

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА
ПОЯСНЮВАЛЬНА ЗАПИСКА
з дисципліни “Об’єктно-орієнтоване програмування”
Обмін житла

Керівник проф.
Студент гр. ПЗП-22-1

Бондарєв В.М.
Крот Ю.В.

Комісія:

проф. Бондарєв
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

Крот Юлія Вікторівна

1 Тема проекту: Обмін житла

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>№</i>	<i>Назва етапу</i>	<i>Термін виконання</i>
1	Видача теми, узгодження і затвердження теми	13.02.2023 - 14.03.2023 р.
2	Формулювання вимог до програми	07.04.2023 р.
3	Розробка підсистем та функцій програми	30.05.2023 - 06.06.2023 р.
5	Тестування і доопрацювання розробленої програмної системи.	05.06.2023 – 07.06.2023 р.
6	Оформлення пояснювальної записки, додатків, графічного матеріалу	07.06.2023 – 08.06.2023 р.
7	Захист	09.06.2023 – 16.06.2023 р.

Студент

Крот Юлія Вікторівна

Керівник

Бондарєв Володимир Михайлович

«7» червня 2023 р.

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 39 с., 1 табл., 19 рис., 2 джерела.

ОБМІН ЖИТЛА, МЕШКАНЦІ, ЗВІТ, ООП, .NET, МОВА C#

Метою роботи є розробка програми «Обмін житла», яка буде надавати користувачам можливість пошуку різноманітного житла для обміну.

В результаті отримана програма, що дозволяє зберігати список житла, власники якого бажають приймати участь у обміні. Є можливість реєстрації клієнтів з житлом, видалення не актуального профілю. Також можна вказувати характеристики свого житла: ім'я власника, район, площа, етаж, тип житла: будинок чи квартира. Користувач може переглянути список зареєстрованого житла, вказати вимоги, за якими відбудеться пошук, та знайти для себе житло, що ідеально підійде для обміну.

В процесі розробки використано середовища Microsoft Visual Studio 2022, платформи .NET 7.0, мова програмування C#.

ЗМІСТ

ВСТУП	6
1 ОПИС ВИМОГ	7
1.1 Сценарії використання	7
1.2 Функціональні вимоги	10
2 ПРОЕКТУВАННЯ ПРОГРАМИ	15
3 ІНСТРУКЦІЯ КОРИСТУВАЧА	17
3.1 Встановлення й запуск програми	17
3.2 Видалення програми	17
3.3 Використання програми	17
ВИСНОВКИ	23
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	24
ДОДАТОК А	25

ВСТУП

Метою курсової роботи є закріплення знань, набутих при вивченні дисципліни “Об’єктно-орієнтоване програмування” , а також придбання навичок у розробці програмних програм, на прикладі додатку “Обмін житла”.

Метою проекту є розробка консольного застосування, яке дасть користувачам можливість пошуку різноманітного житла для обміну. Це спростить пошук пропозицій житла, яке буде відповідати вимогам клієнта. Після реєстрації користувач зможе отримати доступ до переліку з варіантами на обмін, надіслати запрошення, прийняти або відхилити власне. Завдяки використанню програми, власники житла зможуть обмінюватися житлом без зайвих турбот.

Програма, отримана в результаті курсової роботи, призначена для зручного пошуку. Основним призначенням цієї програми є забезпечення користувача інформацією щодо інших власників, які так само бажають приймати участь в обміні житлом, та надання інформації щодо розміщення житла.

1 ОПИС ВИМОГ

1.1 Сценарії використання

Сценарій 1. Реєстрація житла

Передумова

Користувач запустив програму і відкрив головне вікно програми.

Основний сценарій

1. Після запуску програма видає список запропонованих варіантів дії, з них обирається “Зареєструвати нове житло”.
2. Користувач вводить своє ім’я; вказує тип жила; вводить район, площу та поверх; перераховує переваги свого житла та вказує свої вимоги.
3. Програма реєструє це житло.

