КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра автоматизованих систем обробки інформації та управління

**КУРСОВА РОБОТА**

з об’єктно орієнтованого програмування

на тему: «Готель. Сервіс бронювання номерів»

Студента 1-го курсу групи ІС-01

Спеціальності «Інформаційні системи і технології»

Коваля Вадима

Керівник: Крамар Юлія Михайлівна

Кількість балів:

Національна оцінка:

Київ - 2021 рік

**Зміст записки**

**Вступ до записки3**

**Основна частина записки4**

Аналіз предметної області4

Кодування14

Тестування22

**Висновки29**

**Перелік посилань30**

Вступ до записки

Метою даної курсової роботи є реалізація готельного додатку: системи, що допоможе вирішити проблему розміщення туристів у номерах готелю. У програмі є наявна інформація про номери готелю, їх поточний стан (заброньовано, знімається), можливість адміністратора переглянути останні замовлення та загальний рахунок готелю.

Код «містить» масиви (бази даних) користувачів, номерів, замовлень тощо. До того ж, у програмі присутня велика кількість ситуацій, що допоможе готельному бізнесу уникнути казусних ситуацій та наслідків неуважних користувачів.

За допомогою консольного додатку для користувача реалізовується різний функціонал:

А) Як адміністратор, увівши пароль, він може переглянути основний рахунок готелю, побачити інформацію про номери, останні замовлення та навіть змінити пароль.

Б) Як гість (доступ без паролю), він може забронювати кімнату, зняти її, або навіть подовжити зняття номера. Інформація про номери також є доступною для гостя, як і для адміністратора.

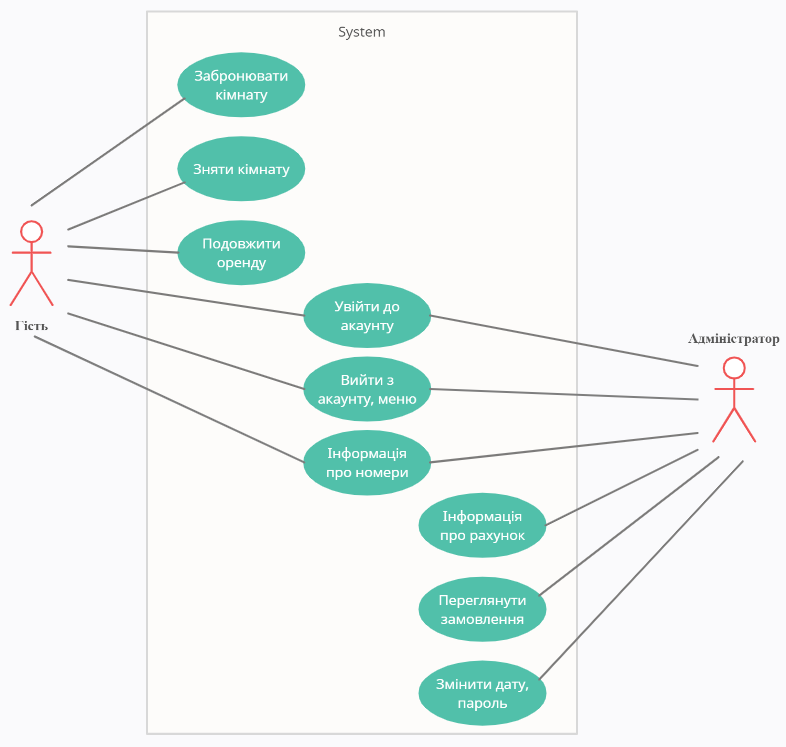
Додаток створено як універсальний, тому що провідною його мовою є англійська. Це є зручним для іноземців. Окрім цього, було детально опрацьовано «back» частину консольного додатку – усе написано білим кольором, але системні повідомлення (як-от вихід з системи, успішне бронювання) – жовтим, а помилки та підказки – червоним. Такий підхід спрощує користування додатком.

