Національний технічний університет України ««Київський політехнічний

інститут імені Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра Автоматизованих Систем Обробки Інформації та Управління

**КУРСОВА РОБОТА**

з Програмування - 2. Структури даних та алгоритми

на тему: «Готель. Сервіс бронювання номерів»

Студента 1 курсу ІС-03 групи

Спеціальності 126 Інформаційні системи та технології

Керівник викладач Вітковська Ірина Іванівна

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Київ - 2021

Національний технічний університет України “КПІ ім. Сікорського”

**Кафедра автоматизованих систем обробки інформації і управління**

Дисципліна Структури даних та алгоритми

**Спеціальність 126 «Інформаційні системи та технології»**

Курс 1 Група ІС-03 Семестр 2

**ЗАВДАННЯ**

на курсову роботу студента

Слободяна Дмитра Андрійовича

1. Тема проекту (роботи) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

2. Строк здачі студентом закінченого проекту (роботи) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані до проекту (роботи) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

5. Перелік графічного матеріалу ( з точним зазначенням обов’язкових креслень )  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

6. Дата видачі завдання \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ЗМІСТ**

[**ВСТУП** 4](#_Toc72144759)

[**РОЗДІЛ 1** 5](#_Toc72144760)

[**Аналіз предметної області** 5](#_Toc72144761)

[1.1 Сценарій роботи програми. 5](#_Toc72144762)

[1.2 Результати аналізу 5](#_Toc72144763)

[**РОЗДІЛ 2** 7](#_Toc72144764)

[**Кодування** 7](#_Toc72144765)

[**РОЗДІЛ 3** 8](#_Toc72144766)

[**ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ** 8](#_Toc72144767)

[3.1 Проект HotelApplication консольний додаток фронтенд 8](#_Toc72144768)

[3.2 Проект HotelLibrary бібліотека класів бекенд 11](#_Toc72144769)

[3.3 Стандартні класи і структури 14](#_Toc72144770)

[**РОЗДІЛ 4** 15](#_Toc72144771)

[**ТЕСТУВАННЯ** 15](#_Toc72144772)

[4.1 Тест 1. Перевірка на граничні умови початкового меню 15](#_Toc72144773)

[4.2 Тест 2. Перевірка на граничні умови опції Reserve 16](#_Toc72144774)

[4.3 Тест 3. Перевірка на граничні умови опції Book now 18](#_Toc72144775)

[4.4 Тест 4. Перевірка на граничні умови опції Watch order status 20](#_Toc72144776)

[4.5 Тест 5. Перевірка опції Exit program 22](#_Toc72144777)

[**ВИСНОВКИ** 23](#_Toc72144778)

[**ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ** 24](#_Toc72144779)

# ВСТУП

Подорожі невід'ємна складова життя кожної людини, якщо вона не є одноденною, ми потребуємо знайти місце для ночівлі. Працівники готелів можуть записувати замовлення номерів у журналі, але це буде не доречно у наш час. Тому вони потребують програмний сервіс бронювання номерів. У даній роботі ми дослідимо цю предменту область та маємо за мету створити програму яка дозволить бронювати номери у готелі.

# РОЗДІЛ 1

# Аналіз предметної області

## 1.1 Сценарій роботи програми.

Для поставленої задачі реалізуємо консольний додаток. Після запуску програми відкривається консольне вікно з меню з чотирьма подальшими діями в програмі: забронювати, замовити зараз, перевірити статус замовлення, вийти з програми; які можна вибрати через відповідний їм номер.

Якщо вибрати забронювати, буде запропоновано ввести по черзі день, місяць і рік заїзду, потім дані про дату виїзду з номеру. Після цього буде виведено список цін на різні категорії номерів, які різняться кількістю максимальних місць та вигляду з вікна. При виборі одного з варіантів, буде представлено кількість діб перебування в готелі, ціну за номер на вказаний час і ідентифікаційний номер замовлення.