Додатковий сценарій

1. Після запуску програма видає список запропонованих варіантів дії, з них обирається “Зареєструвати нове житло”.
2. Користувач вводить своє ім’я; вказує тип жила; вводить район, площу та поверх; перераховує переваги свого житла та вказує свої вимоги.
3. Під час введення даних програма встановлює, що дані не коректні.
4. Програма сповіщає користувача про помилку в даних і надає можливість заново ввести інформацію.

Сценарій 2. Вибір з існуючого житла

Передумова

Користувач запустив програму і відкрив головне вікно програми.

Основний сценарій

1. Після запуску програма видає список запропонованих варіантів дії, з них обирається “Обрати інсуюче житло”.
2. Користувач бачить список житла, обирає своє та натискає відповідну цифру.
3. Програма вітає користувача та вказує його район. Подальші дії відбуватимуться з акаунту власника цього житла.

Додатковий сценарій

1. Після запуску програма видає список запропонованих варіантів дії, з них обирається “Обрати інсуюче житло”.
2. Інші користувачі поки що не зареєструвалися, тому програма сповіщає про те що зареєстрованого житла ще нема, та пропонує додати нове.

Сценарій 3. Пошук житла

Передумова

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

Основний сценарій

1. У головному меню користувач обирає розділ “пошук житла”, де він зможе ознайомитися з можливими варіантами.
2. Програма виводить список зареєстрованого житла, інформацію про нього, та очікує коли користувач зробить вибір.
3. Користувач обирає житло, відповідною цифрою. Програма створює запит та повідомляє про це користувача.

Додатковий сценарій

1. У головному меню користувач обирає розділ “пошук житла”, де він зможе ознайомитися з можливими варіантами.

2. Зареєстрованих користувачів ще не існує, тому програма сповіщує про це користувача та каже що пошук неможливий. Користувач повертається до головного меню.

Сценарій 4. Перегляд надісланих запитів

Передумова

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

Основний сценарій

1. У головному меню користувач обирає розділ “Переглянути надіслані пропозиції”, де він зможе побачити свій запит.
2. Користувач скасовує цей запит, та отримує можливість надіслати новий. Повертається до головного меню

Додатковий сценарій

1. У головному меню користувач обирає розділ “переглянути надіслані пропозиції”, де він зможе побачити свій запит.
2. Жодного запиту ще не було надіслано, тому програма сповіщає про це та повертає користувача до головного меню.

Сценарій 5. Видалення житла

Передумова

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

Основний сценарій

1. У головному меню користувач обирає розділ “Видалити житло”, де він зможе побачити варіанти житла для видалення.
2. Власник житла обирає яке хоче видалити, вводить відповідну цифру та enter.

Програма видаляє житло, більше його не можливо обрати для запиту на обмін.

Додатковий сценарій

1. У головному меню користувач обирає розділ “Видалити житло”, де він зможе побачити варіанти житла для видалення.
2. Користувач обирає житло, під яким він зайшов до профілю.
3. Програма повідомляє про неможливість такої програми, уточнивши що потрібно вийти з акаунту цього житла, перш ніж видаляти його.
4. Користувач повертається до головного меню.

1.2 Функціональні вимоги

Функція 1. Реєстрація житла

На екран виводиться головне меню, де користувач має можливість перейти до розділу реєстрування житла, натиснувши цифру 2 (рис. 1.1).

```
*-----Вітаємо в головному меню!-----*  
Оберіть один з перелічених варіантів  
  
1. Обрати інсуюче житло  
2. Зареєструвати нове житло  
3. Вийти з додатку
```

Рисунок 1.1 – Головне меню програми

Після вибору розділу “Зареєструвати нове житло” користувач потрапляє до меню, де він вносить інформацію для реєстрації житла. Є поле для визначення типу житла, у разі якщо користувач вказує відповідь, яка не відповідає вимогам програми, виводиться прохання надати відповідь ще раз.