Основна частина записки

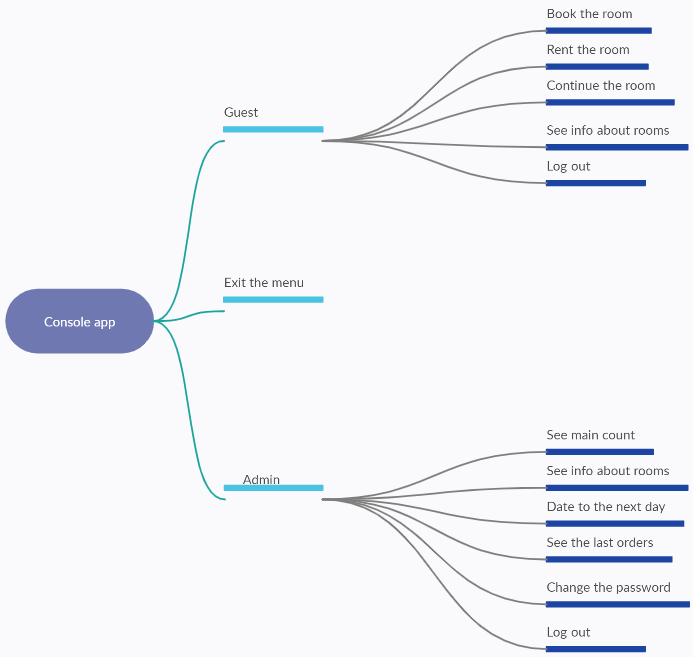
Аналіз предметної області

Наведена діаграма ілюструє функціонал програми. Для обох видів користувачів (гість та адміністратор) загальний функціонал є таким:

* Вхід до акаунту, що потребує уведення паролю для користувача типу адміністратор
* Бронювання кімнати
* Зняття кімнати (оренда)
* Подовження оренди (зняття)
* Перегляд загального грошового рахунку готелю
* Перегляд кількості замовлень та останніх п’яти з подробицями
* Змінення дати системи (для наглядності) та паролю користувача типу адміністратор
* Вихід з акаунту та меню

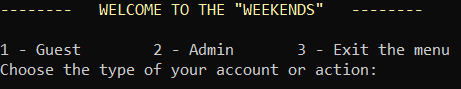


*Рис 1. Use Case діаграма програми*



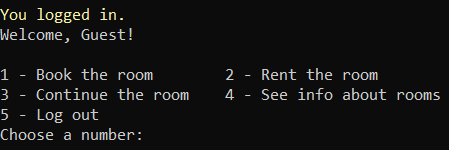
*Рис 2. Mind map діаграма програми (Сценарій)*

Увійшовши до системи, користувач бачить початкове меню, що дає можливість зайти як гість, адміністратор, або вийти з меню:



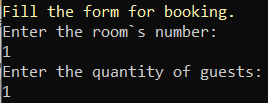
**Ситуація 1: Вхід до системи як гість готелю**

Якщо обрано тип користувача як гість, то йому надаються такі можливості, як бронювання, зняття (оренда), подовження номера. Окрім цього, можна переглянути інформацію про доступні номери в готелі та вийти з меню гостя:

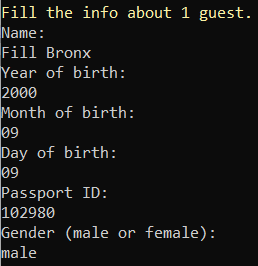


**Ситуація 1.1: Обрано пункт бронювання кімнати**

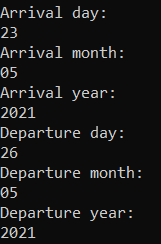
При бронюванні кімнати потрібно заповнити відповідну анкету для бронювання. Для цього необхідно увести номер кімнати та кількість гостей:



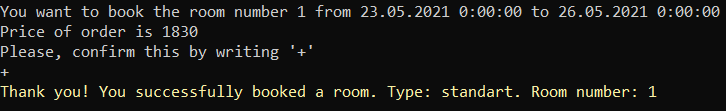
Після успішного заповнення попередніх двох пунктів користувач повинен закінчити анкету інформацією про гостей, що будуть перебувати в номері:



Заповнивши інформацію про усіх гостей (тут – один гість), потрібно надати інформацію про заплановану дату заселення та виселення з номеру:

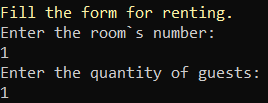


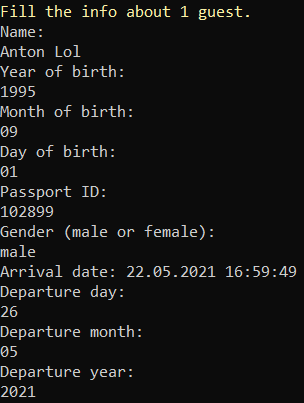
Остаточно програма виводить дані, що увів користувач, щоб він зміг перевірити їх, побачити ціну та погодитись (або ні) з замовленням. При успішному замовленні, є відповідне повідомлення:

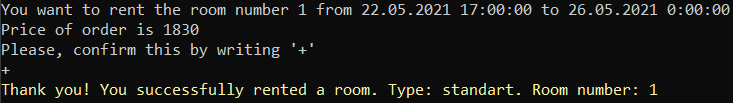


**Ситуація 1.2: Обрано пункт оренди (зняття) кімнати**

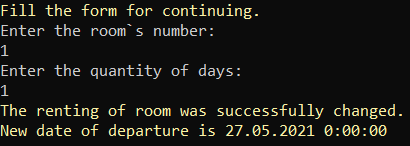
При знятті кімнати потрібно заповнити відповідну анкету для зняття. Для цього необхідно (аналогічно до п.1.1.) увести номер кімнати, кількість гостей, інформацію про гостей та дату виселення.





