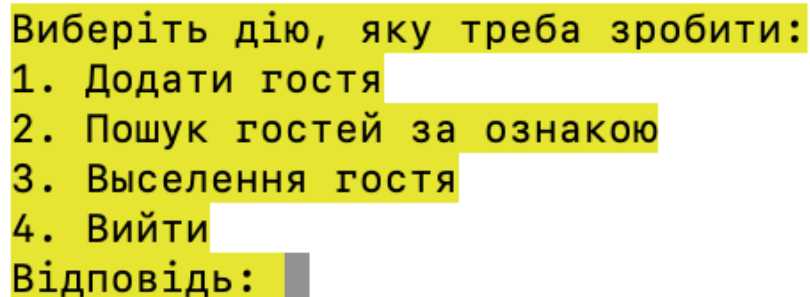
Рис 3.1 – Зображення головного меню

При натисканні кнопки «Вийти з програми» програма закривається, а при натисканні інших кнопок, відкривається відповіді меню

### 3.2.1 Робота зі сторінкою «Дія над гостями»

Коли адміністратор потрапляє на це меню, то перед ним відображається меню з 4 діями: «Додати гостя», «Пошук гостя за ознакою», «Виселення гостя», «Вийти».

Завдяки цифрам 1-4, адміністратор обирає дію.



Виберіть дію, яку треба зробити:

1. Додати гостя
2. Пошук гостей за ознакою
3. Виселення гостя
4. Вийти

Відповідь:

Рис 3.2 – Зображення меню з діями над гостями

#### 3.2.1.1 Робота зі формою «Додати гостя»

Адміністратору треба заповнити всі данні, тогі гость буде додан у базу, якщо адміністратор зробить помилку, то впливає помилка

```

Введіть ім'я
Відповідь: Дарина
Введіть прізвище
Відповідь: Ханіна
Введіть номер паспорту
Відповідь: 123456
Введіть день вид'їзду
Відповідь: 28
Введіть місяць вид'їзду
Відповідь: 5
Введіть рік вид'їзду
Відповідь: 2023
Список доступних номерів:
1 – Назва: SingleRoom, Кол-во вільних місць: 1, Ціна: 300,0
2 – Назва: DoubleRoom, Кол-во вільних місць: 2, Ціна: 500,0
3 – Назва: TwinRoom, Кол-во вільних місць: 3, Ціна: 1000,0
4 – Назва: QueenRoom, Кол-во вільних місць: 4, Ціна: 2000,0
5 – Назва: KingRoom, Кол-во вільних місць: 5, Ціна: 5000,0
Виберіть:1
Гість додан в базу

```

Рис 3.3 – Зображення форми додавання нового гостя

### 3.2.1.2 Робота зі формою «Пошук гостей за ознакою»

Адміністратор повинен вибрати дію, в залежності від цього він може побачити гостей чи гостя за ознакою:

```

Виберіть дію, яку треба зробити:
1. Список усіх гостей
2. Пошук гостя за номером паспорту
3. Пошук гостей за датою виїзду
Відповідь: 

```

Рис 3.4 – Зображення меню з діями пошуку гостей за ознакою

1 – Ім'я: Дарина, Прізвище: Ханіна, В'їзд: 24.05.2023 00:00:00, Виїзд: 28.05.2023 00:00:00, Номер паспорту: 123456, Номер: SingleRoom

Рис 3.5 – Зображення результату вибору пошуку списка усіх гостей

```

Виберіть дію, яку треба зробити:
1. Список усіх гостей
2. Пошук гостя за номером паспорту
3. Пошук гостей за датою виїзду
Відповідь: 2
Введіть номер паспорту

```

Рис 3.6 – Зображення результату вибору пошуку гостя за номером паспорту

```

Виберіть дію, яку треба зробити:
1. Список усіх гостей
2. Пошук гостя за номером паспорту
3. Пошук гостей за датою виїзду
Відповідь: 3
Введіть день вид'їзду
Відповідь: 

```

Рис 3.7 – Зображення результату вибору пошуку гостя за датою виїзду

### 3.2.1.3 Робота зі формою «Виселення гостя»

Адміністратор повинен вибрати якого гостя треба видалити з бази:

```

1 – Ім'я: Дарина, Прізвище: Ханіна, В'їзд: 24.05.2023 00:00:00, Виїзд: 28.05.2023 00:00:00,Номер паспорту: 123456, Номер: SingleRoom
Якого гостя треба виселити:
Відповідь: 1
Чек за 4 днів – 1200,0

```

Рис 3.8 – Зображення результату виселення гостя

### 3.2.2 Робота зі сторінкою «Дія над номерами»

Виберіть дію, яку треба зробити:  
 1. Додати кімнату у базу  
 2. Пошук усіх номерів  
 3. Вихід  
 Відповідь:

Рис 3.9 – Зображення меню з діями над номерами

#### 3.2.2.1 Робота зі формою «Додати кімнату у базу»

Адміністратор повинен вибрати який номер треба додати у базу:

Виберіть назву кімнати (1 – SingleRoom, 2 – DoubleRoom, 3 – TwinRoom, 4 – QueenRoom , 5 – KingRoom)  
 Відповідь:

Рис 3.10 – Зображення результату вибору додати кімнату у базу

#### 3.2.2.2 Меню з усіма номерами

1 – Назва: SingleRoom, Усього місць: 1, Кол-во вільних місць: 0, Ціна: 300,0, Гості:  
 Ім'я: Дарина, Прізвище: Ханіна, В'їзд: 24.05.2023 00:00:00, Виїзд: 7.06.2023 00:00:00, Номер паспорту: 123456  
 2 – Назва: DoubleRoom, Усього місць: 2, Кол-во вільних місць: 2, Ціна: 500,0, Гості:  
 ще не має  
 3 – Назва: TwinRoom, Усього місць: 3, Кол-во вільних місць: 3, Ціна: 1000,0, Гості:  
 ще не має  
 4 – Назва: QueenRoom, Усього місць: 4, Кол-во вільних місць: 4, Ціна: 2000,0, Гості:  
 ще не має  
 5 – Назва: KingRoom, Усього місць: 5, Кол-во вільних місць: 5, Ціна: 5000,0, Гості:  
 ще не має

Рис 3.11 – Зображення результату вибору пошуку списку усіх номерів

## ВИСНОВОК

У ході виконання курсової роботи на тему «Адміністратор готеля» був створений консольний застосунок, у якому адміністратор готеля має змогу додавати гостей в обраний номер із п'яти, вже доданих у програму на старті, номерів. Також він може виселяти їх і видавати чек за кількість днів, які гість повинен був прожити у своєму номері. Крім того, адміністратор може переглядати список усіх гостей та список усіх номерів, з повною інформацією про них.

В результаті розробки був створений консольний застосунок. Були підвищені навички з програмування на мові C#, працювання з файлами у форматі json, вивчені правила використання трехступної архітектури. Стандарти оформлення документу звіту були засвоєні. Програма створювалась відповідно до заданого плану.

У подальшому програму можна удосконалити збільшенням функціоналу обробки та аналізу даних, та можливо оптимізацією деяких методів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. C# docs - get started, tutorials, reference. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 01.05.2023).
2. Учебник по языку C# 11 и платформе .NET 7. METANIT.COM - Сайт о программировании. URL: <https://metanit.com/sharp/tutorial/> (дата звернення: 05.05.2023).