В цьому полі реєстр не важливий, програма сама збереже відповідь в відповідному форматі. (рис 1. 2.)

Деякі з пунктів мають свої вимоги, а саме:

- Поле ім'я має не більше 2 слів;
- Поле тип будинку: приймається відповідь “дім”, або ”квартира”;
- Вимоги та інформація про житло надається через кому.

Якщо всі дані введено вірно, житло буде створене та додано до списку.

Програма сповістить про це користувача та подякує за виконану роботу.

-----Створюєм профіль з житлом-----

Введіть ім'я власника житла: Андрій

Оберіть тип вашого житла. Це дім чи квартира? ддд
Надана інформація некоректна. Вкажіть тип житла: дім чи квартира

Оберіть тип вашого житла. Це дім чи квартира? Дім

Введіть ваш район: Холодногірський

Введіть площу: 65.9

Введіть поверх (або кількість поверхів в вашому будинку): 5

Введіть переваги вашого житла (через кому):
є балкон, ремонт

Введіть власні вимоги до житла (через кому):
розташування, велика територія

Житло додане! Дякуєм.

Рисунок 1.2 – Додавання нового житла

Функція 2. Вибір з існуючого житла

На екран виводиться головне меню, де користувач має можливість обрати існуюче житло (рис. 1.1).

Після переходу до розділу, на екран виведено вже зареєстровані профілі. Користувач обирає своє за допомогою відповідної цифри (рис. 1.3). На екрані відображається не вся інформація, користувач може побачити такі поля:

- Ім'я;
- Район;
- Тип житла.

Користувач обирає своє житло, програма вітається та називає район.
Далі переходимо до головного меню (рис 1.4).

```
*-----Вітаємо! Оберіть ваше житло-----*
1. Ім'я: Юлія. Район: Шевченківський. Тип житла: Дім.
2. Ім'я: Андрій. Район: Слобідський. Тип житла: квартира.
3. Ім'я: Микола. Район: Холодногірський. Тип житла: квартира.
4. Ім'я: Андрій. Район: Холодногірський. Тип житла: Дім.

> 1
Вітемо, Юлія! Ваш район: Шевченківський
```

Рисунок 1.3 – Список житла

```
*-----Головне меню-----*
1. Додати житло
2. Видалити житло
3. Пошук житла
4. Переглянути надіслані пропозиції
5. Вихід
```

Рисунок 1.4 – Головне меню після аутентифікації

Функція 3. Пошук житла

З головного меню (рис. 1.4) користувач обирає розділ “Пошук житла” та потрапляє до нового розділу (рис. 1.5).

Користувач отримує доступ до інформації про житло інших користувачів, з якою він може ознайомитися та обрати житло щоб створити запрошення на обмін. Доступні поля: ім'я власника, район, тип житла, площа та етаж.

Після створення запрошення програма перенаправляє до головного меню.

-----Пошук житла-----

Оберіть номер житла для створення запрошення на обмін.
Для повернення до попередньої сторінки натисніть 0:

1. Ім'я власника: Юлія; Район: Шевченківський; Тип житла: Дім; Площа: 42.7; Етаж: 1.
2. Ім'я власника: Андрій; Район: Слобідський; Тип житла: квартира; Площа: 79.7; Етаж: 6.
3. Ім'я власника: Микола; Район: Холодногірський; Тип житла: квартира; Площа: 40.3; Етаж: 2.
4. Ім'я власника: Андрій; Район: Холодногірський; Тип житла: Дім; Площа: 65.9; Етаж: 5.

> 2

Запрошення відправлено до житла у районі: Слобідський!

Рисунок 1.5 – Розділ пошуку житла

Функція 4. Перегляд надісланих запитів

З головного меню (рис. 1.4) користувач переходить до розділу “Переглянути надіслані пропозиції” (рис. 1.5).

-----Запит, який ви надіслали-----

Ім'я власника: Микола; Район: Холодногірський; Тип житла: квартира; Площа: 40.3; Етаж: 2.