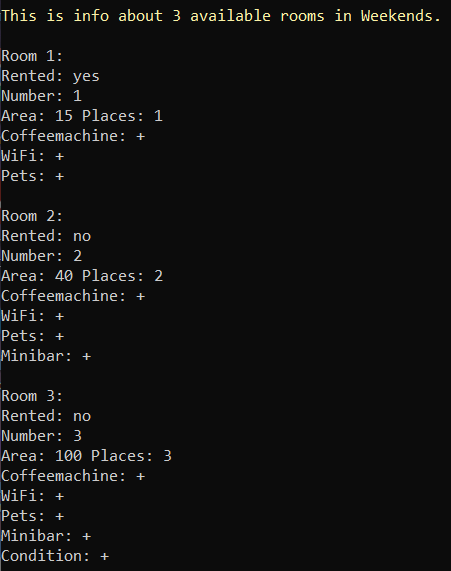


**Ситуація 1.3: Обрано пункт продовження оренди кімнати**



Для продовження оренди потрібно заповнити відповідну анкету. У ній необхідно увести номер кімнати, оренда якої має бути продовжена та кількість днів до продовження. За успішного заповнення виводиться повідомлення та оновлена дата кінця оренди кімнати.

**Ситуація 1.4: Обрано пункт перегляду номерів як гість**



Виводиться повідомлення з кількістю різних номерів в готелі та інформацію про кожен з них, що містить:

* Номер кімнати
* Площу
* Кількість місць
* Наявність додаткових послуг та зручностей:
* Standart Room:

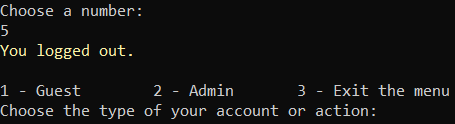
Кавоварка, WI-FI, дозвіл на перебування тварин

* Middle Room:

Кавоварка, WI-FI, дозвіл на перебування тварин, мінібар

* Lux Room:  
  Кавоварка, WI-FI, дозвіл на перебування тварин, мінібар, кондиціонер

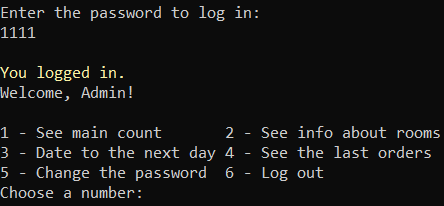
**Ситуація 1.5: Обрано пункт виходу з меню гостя**



Виводиться відповідне повідомлення про вихід з меню як гість. Надається можливість працювати знову з початковим меню програми.

**Ситуація 2: Вхід до системи як адміністратор готелю**

Якщо обрано тип користувача як адміністратор то йому надаються такі можливості, як перегляд основного рахунку готелю, зміна поточної дати, перегляд останніх замовлень та кількості усіх замовлень в готелі на даний час, зміна паролю акаунта адміністратора. Окрім цього, за аналогією з типом акаунту «гість», можна переглянути інформацію про доступні номери в готелі та вийти з меню адміністратора:

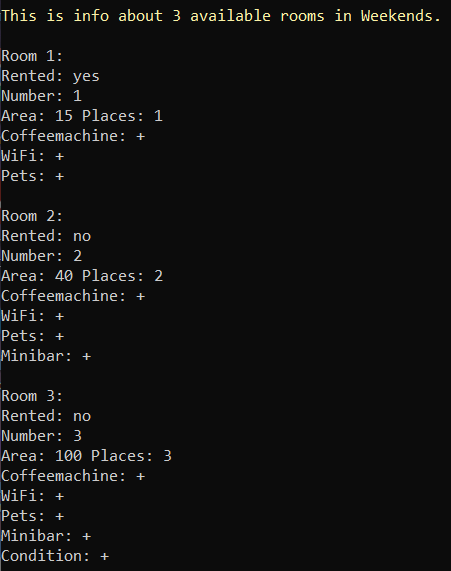


**Ситуація 2.1: Обрано пункт перегляду рахунку готелю**

Виводиться на екран повідомлення про суму на основному рахунку готелю станом на поточну дату.