Якщо вибрати замовити зараз, буде запропоновано ввести дату виїзду. Після цього буде виведено список цін на різні категорії номерів, які різняться кількістю максимальних місць та вигляду з вікна. При виборі одного з варіантів, буде представлено кількість діб перебування в готелі, ціну за номер на вказаний час і ідентифікаційний номер замовлення.

Якщо вибрати подивитись статус замовлення, буде запропоновано ввести номер замовлення. Після цього буде виведено статус замовлення: зайнятий, зарезервований на дату заїзду, закритий.

Якщо вибрати вихід – програма завершить свою роботу.

## 1.2 Результати аналізу

Для роботи в програмі користувач вводить всі вхідні дані через консоль. Вихідні дані виводяться на консоль.

Основні методи для того щоб получити вхідні дані на початку роботи програми це:

* введення дати;
* перевірка дати заїзду (відносно поточного часу);
* перевірка дати виїзду (відносно дати заїзду).

За допомогою цих методів ініціалізуються локальні змінні у головній функції. Потім визначається час, на який знімають номер. Виводиться список номерів один з яких користувач має вибрати, вибравши відповідний номер.

Значення номеру кімнати, дати заїзду та проміжку часу, на який знімають номер предається як параметр у метод замовити кімнату, який знаходиться у бібліотеці класів.

Метод замовити кімнату визначає ідентифікаційний номер замовлення, обраховує вартість замовлення. Вартість замовлення обраховується перемноженням часу, на який знімають номер, та вартісті номера за одну добу. Користувачу виводиться повідомлення, у якому вказано вартість замовлення та його ідентифікаційний номер, на цьому етапі створення замовлення завершується. Кожне створене замовлення зберігається у списку замовлень у бібліотеці.

Визначення статусу замовлення здійснюється за допомогою властивості, яка дає доступ до закритого поля. Під час виклику властивості статус замовлення спершу ініціалізується, а потім виводиться користувачу. Це зумовлено тим що статус замовлення залежить від поточного часу.

Усі вхідні дані контролюються на граничні умови через виключення.

# РОЗДІЛ 2

# Кодування

У вимогах до курсової роботи було запропоновано вибір мови програмування між C++ та C#.

Мова C# є мовою спеціально розроблено для об'єктно орієнтованого програмування, на відміну від C++, яку проапгрейдили з мови C, щоб писати на ній ООП.

Мова С# має багато переваг над С++, розглянемо головні з них. У C# ми не працюємо з пам'ятю безпосередньо, що захищає нас від багатьох неприємних ситуацій. C# набагато легше відладжувати, що сильно допоможе нам, під час етапу кодування. Ще однією перевагою є можливість легкого створення та використання власної бібліотеки класів.

Тому ми вибрали мову C# для проектування нашого програмного забезпечення, кодувати будемо у середовищі Visual Studio 2019.

# РОЗДІЛ 3

# ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розглянемо детально інтерфейс основних класів програми, діаграма яких показана на рисунку (3.1), та особливості їх реалізації.