1. Скасувати
2. Вихід

Рисунок 1.6 - Розділ перегляду

Доступні до перегляду поля ті самі, що й в функції 3. В данному розділі користувач має можливість скасувати свій запит (натиснувши цифру 1), або повернутися до попередньої сторінки (натиснувши цифру 2).

У разі якщо до цього користувач не надіслав запит, програма сповіщає його про це, та повертає до головного меню.

Функція 5. Видалення житла

З головного меню (рис. 1.4) користувач переходить до розділу “Видалити житло” (рис. 1.7).

-----Видалення житла-----

Оберіть житло, яке бажаєте видалити. Для повернення до попередньої сторінки натисніть 0.

- 1.Ім'я: Андрій. Район: Слобідський. Тип житла: квартира.
- 2.Ім'я: Микола. Район: Холодногірський. Тип житла: квартира.
- 3.Ім'я: Юлія. Район: Шевченківський. Тип житла: Квартира.

Рисунок 1.7 – Розділ для видалення зі списком житла

Користувачу доступний список зареєстрованого житла, з такими полями: Ім'я власника, район та тип житла. Для видалення житла потрібно обрати відповідну цифру, після натискання enter програма видалить житло зі списку. У разі якщо користувач не хоче нічого видаляти, треба надати 0 та enter – програма поверне його до попередньої сторінки.

-----Видалення житла-----

Оберіть житло, яке бажаєте видалити. Для повернення до попередньої сторінки натисніть 0.

1.Ім'я: Андрій. Район: Слобідський. Тип житла: квартира.

2.Ім'я: Юлія. Район: Шевченківський. Тип житла: Квартира.

2

Це ваше житло, вийдіть з профілю, перш ніж видаляти його.

Рисунок 1.8 – Повідомлення у разі спроби видалити активний профіль житла

Якщо обрати цифру, яка відповідає за житло користувача – профіль житла не буде видалено (рис. 1.8). Програма сповістить власника про неможливість цієї дії та поверне його до головного меню.

2 ПРОЕКТУВАННЯ ПРОГРАМИ

На етапі проектування була обрана консольна архітектура.

Загальна сруктура програми базується на головному класі. Його основним завданням є ініціалізація допоміжних класів та взаємодія з ними. Така структура дозволяє виконувати різноманітні операції з файлами, які зберігають дані, виводити необхідні повідомлення до консолі та оброблять інформацію, яку надає користувач. Взаємодію частин програми та співвідношення між класами відображено за допомогою діаграми (рис 2.1).

Дані зберігатимуться у форматі JSON.