**Ситуація 2.2: Обрано пункт перегляду номерів як адміністратор**



(Аналогічно до типу користувача «гість») Виводиться повідомлення з кількістю різних номерів в готелі та інформацію про кожен з них, що містить:

* Номер кімнати
* Площу
* Кількість місць
* Наявність додаткових послуг та зручностей:
* Standart Room:

Кавоварка, WI-FI, дозвіл на перебування тварин

* Middle Room:

Кавоварка, WI-FI, дозвіл на перебування тварин, мінібар

* Lux Room:  
  Кавоварка, WI-FI, дозвіл на перебування тварин, мінібар, кондиціонер

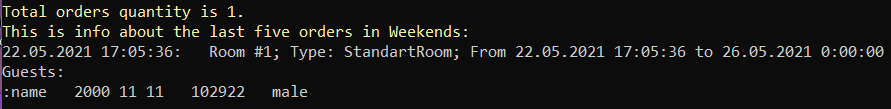
**Ситуація 2.3: Обрано пункт зміни датування системи**



При обранні пункту 3 профіля адміністратора, поточна дата системи збільшується на один день. Уведено даний функціонал для наочності опрацювання поточного стану номерів готелю.

**Ситуація 2.4: Обрано пункт перегляду замовлень**

При обранні цього пункту, адміністратор може побачити загальну кількість замовлень в готелі станом на поточну дату, а також останні п’ять замовлень:

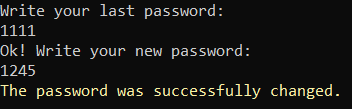


Відображаються такі деталі замовлень:

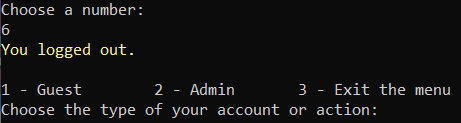
* Дата та час оформлення
* Номер кімнати
* Тип кімнати
* Дата заселення та виселення з кімнати
* Інформація про гостей:
* Ім’я
* Дата народження
* ID паспорту
* Стать

**Ситуація 2.5: Зміна паролю акаунта адміністратора**

За архітектурою програми, у адміністратора є можливість змінити пароль до свого акаунта. Для цього, задля безпеки, потрібно спочатку увести свій старий пароль, і якщо він збігається з поточним, то можна замінити пароль на новий. Також виводиться повідомлення системи, якщо операція успішна:



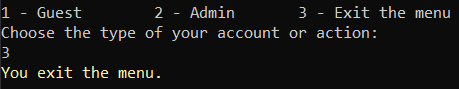
**Ситуація 2.6: Обрано пункт виходу з меню адміністратора**



Виводиться відповідне повідомлення про вихід з меню як гість. Надається можливість працювати знову з початковим меню програми.

**Ситуація 3: Вихід з головного меню**

При натисненні даної цифри, користувач може завершити роботу програми, паралельно вийшовши з головного меню додатку:



Кодування

***Готель***

Створено об’єкт готелю (класу Hotel) за допомогою конструктора з параметром, куди передавався параметр рядкового типу “Weekends” – це і є назвою готелю. Клас готелю містить масив кімнат, що заповнюються екземплярами посилання Room. Room – це допоміжний клас, що допомогає класифікувати кімнати. Він реалізує інтерфейс IRoom (методи Book та Rent) і є абстрактним. Присутні три похідні класи для класу Room, це – StandartRoom, MiddleRoom, LuxRoom – різні види номерів в готелі. Кожен з дочірних класів також має конструктор з параметрами для створення екземпляру кімнати цього типу. Клас Hotel додатково має масив замовлень, поточну дату системи і делегат NextDayDelegate для реалізації функціоналу адміністратора зміни поточної дати системи. При створенні екземпляру класа Hotel, в делегат додаються такі методи: 1 – ChangeData (зміна дати на 1 добу), 2 – CheckRooms (зміна стану номерів готелю)

***Бронювання***

Бронювання номерів відбувається завдяки методу Book класу Hotel. На вхід метод приймає делегат BookHandler для відпрацювання події бронювання, а також номер кімнати, масив інформації про гостей, дату заселення та виселення. Якщо кімната з відповідним номером є в готелі та не є заброньованою, то дані передаються далі в метод Book класу Room, що додає в масив нову заброньовану кімнату та заповнює поля про бронювання кімнати, гостей, дати проживання. Для реалізації бронювання також використано методи: FindRoom (приймає номер кімнати та надає інформацію чи є кімната з даним номером в готелі), Price (приймає номер кімнати, інформацію про гостей, дати проживання і надає інформацію про ціну), PayFor (приймає поле цілочисельного типу за посиланням – рахунок готелю, дані про бронювання, що беруться з масиву bookingData та «поповнює» рахунок готелю, ніби користувач оплатив своє перебування у номері).

***Зняття (оренда)***

Загалом схоже на процес бронювання номера. Для цього процесу використовується метод Rent класу Hotel. Якщо у готелі є кімната з таким номером, то дані передаються до екземплярного методу Rent (приймає масив гостей та дату виселення). Також використовується метод PayFor для зміни рахунку готелю.