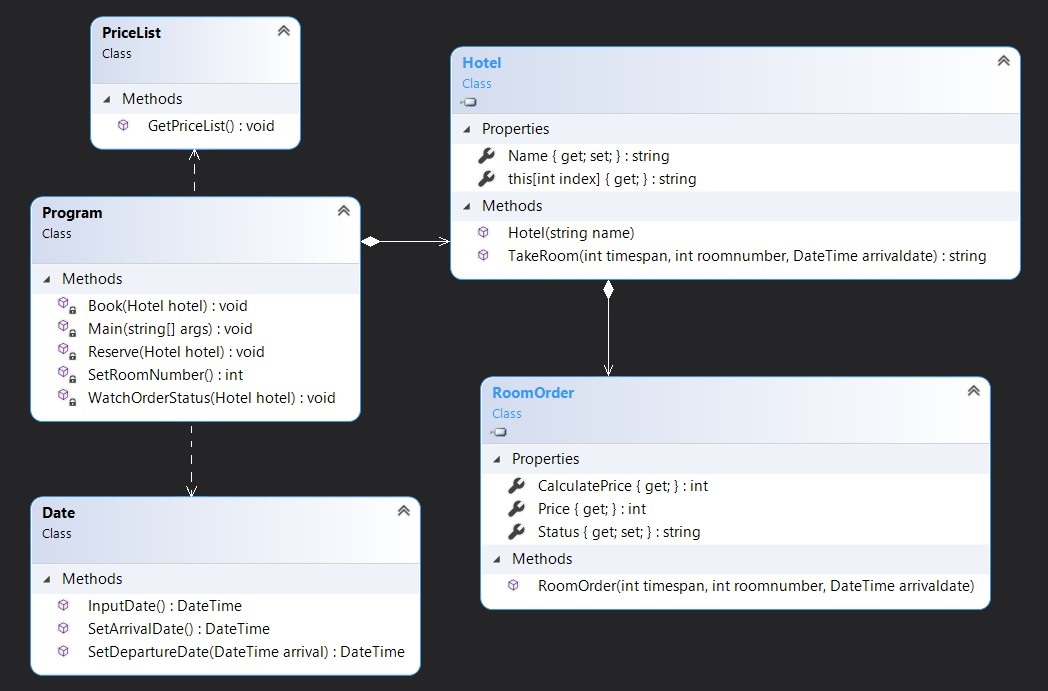


Рисунок 3.1 Діаграма класів

## 3.1 Проект HotelApplication консольний додаток фронтенд

Клас Program це клас головної функції, тут починає роботу наша програма на рисунку (3.2) показано його інтерфейс. Разом з ним розглянемо два залежних йому класи, функції яких він використовує – Date, PriceList, на рисунках (3.2), (3.3) показано їх інтерфейс.

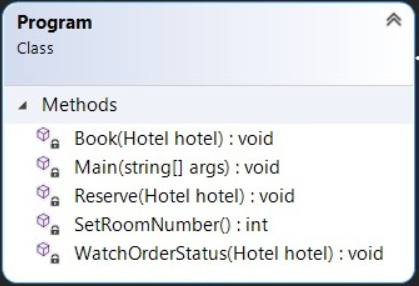


Рисунок 3.2 Опис класу Program

Клас Program має наступні атрибути:

* Main (string[] args) – статичний метод, головна функція, звідси починає роботу наша програма. У ній ми створюємо hotel типу Hotel, викликаються методи Reserve, Book або WatchOrderStatus, для подальної роботи в програмі;
* Reserve (Hotel hotel) – статичний метод призначений для опції забронювати номер. У ньому дата заїзду та виїзду ініціалізується за допомогою методів класу Date, викликається метод GetPriceList класу PriceList, timespan ініціалізовується як різниця дат заїзду і виїзду, з консолі вводиться roomnumber. За цими данними створюється нове замовлення викликом методу класу Hotel. Вкінці виводиться повідомлення про кількість днів перебування в готелі, вартість замовлення та його ідентифікаційний номер;
* Book (Hotel hotel) – статичний метод для опції замовити номер зараз. У ньому дата заїзду ініціалізується 12 годиною попереднього дня, якщо замовлення проводиться до 12 години, інакше – 12 годиною того дня коли проводиться замовлення; дата виїзду ініціалізується за допомогою методу класу Date, викликається метод GetPriceList класу PriceList, timespan ініціалізовується як різниця дат заїзду і виїзду, з консолі вводиться roomnumber. За цими данними створюється нове замовлення викликом методу класу Hotel. Вкінці виводиться повідомлення про кількість днів перебування в готелі, вартість замовлення та його ідентифікаційний номер;
* WatchOrderStatus (Hotel hotel) – статичний метод для перегляду статусу замовлення за його ідентифікаційним номером. Метод повертає рядок статусу;
* SetRoomNumber() – статичний метод, перевіряє чи існує введене значення номеру кімнати та повертає його, інакше просить ввести номер ще раз.