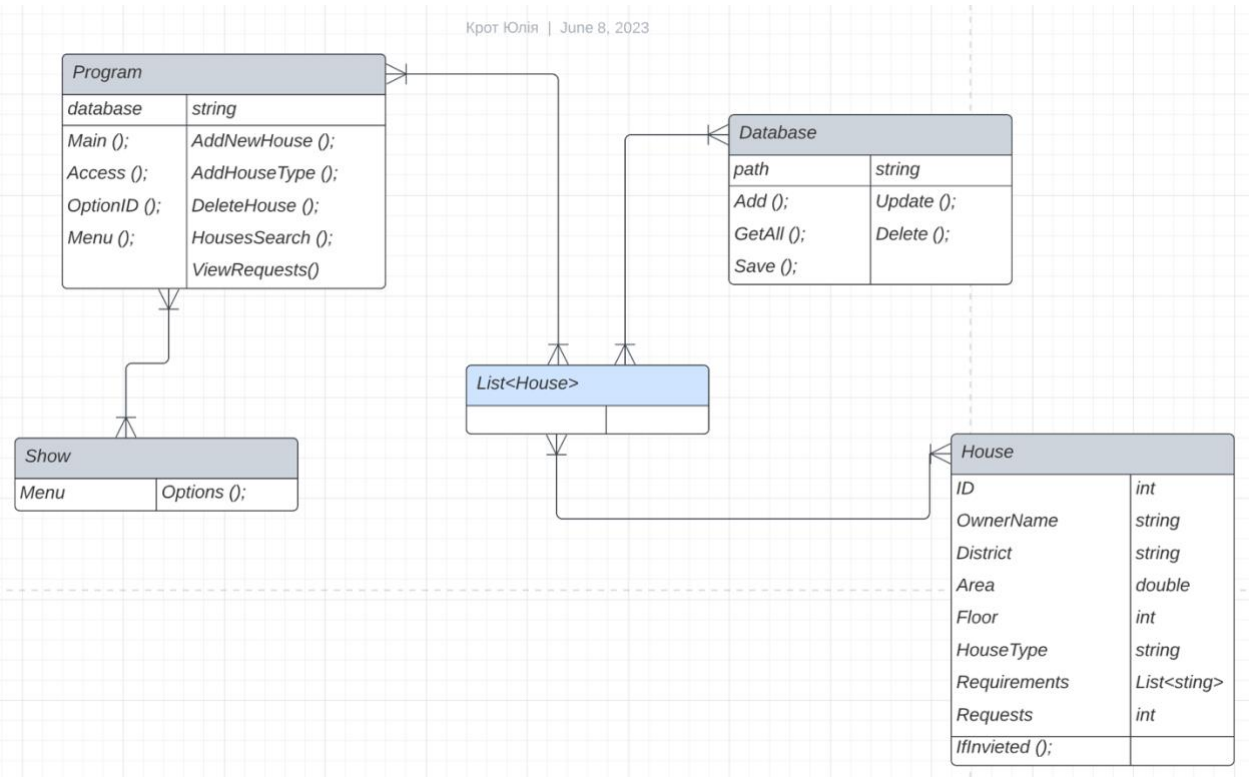


Рисунок 2.1 - Взаємодія класів у програмі

Проект має наступні класи:

- Program – головний клас;

- House – клас житла;
- Database – клас для завантаження та збереження даних;
- Show – допоміжний клас, для виводу інформації до консолі.

3 ІНСТРУКЦІЯ КОРИСТУВАЧА

3.1 Встановлення й запуск програми

Процедура встановлення програми: завантаження .zip архіву з програмою, та розархівування . Скопіювати шлях до папки, де збережено файл Program.cs, та за допомогою терміналу перейти до цієї папки. Після чого ввести команду `dotnet run` і програму буде запущено.

3.2 Видалення програми

Для видалення необхідно видалити папку проекту з Вашого комп'ютеру.

3.3 Використання програми

Коли користувач запускає програму, відкривається головне меню (рис. 3.3.1), де є список можливих варіантів використання програми. Є можливість обрати вже зареєстроване житло, додати своє, або завершити роботу з програмою.

-----Вітаємо в головному меню!-----
Оберіть один з перелічених варіантів

- 1. Обрати існуюче житло**
- 2. Зареєструвати нове житло**
- 3. Вийти з додатку**

Рисунок 3.1.1 – Головне меню програми після запуску

Якщо обрати цифру 2 та натиснути `enter`, ми перейдемо до розділу, де зможемо зареєструвати житло (рис. 3.1.2). Після того, як всі дані будуть успішно введені, житло буде додано, програма сповістить про це користувача.

-----Створюєм профіль з житлом-----

Введіть ім'я власника житла: Юлія

Оберіть тип вашого житла. Це дім чи квартира? Квартира

Введіть ваш район: Шевченківський

Введіть площу: 75.8

Введіть поверх (або кількість поверхів в вашому будинку): 4

Введіть переваги вашого житла (через кому):
розташування, комфорт, близько до шкіл та магазинів

Введіть власні вимоги до житла (через кому):
приватність, безпека

Житло додане! Дякуєм.