! Зверніть увагу, що при знятті кімнати, на відміну від бронювання, не вказується дата заїзду – вона датується сьогоднішнім днем. До того ж, для кімнат з бронюванням використовується масив, бо декілька користувачів можуть одночасно забронювати кімнату, а от зняти (орендувати) одночасно ту ж кімнату на такий період може лише один гість.

***Подовження оренди***

Організоване за допомогою статичного методу ContinueRoom, що приймає об’єкт (екземпляр) класу Hotel. Після уведення номера кімнати та кількості днів для продовження, перевіряється чи існує кімната з заданим номером у готелі та чи знімається вона наразі кимось з гостей. Якщо так, то збільшується дата виселення за допомогою перевантаження оператора «+» у класі RentedRoom (приймає об'єкт класу RentedRoom (Room) та кількість днів – цілочисельне значення, додає цю кількість днів до поля ToDate поточного номера (кімнати) та повертає цю кімнату).

***Інформація про кімнати***

Інформація про кімнати виводиться користувачу за допомогою статичного методу RoomsInfo, що приймає об’єкт (екземпляр) класу Hotel. Метод виводить на екран загальну кількість номерів у готелі, а також за допомогою циклу «біжить» по усім кімнатам масиву та виводить про них такі дані, як номер кімнати, площу кімнати, кількість місць для проживання. За допомогою оператора is перевіряється до якого типу (похідного класу) відноситься кімната та виводиться відповідний перелік подробиць про дану кімнату. Для усіх типів виводиться інформація про наявність кавоварки, WI-FI та дозволу на перебування у номері тварин. Якщо типом є MiddleRoom, додатково надається інформація про наявність мінібару, якщо це LuxRoom, то йде річ про мінібар та кондиціонер. Ключовим є пункт Rented: (yes/no), що відображає поточний стан номеру – чи є він зайнятим, чи вільним.

***Зміна поточної дати***

Щоб правильно організувати зміну дати в системі, додано делегат nextDayDelegate, що включає в себе два методи без параметрів та з типом параметру, що повертається void. Перший метод - CheckRooms, що змінює стан номерів відносно сьогоднішнього дня. Якщо станом на поточну (вже змінену) дату кінцева дата оренди (зняття) номера збігається з поточною (зміненою) датою, то кімната переходить до статусу «вільно». Якщо поточна (оновлена) дата – дата початку заяви на бронювання номера, то номер автоматично переходить до статусу «зайнято», дані з bookingData цього номеру копіюються в інформацію даної кімнати в rentingData. Після перенесення інформації, поточна заява про бронювання видаляється і довжина масиву зменшується. Інший метод – ChangeDate, що збільшує кількість днів на одиницю в вбудованій структурі DateTime.

**Відслідковування замовлень**

Адміністратор може побачити загальну кількість замовлень у готелі, а також останні п’ять замовлень у подробицях. Для цього є метод GetLastOrders класу Hotel, що повертає поле класу – масив string[] ordersArray. Для додавання замовлення у масив замовлень присутній метод AddOrder, що приймає тип кімнати, її номер, інформацію про гостей (!в типі string), а також дати заселення\виселення + дату замовлення.

**Рахунок готелю**

Кожного разу при успішній оплаті за бронювання або зняття номеру, рахунок готелю поповнюється методом PayFor з посиланням на рахунок.

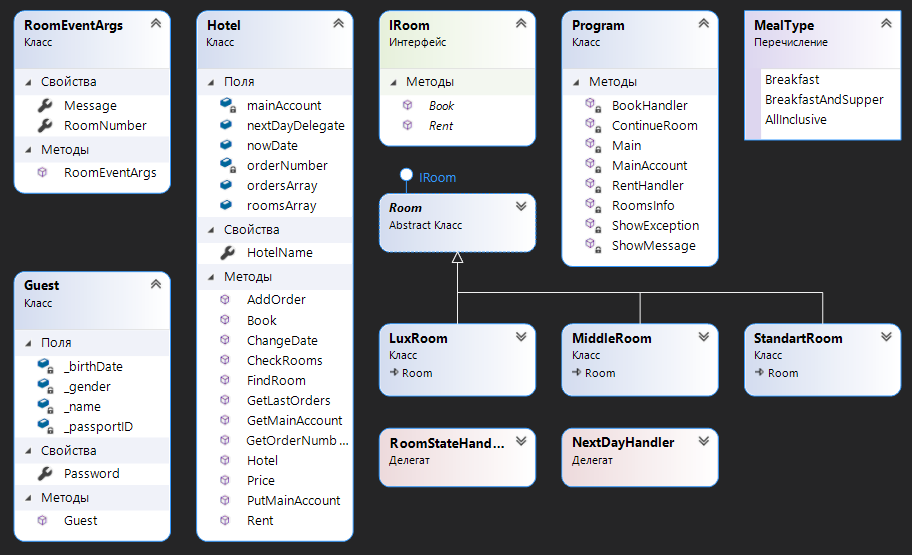
Опис програмного забезпечення

Вихідна програма містить 10 файлів, що сортовані за папками. З переліком можна ознайомитись нижче:

* GuestClass (папка)
* Guest.cs
* HotelClass (папка)
* Hotel.cs
* IRoom + Room (папка)
* IRoom.cs
* Room.cs
* RoomTypes (папка)
* LuxRoom.cs
* MiddleRoom.cs
* StandartRoom.cs
* Exceptions.cs
* RoomStateHandler.cs
* Program.cs

Надано перелік методів програмного коду:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Клас** | **Функція** | **Вхідні п-ри** | **Вихідні п-ри** | **Дія функції** |
| **Array** | Resize | ref T[] array, int newSize | void | Змінює к-ть елементів в масиві (розмір) |
| **Console** | WriteLine | string format, object arg0, object arg1 | void | Виводить строку на екран (консоль) |
| ReadLine | - | string | Зчитує уведену строку |
| **Convert** | ToInt32 | string | int32 | «Перетворює» тип змінної на цілочисельну |
| ToString | int32 | string | «Перетворює» тип змінної на строкову |
| ToChar | object | char | «Перетворює» тип змінної на символьну |
| **Date**  **Time** | Now | - | DateTime | Повертає поточну дату та час |
| AddDays | double | DateTime | Збільшує дату на к-ть днів |
| **Hotel** | Price | int, Guest[], DateTime, DateTime | int | Повертає ціну за вказаний номер |
| Book | RoomStateHandler, int, Guest[], int, int, int, int, int, int | void | Метод для бронювання номеру |
| Rent | RoomStateHandler, int, Guest[], int, int, int | void | Метод для зняття номеру |
| CheckRooms | - | void | Оновлює стан номерів відносно нової дати |
| FindRoom | int | Room/null | Повертає кімнату з заданим номером або null |
| ChangeDate | - | void | Збільшує к-ть днів поточної дати на один |
| GetMainAccount | - | int | Повертає значення грошового рахунку готелю |
| PutMainAccount | int | void | «Поповнює» грошовий рахунок готелю |
| AddOrder | string, int, string[], DateTime, DateTime, DateTime | void | Додає замовлення у масив замовлень |
| GetOrderNumber | - | int | Повертає загальну к-ть замовлень |
| DerLastOrders | - | string[] | Повертає масив замовлень |
| **Standart**  **Room** | Rent | Guest[], int, int, int | void | Організовує оренду номера |
| Book | Guest[], int, int, int, int, int, int | void | Організовує бронювання номера |
| PayFor | ref int, Guest[], DateTime, DateTime | void | «Оплачує» замовлення |
| GetPrice | Guest[], DateTime, DateTime | int | Повертає розраховану ціну замовлення |
| CoffeMachine | - | bool | Повертає наявність кавоварки\* |
| **Middle**  **Room** | Rent | Guest[], int, int, int | void | Організовує оренду номера |
| Book | Guest[], int, int, int, int, int, int | void | Організовує бронювання номера |
| PayFor | ref int, Guest[], DateTime, DateTime | void | «Оплачує» замовлення |
| GetPrice | Guest[], DateTime, DateTime | int | Повертає розраховану ціну замовлення |
| CoffeMachine | - | bool | \* |
| MiniBar | - | bool | Повертає наявність мінібару\*\* |
| **Lux**  **Room** | Rent | Guest[], int, int, int | void | Організовує оренду номера |
| Book | Guest[], int, int, int, int, int, int | void | Організовує бронювання номера |
| PayFor | ref int, Guest[], DateTime, DateTime | void | «Оплачує» замовлення |
| GetPrice | Guest[], DateTime, DateTime | int | Повертає розраховану ціну замовлення |
| CoffeMachine | - | bool | \* |
| Minibar | - | bool | \*\* |
| CondiTion | - | bool | Повертає наявність кондиціонеру |
| **Program** | ShowException | Exception | void | Виводить повідомлення про помилку |
| MainAccount | Hotel | void | Виводить стан рахунку |
| RoomsInfo | Hotel | void | Виводить інформацію про номери |
| ShowMessage | string | void | Виводить системне повідомлення |
| BookHandler | object, RoomEventArgs | void | Виводить повідомлення події |
| RentHandler | object, RoomEventArgs | void |
| ContinueRoom | Hotel | void | Організовує продовження оренди номера |

*Рис 3. Діаграма класів програми у VS*

**Узагальнення:** (Наслідування) Класи StandartRoom, MiddleRoom, LuxRoom є успадкованим від базового класу Room.

**Реалізація:** (Інтерфейс) Абстрактний клас Room реалізує інтерфейс IRoom.

**Агрегація:** (Слабкий зв’язок між класами) Клас Hotel має поле – масив об’єктів (екземплярів) іншого класу Room, тобто він може «включати» одну або більше кімнат.