Клас PriceList призначений для виведення списку цін на номери.

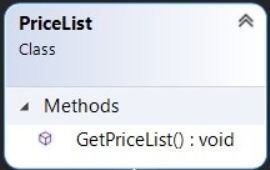


Рисунок 3.3 Опис клсу PriceList

Клас PriceList має наступний атрибути:

* GetPriceList () – статичний метод, який виводить список цін на номери.

Клас Date призначений для контрольованого введення дат заїзду на виїзду. Граничні умови такі:

1. дата заїзду для бронювання не може бути меншою за поточну дату;
2. дата виїзду не може бути меншою за дату заїзду.

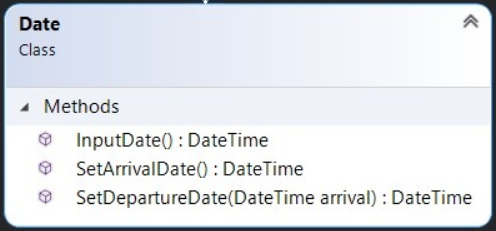


Рисунок 3.4 Опис класу Date

Клас Date має наступні атрибути:

* InputDate () – статичний метод призначений для введення дати та її повернення.
* SetArrivalDate () – статичний метод призначений для контролю правильності введення дати заїзду, та її повторного введення, якщо дата була введена неправильно. Повертає дату заїзду типу DateTime, як 12 година дня цієї дати.
* SetDepartureDate (DateTime arrivaldate) – статичний метод призначений для контролю правильності введення дати виїзду відносно дати arrivaldate, та її повторного введення, якщо дата була введена неправильно. Повертає дату виїзду типу DateTime, як 12 година дня цієї дати.

## 3.2 Проект HotelLibrary бібліотека класів бекенд

Клас RoomOrder це клас замовлень на рисунку (3.4) показано його інтерфейс. У ньому зберігаються всі дані про замовлення.

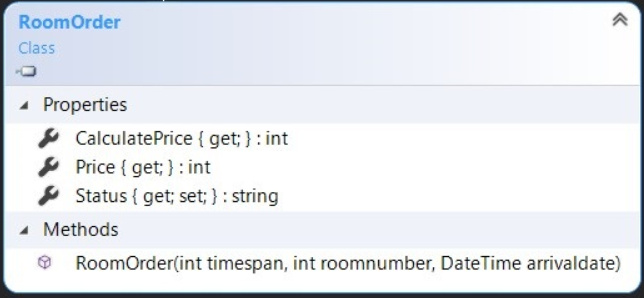


Рисунок 3.5 Опис класу RoomOrder

Клас RoomOrder має наступні атрибути:

* \_timespan – приватне поле цілочисельного типу, яке зберігає кількість днів на які зроблено замовлення;
* \_roomnumber – приватне поле цілочисельного типу, яке збіргає номер кімтати;
* \_price – приватне поле цілочисельного типу, яке зберігає вартість замовлення;
* \_arrivaldate – приватне поле типу DateTime, яке зберігає дату заїзду;
* \_status – приватне поле рядкового типу, яке зберігає статус замовлення;
* \_pricelist – приватне поле, цілочисельний масив, який доступний тільки для читання, зберігає ціни номерів на добу;
* RoomOrder (int timespan, int roomnumber, DateTime arrivaldate) – конструктор класу, що ініціалізовує всі поля класу крім поля \_status;
* CalculatePrice – властивість призначена для ініціалізації поля \_price в конструкторі класу, повертає ціле значення – \_timespan помножене на ціну номеру \_roomnumber, яка зберігається в списку \_pricelist;
* Price – властивість, яка повертає значення \_price;
* Status – властивість, яка має наступні get і set:
  1. get – повертає \_status;
  2. set:
     1. Якщо значення дати \_arrivaldate знаходиться на інтервалі від поточного часу до часу відправлення, то \_status ініціалізовується значенням зайнятий;
     2. Якщо значення дати \_arrivaldate менше на від поточного часу, то \_status ініціалізовується значенням заброньовано на дату заїзду;
     3. Інакше \_status ініціалізовуться значенням закрите.