Рисунок 3.1.2 – Приклад заповнення даних для додавання житла

Якщо до цього ви вже додавали житло, то достатньо буде натиснути цифру 1 та enter у головному меню , а потім обрати своє житло (рис. 3.1.3). Якщо жодного зареєстрованого житла ще немає у програмі, а користувач перейшов до меню “Обрати існуюче житло”, програма сповістить його, що варіанти для вибору відсутні та запропоную створити власний профіль, після чого поверне власника до головного меню програми.

-----Вітаємо! Оберіть ваше житло-----

1. Ім'я: Андрій. Район: Слобідський. Тип житла: квартира.
2. Ім'я: Микола. Район: Холодногірський. Тип житла: квартира.
3. Ім'я: Андрій. Район: Холодногірський. Тип житла: Дім.
4. Ім'я: Юлія. Район: Шевченківський. Тип житла: Квартира.

> 4

Вітемо, Юлія! Ваш район: Шевченківський

Рисунок 3.1.3 – Меню з вибором житла для аутентифікації

Після виконання будь-якого з цих варіантів, користувач потрапляє до головного меню, яке має більше можливостей, оскільки власник житла зареєстрований (рис. 3.1.4).

```

*-----Головне меню-----*
1. Додати житло
2. Видалити житло
3. Пошук житла
4. Переглянути надіслані пропозиції
5. Вихід

```

Рисунок 3.1.4 – Головне меню програми

Оберемо пункт “Пошук житла”. На цей момент можемо побачити список варіантів для обміну та деяку інформацію про житло, яка допоможе користувачу при виборі.

Інструкція для використання цього розділу знаходиться вгорі, користувач обирає відповідний номер житла, до якого хоче відправити запрошення та натискає enter. Програма обробляє запит та надсилає запит, сповіщає про це користувача (рис.3.1.5). Після цього додаток поверне користувача до головного меню.

```

*-----Пошук житла-----*
Оберіть номер житла для створення запрошення на обмін.
Для повернення до попередньої сторінки натисніть 0:

1. Ім'я власника: Андрій; Район: Слобідський; Тип житла: квартира; Площа: 79.7; Етаж: 6.
2. Ім'я власника: Микола; Район: Холодногірський; Тип житла: квартира; Площа: 40.3; Етаж: 2.
3. Ім'я власника: Андрій; Район: Холодногірський; Тип житла: Дім; Площа: 65.9; Етаж: 5.
4. Ім'я власника: Юлія; Район: Шевченківський; Тип житла: Квартира; Площа: 75.8; Етаж: 4.
> 2

Запрошення відправлено до житла у район: Холодногірський!

```

Рисунок 3.1.5 – Меню пошуку житла

Якщо жоден з варіантів не підходить, користувач може повернутися на попередню сторінку, натиснувши цифру 0 та enter.

Тепер перевіримо чи відправився наш запит. Для цього у головному меню (рис. 3.1.4) перейдемо до розділу “Переглянути надіслані пропозиції”, натиснувши цифру 4 та enter. Опинимося у розділі, де можемо побачити наш запит.

Є кілька варіантів що ми можемо зробити далі. Перший - це скасувати запит, натиснувши 1 та enter. В такому випадку програма видалить існуючий запит та сповістить про це користувача (рис. 3.1.6).

Другий варіант – це повернутися до головного меню, програма нічого не змінює.

```
*-----Запит, який ви надіслали-----*
Ім'я власника: Микола; Район: Холодногірський; Тип житла: квартира; Площа: 40.3; Етаж: 2.
1. Скасувати
2. Вихід
> 1
Запит скасовано
```

Рисунок 3.1.6 – Перегляд існуючого запиту

Якщо до цього запит не був надісланий, або якщо користувач перейшов до цього розділу після скасування запиту – програма повідомляє про відсутність запитів та повертає користувача до головного меню.

```
*-----Запит, який ви надіслали-----*
Ви поки що не надіслали жодного запиту.
```