**Композиція:** (Сильний зв’язок між класами) Без об’єкта одного класу не може існувати об’єкт іншого класу. Наприклад, екземпляр (об’єкт) класу А створюється з передаванням об’єкту класу В як параметр.

**Асоціація:** (Об’єкт-екземпляр класу як поле іншого) Абстрактний клас Room як закрите поле має пункт з перелічуваного типу MealType.

Тестування

Програмний код передбачає обробку 20 різних виключних ситуацій, серед яких 15 власних виключень, що містяться у класі Exceptions.cs.

**Методика тестування**

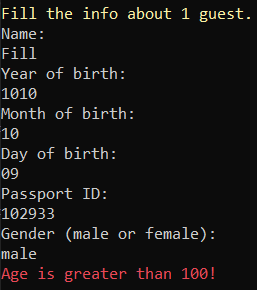
Тестування здійснюється для перевірки двох видів ситуацій: тих, що передбачені компілятором, а також тих, що є користувацькими – уведеними програмістом в код програми. Відповідні виключення можна розділити на підгрупи в залежності від призначення:

* Правильність заповнення інформації про гостя:
* GreatAgeException
* SmallAgeException
* WrongMonthException
* WrongDayException
* WrongGender
* IncorrectID
* IncorrectName
* Коректне уведення даних про номер:
* TooMuchGuests
* TooLittleGuests
* EndDateIsLessThanStartDate
* RoomNotFound
* RoomAlreadyBooked
* Правильність заповнення замовлення та вибору в меню:
* IncorrectNumber
* IncorrectOrder
* FormatException
* Коректний пароль в адміністраторському профілі:
* WrongPassword

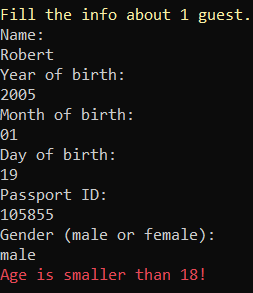
**Тести**

Проведено таку кількість тестів, що відповідає кількості можливих виключних ситуацій для користувача.

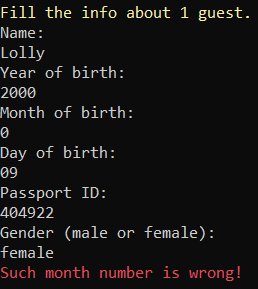
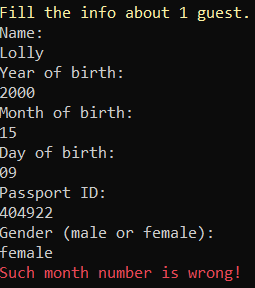
**GreatAgeException**: вік гостя перевищує 100 років



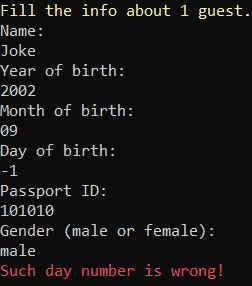
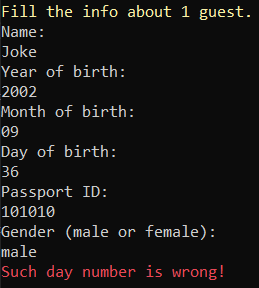
**SmallAgeException**: вік гостя менше, ніж 18 років



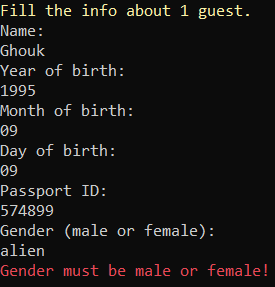
**WrongMonthException**: цифра місяця більше, ніж 12, або менше, ніж 1



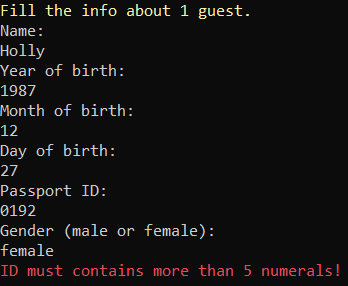
**WrongDayException**: цифра дня більше, ніж 31, або менше, ніж 1



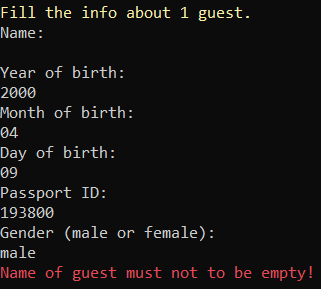
**WrongGender**: стать гостя не уведена як male (чоловік) або female (жінка)