Клас Hotel призначений для збереження списку всіх замовлень та можливості перегляду статусу деякого замовлення. Інтерфейс класу показаний на рисунку (3.5).

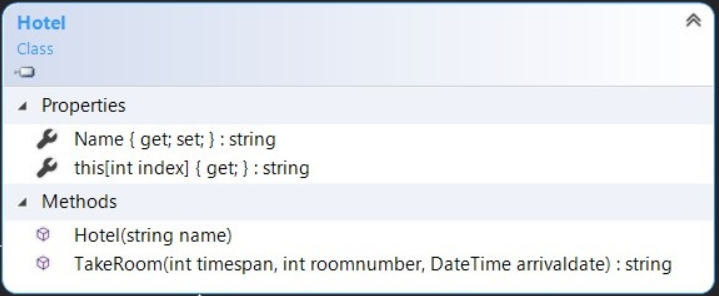


Рисунок 3.6 Опис класу Hotel

Клас Hotel має такі атрибути:

* \_id – приватне поле цілочисельного типу, яке зберігає ідентифікаційний номер останнього замовлення;
* roomorders – приватне поле, список типу List<RoomOrder>;
* Hotel (string name) – конструктор класу, ініціалізовуємо назву готелю автовластивістю Name;
* TakeRoom (int timespan, int roomnumber, DateTime arrivaldate) – метод призначений для створення нового замовлення. Поміщає створене замовлення у список \_roomOrders. Повертає рядок, що містить вартість замовлення \_price та ідентифікаційний номер \_id);
* this[int index] – індексатор реалізовує контрольований доступ до списку замовлень \_roomOrders, призначений для ініціалізації та повернення статусу деякого замовлення за допомогою властивості Status.

## 3.3 Стандартні класи і структури

У нашому програмному забезпечені ми викорисовували такі класи і структури:

1. List<T> – являє строго типізований список об'єктів, доступних за індексом. Підтримує методи для пошуку за списком, виконання сортування і інших операцій зі списками. Ми використовували метод Add – додати елемент в кінець списку.
2. Console – надає стандартні потоки для консольних додатків: вхідний, вихідний і потік повідомлень про помилки. Цей клас не успадковується. Ми використовували методи: WriteLine, Readline, ForegroundColor.
3. DateTime – структура, для роботи з часом в .Net. Вона представляє дату і час від 00:00:00 1 січня 0001 року до 23:59:59 31 грудня 9999 року. Ми використовували велику кількість властивостей та методів цієї структури.

# РОЗДІЛ 4

# ТЕСТУВАННЯ

Проведемо тестування створеного програмного забезпечення. Наша методика тестування основана на тому, що користувач може вводити граничні дані з якими програма не може працювати.

План тестування – ми покроково перевіримо чи передбачено програмою всі види граничних даних.

Запустимо програму. Початкове меню зображено на рисунку (4.1).

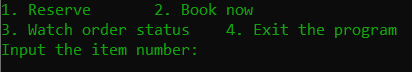


Рисунок 4.1 Початкове меню програми

Зауваження! Програма працює з часом, тому результи виконання залежать від поточного часу запуску програми. Всі наступні тести проведено 16.05.2021 о 21 годині.

## 4.1 Тест 1. Перевірка на граничні умови початкового меню

Введемо до початкового меню номер опції 0. Результат на рисунку (4.2).

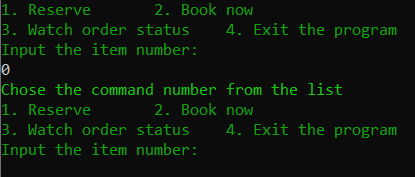


Рисунок 4.2

Введемо далі номер 5. Результат рисунок (4.3).