Рисунок 3.1.7 – Повідомлення програми, у разі відсутності запитів

Також є розділ “Видалити житло”. Перейдемо до нього з головного меню (рис. 3.1.4). Можемо побачити список зареєстрованого житла, щоб видалити одне з них натиснемо відповідну цифру та enter. Житло буде видалене (рис. 3.1.8).

```
*-----Видалення житла-----*
Оберіть житло, яке бажаєте видалити. Для повернення до попередньої сторінки натисніть 0.
1.Ім'я: Андрій. Район: Слобідський. Тип житла: квартира.
2.Ім'я: Микола. Район: Холодногірський. Тип житла: квартира.
3.Ім'я: Юлія. Район: Шевченківський. Тип житла: Квартира.
```

Рисунок 3.1.8 – Видалення житла

Тепер перейдемо до розділу “Пошук житла”, і засвідчимо, що житла, яке ми щойно видалили дійсно більше нема у пропозиціях (рис. 3.1.9).

-----Пошук житла-----

Оберіть номер житла для створення запрошення на обмін.
Для повернення до попередньої сторінки натисніть 0:

1. Ім'я власника: Андрій; Район: Слобідський; Тип житла: квартира; Площа: 79.7; Етаж: 6.
2. Ім'я власника: Микола; Район: Холодногірський; Тип житла: квартира; Площа: 40.3; Етаж: 2.
3. Ім'я власника: Юлія; Район: Шевченківський; Тип житла: Квартира; Площа: 75.8; Етаж: 4.

Рисунок 3.1.9 – Розділ для пошуку житла після видалення одного з варіантів

Зауважте, що при спробі видалення власного житла, це буде неможливо, адже користувач зараз виконує всі дії особисто і не може видалити активний профіль (рис. 3.1.10). Програма сповістить про це та поверне користувача до головного меню.

-----Видалення житла-----

Оберіть житло, яке бажаєте видалити. Для повернення до попередньої сторінки натисніть 0.

1. Андрій – Слобідський – 87
2. Микола – Холодногірський – 66
3. Юлія – Шевченківський – 27

> 3

Це ваше житло, вийдіть з профілю, перш ніж видаляти його.

Рисунок 3.1.10 – Повідомлення у разі намагання видалити активне житло

Щоб вийти з програми, необхідно перейти до головного меню (рис. 3.1.4) та обрати п'ятий пункт “Вихід”. Потім вибрати третій пункт “Вийти з додатку” (рис. 3.1.1). Програма закінчить свою роботу та попрощається з користувачем.

ВИСНОВКИ

Програма, отримана в результаті виконання курсової роботи відповідає заявленим цілям. Розроблено програму, яка дає користувачам можливість пошуку різноманітного житла для обміну. Функціонал програми дозволяє користувачу додавати житло, видаляти його, надсилати запити та переглядати їх.

Програму можна вдосконалити розширенням можливостей та функціоналу додатку, покращенням інтерфейсу та дизайну програми на більш зручний та сучасний. Додаток “Обмін житла” перспективний, тому можна знайти багато варіантів удосконалення.

Після розробки консольного додатку були закріплені знання з дисципліни “Об’єктно-орієнтоване програмування” та набуто навичок з кодування, проектування. Також було покращено знання з мови C# та отримано досвід розробки програмних продуктів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бондарєв В.М. Електронний учбовий посібник з дисципліни «Основи програмування» URL: <http://tss.co.ua:5555/> (дата звернення: 02.06.2023)
2. .NET documentation | Microsoft Learn URL: <https://learn.microsoft.com/en-us/dotnet/> (дата звернення: 02.06.2023)

ДОДАТОК А

Код програми

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Reflection;
using System.Xml.Linq;
using Newtonsoft.Json;

namespace HouseExchange
{
    public class House
    {
        public int ID { get; set; }
        public string OwnerName { get; set; }
        public string District { get; set; }
        public double Area { get; set; }
        public int Floor { get; set; }
        public string HouseType { get; set; }
        public List<string> Info { get; set; }
        public List<string> Requirements { get; set; }
        public int Requests { get; set; }
        public bool HasBeenInvited { get; set; }

        public House(int id, string ownerName, string district, double area, int floor,
string houseType, List<string> info, List<string> requirements)
```