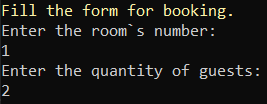
**IncorrectID**: уведений айді-код має менше 5-ти цифр



**IncorrectName**: уведене ім’я є пустим

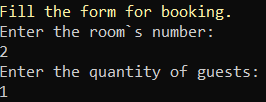


**TooMuchGuests**: уведена кількість гостей перевищує передбачувану даною кімнатою (Наприклад, для кімнати №1 кількість гостей – один)



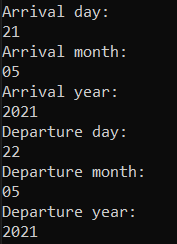


**TooLittleGuests**: уведена кількість гостей менша за передбачувану даною кімнатою (Наприклад, для кімнати №2 кількість гостей – два)



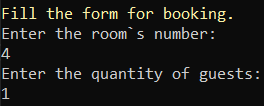


**EndDateIsLessThanStartDate**: дата заселення та виселення раніше, ніж сьогодні (Тестування актуальне станом на 23.05)



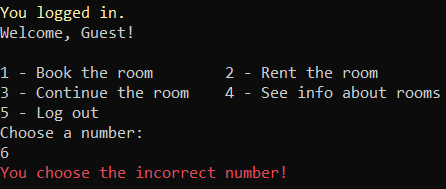


**RoomNotFound**: кімната з заданим номером не існує в готелі (Наприклад, у поточній версії є кімнати з номерами 1, 2, 3)

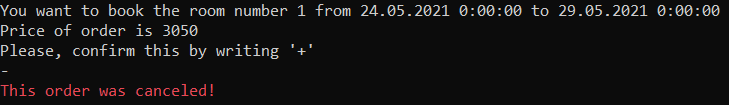




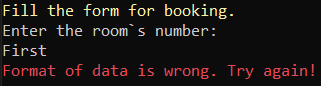
**IncorrectNumber**: уведено неправильний номер меню (Наприклад, у меню гостя є лише 5 «кнопок»)



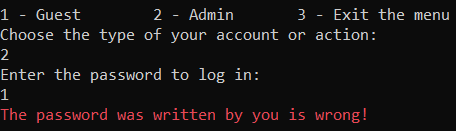
**IncorrectOrder**: замовлення було анульоване



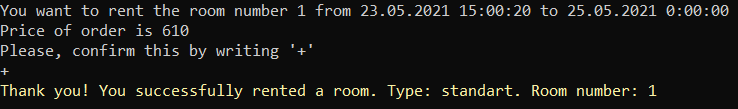
**FormatException**: уведено дані в неправильному форматі (Наприклад, номер кімнати уведений буквами)

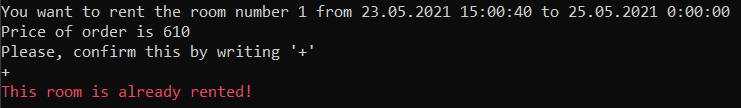


**WrongPassword**: пароль від адмінпрофіля є неправильним



**RoomAlreadyBooked (Rented)**: кімната вже є зайнятою на задані дати





Висновки

За допомогою даного програмного продукту можна безпечно та комфортно організувати роботу для структур готельного бізнесу: передбачено бронювання номерів, їх оренду та подовження оренди. Корисним є те, що гість завжди може ознайомитись з актуальним списком кімнат, а адміністратор з поточним загальним рахунком готелю. Окрім цього, програмний код передбачає велику кількість ситуацій, що можуть призвести до некоректних замовлень та роботи програми в цілому. Такі ситуації обробляються та виводять повідомлення про помилку користувачеві.

У курсовій роботи задіяні основні парадигми та принципи об’єктно-орієнтованого програмування – провідної теми даної діяльності. Застосовано основні «компоненти» ООП, такі, як:

* класи,
* властивості та автовластивості,
* наслідування,
* поліморфізм,
* абстрактні класи та інтерфейси,
* перевантаження операторів тощо.

Перелік посилань

1. Язык программирования C# 7 и платформы .NET и .NET Core (8-ое издание) / Э. Троелсен, Ф. Джепикс, 2020 – 1328 с.
2. C# 4.0 Полное руководство / Г. Шилдт, 2011 – 1056 с.
3. Программирование на C# для начинающих / А. Васильев, 2018 – 584 с.
4. C# Tutorial – Full Course for Beginners / freecodecamp.org
5. C# Уроки От новичка к профессионалу / SimpleCode Youtube
6. Изучение языка С# / Гоша Дударь YouTube
7. Видеокурс по языку программирования C# / Eugene Popov YouTube
8. Отношение классов – от UML к коду / habr.com
9. Полное руководство по C# 9 и .NET 5 / metanit.com/sharp/
10. C# / stepik.org