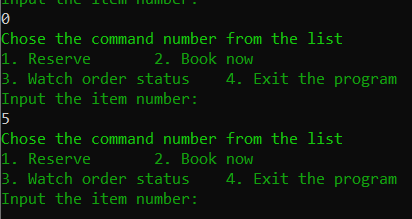


Рисунок 4.3

Введені нами значення були граничними, програма їх опрацювала і запропонувала ввести номер з списку. Тест 1 пройдено успішно.

## 4.2 Тест 2. Перевірка на граничні умови опції Reserve

Введемо до початкового меню номер опції 1. Результат на рисунку (4.4).

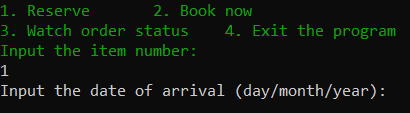


Рисунок 4.4

Тест 2.1. Перевірка на граничні умови дати заїзду.

Введемо сьогоднішню дату, якщо виключення буде опрацьовано введемо завтрішню дату – 17.05.2021. Результат – рисунок (4.5).

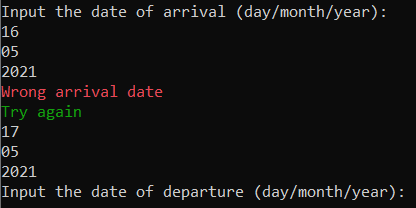
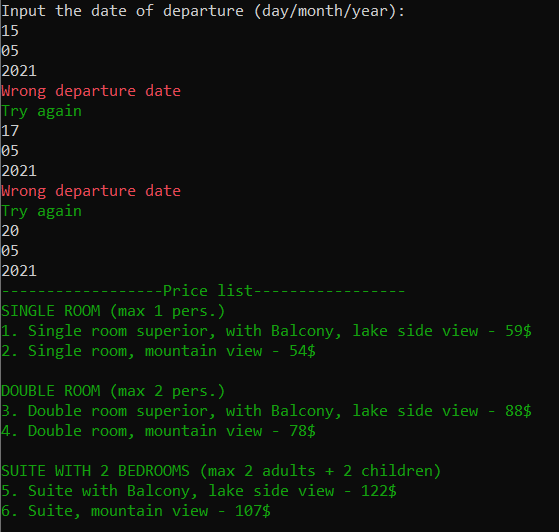


Рисунок 4.5

Тест 2.1 пройдено успішно.

Тест 2.2 Перевірка на граничні умови дати виїзду.

Введемо дату меншу за дату заїзду – 15.05.2021, якщо виключення буде опрацьовано введемо дату заїзду – 17.05.2021, потім – 20.05.2021. Результат – рисунок (4.6).



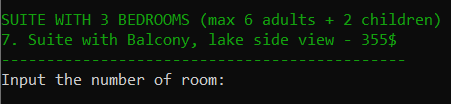


Рисунок 4.6

Тест 2.2 пройдено успішно.

Тест 2.3 Перевірка на граничні умови номеру кімнати.

Введемо номер 0, якщо виключення буде опрацьовано введемо номер 8, потім – 2. Результат – рисунок (4.7).

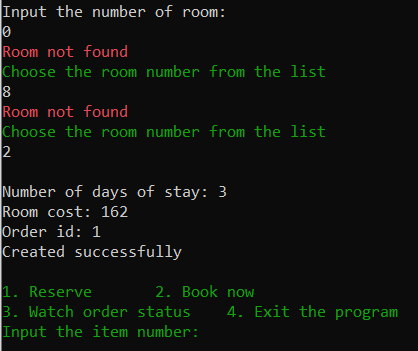


Рисунок 4.7

Програма вивела повідомлення про успішне створення замовлення, також кількість днів перебування в готелі, вартість номеру на цей час та ідентифікаційний номер замовлення. Всі дані обраховано правильно. Тест 2.3 пройдено успішно.

Тест 2 Про перевірку граничних умов опції Reserve пройдено успішно.

## 4.3 Тест 3. Перевірка на граничні умови опції Book now

Продовжимо роботу в програмі. Введемо до початкового меню номер опції 2. Результат на рисунку (4.8).

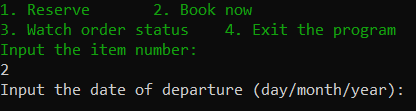


Рисунок 4.8

Тест 3.1. Перевірка на граничні умови дати виїзду.

Введемо сьогоднішню дату, якщо виключення буде опрацьовано введемо дату – 19.05.2021. Результат – рисунок (4.9).

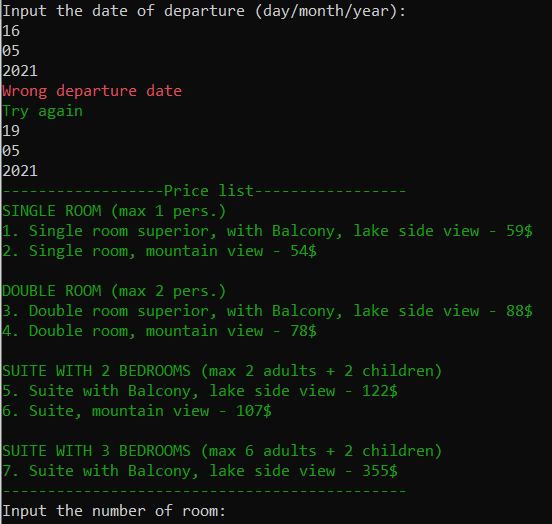


Рисунок 4.9

Тест 3.1 пройдено успішно.

Тест 3.2 Перевірка на граничні умови номеру кімнати.

Спершу просто натиснемо Enter, якщо виключення буде опрацьовано введемо номер 0, потім – 8, потім – 3. Результат – рисунок (4.10).

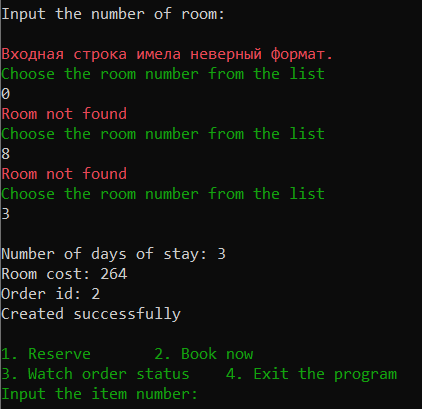


Рисунок 4.10

Програма вивела повідомлення про успішне створення замовлення, також кількість днів перебування в готелі, вартість номеру на цей час та ідентифікаційний номер замовлення. Всі дані обраховано правильно. Тест 3.2 пройдено успішно.

Тест 3 Про перевірку граничних умов опції Book now пройдено успішно.

## 4.4 Тест 4. Перевірка на граничні умови опції Watch order status

Продовжимо роботу в програмі. Введемо до початкового меню номер опції 3. Результат на рисунку (4.11).

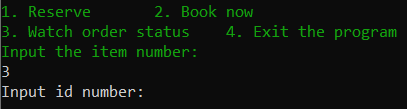


Рисунок 4.11

Тест 4.1. Перевірка на граничні умови ідентифікаційного номеру замовлення.

Введемо – 0. Результат – рисунок (4.12).

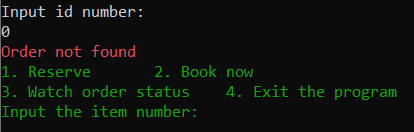


Рисунок 4.12

Програма не знаходить такого замовлення, подомляє нам і повертається до головного меню. Виберемо опцію 3 та введемо номер 1. Результат – рисунок (4.13).

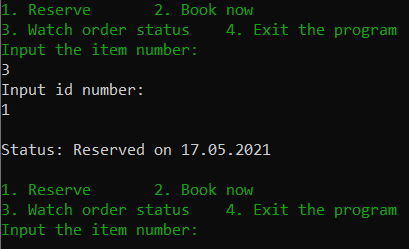


Рисунок 4.13

Програма вивела повідомлення про статус замовлення 1, статус правильний. Подивимося ще статус замовлення 2. Результат – рисунок (4.14).

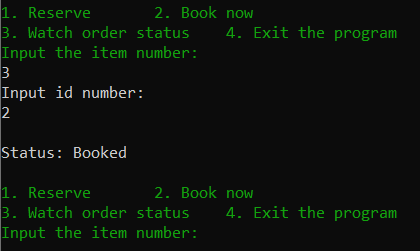


Рисунок 4.14

Програма вивела повідомлення про статус замовлення 1 та 2, статуси правильні. Тест 4.1 пройдено успішно.

Тест 4 Про перевірку граничних умов опції Watch order status пройдено успішно.

## 4.5 Тест 5. Перевірка опції Exit program

Продовжимо роботу в програмі. Введемо до початкового меню номер опції 4. Результат на рисунку (4.15).

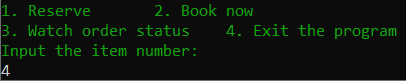


Рисунок 4.15

Натиснувши Enter програма завершує свою робрту.

Тест 5 Про перевірку опції Exit пройдено успішно.

Тестування розробленого програмного забезпечення пройшло успішно.

# ВИСНОВКИ

Проведена нами робота дає підстави зробити такі висновки.

Спершу ми проаналізували предметну область «готель», описавши сценарій роботи програми, яка призначена для бронювання номерів в готелі. Ми вирішили реалізувати консольний додаток. Результатом нашого аналзу є конкретизація постановки завдання. Вхідні і вихідні дані вводяться і виводяться в консоль. Основні операції і методи це введення дат заїзду та виїзу, створення замовлення та перегляд статусу вже створеного замовлення. Робота в програмі здійснюється вибором обмежених в кількості опцій.

Для створення програмного забезпеченн ми обрали мову програмування C#, працювали у середовищі Visual Studio 2019.

У наступному розділі ми описали створене програмне забезпечення. Детально розглянули інтерфейс класів та їх атрибути. Розроблене рішення у середовищі програмування складається з двох проектів:

* HotelApplication консольний додаток фронтенд – у цьому проекті міститься головна функція та весь інтерфейс додатку;
* HotelLibrary бібліотека класів бекенд – у цьому проекті містяться всі класи реалізації.

Також ми розглянули стандартні класи і структури, які використовували для реалізації нашого програмного забезпечення.

Ми провели тестування розробленої програми. Методика полягала у введенні граничних даних, з якими програма не може працювати. Всі тестування були пройдено успішно, програма опрацьовує всі види виключень.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. C# и .NET | Работа с датами и временем [Електронний ресурс] // metanit.com. – 2016. – Режим доступу до ресурсу: <https://metanit.com/sharp/tutorial/19.1.php>.
2. Price List [Електронний ресурс] // Impel Bohinj d.o.o. – Режим доступу до ресурсу: <http://www.bohinj-hotel.com/hotel/price-list.html>.
3. List<T> Класс (System.Collections.Generic) | MicrosoftDocs [Електронний ресурс] // Microsoft. – 2021. – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/api/system.collections.generic.list-1?view=net-5.0#definition>.
4. Console Класс (System) | Microsoft Docs [Електронний ресурс] // Microsoft. – 2021. – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/api/system.console?view=net-5.0>.
5. Slobodian D. 1 Course Work [Електронний ресурс] / Dmytro Slobodian – Режим доступу до ресурсу: <https://github.com/dimaslobodian/1-Course-Work-IS-03-Dmytro-Slobodian.git>.