```
{
    ID = id;
    OwnerName = ownerName;
    District = district;
    Area = area;
    Floor = floor;
    HouseType = houseType;
    Info = info;
    Requirements = requirements;
    Requests = -1;
    HasBeenInvited = false;
}

public bool IfInvited()
{
    return Requests != -1;
}

}

class Database
{
    private string path;

    public Database(string path)
    {
        this.path = path;
    }

    public List<House> GetAll()
```

```
{  
    if (!File.Exists(path))  
    {  
        return new List<House>();  
    }  
  
    string json = File.ReadAllText(path);  
    return JsonConvert.DeserializeObject<List<House>>(json);  
}  
  
public void Add(House house)  
{  
    List<House> houses = GetAll();  
    houses.Add(house);  
    SaveAll(houses);  
}  
  
public void Delete(int id)  
{  
    List<House> houses = GetAll();  
  
    houses.RemoveAll(u => u.ID == id);  
    SaveAll(houses);  
}  
  
public void Update(House inp)  
{  
    List<House> houses = GetAll();  
    int i = 0;
```

```

foreach (House house in houses)
{
    if (house.ID == inp.ID)
    {
        houses[i] = house;
        SaveAll(houses);
        break;
    }

    i++;
}

}

public bool IsEmpty()
{
    return !File.Exists(path) || new FileInfo(path).Length == 0;
}

public void SaveAll(List<House> houses)
{
    string json = JsonConvert.SerializeObject(houses);
    File.WriteAllText(path, json);
}

}

class Program
{
    private static Database database;

    static void Main(string[] args)

```

```
{
    database = new Database("database.json");

    while (true)
    {
        Console.WriteLine("*-----Вітаємо в головному меню!-----*");
        Console.WriteLine("Оберіть один з перелічених варіантів");
        Console.WriteLine("");
        Show.Menu(new string[] { "Обрати інсуюче житло", "Зареєструвати
нове житло", "Вийти з додатку" });

        int choice = Show.Options(3);
        Console.Clear();

        switch (choice)
        {
            case 1:
                Access();
                break;
            case 2:
                AddNewHouse(true);
                break;
            case 3:
                Console.WriteLine("До побачення!");
                return;
        }

        Console.Clear();
    }
}
```

```

    }

    static void Access()
    {
        if (database.IsEmpty())
        {
            Console.WriteLine("Зарестрованого житла нема, додайте нове.");
            Thread.Sleep(2000);
            return;
        }

        Console.WriteLine("*-----Вітаємо! Оберіть ваше житло-----*");
        List<House> houses = database.GetAll();

        for (int i = 0; i < houses.Count; i++)
        {
            House house = houses[i];
            Console.WriteLine($"{i + 1}.  Ім'я: {house.OwnerName}.  Район: {house.District}. Тип житла: {house.HouseType}.");
        }

        int choice = Show.Options(houses.Count);
        House house1 = houses[choice - 1];
        House selectedHouse = house1;

        Console.WriteLine($"Вітаємо, {selectedHouse.OwnerName}! Ваш район: {selectedHouse.District}\n");
        Menu(selectedHouse);
    }

```

```

        Console.Clear();
    }

    static void AddNewHouse(bool Working)
    {
        static string AddOwnerName()
        {
            Console.WriteLine("\nВведіть ім'я власника житла: ");
            string res = Console.ReadLine();

            if (res.Split(" ").Length > 2)
            {
                Console.WriteLine("Надана інформація некоректна. Введіть тільки ім'я");
                return AddOwnerName();
            }

            return res;
        }

        static string AddHouseType()
        {
            Console.WriteLine("\nОберіть тип вашого житла. Це дім чи квартира? ");
            string res = Console.ReadLine();
            if (res.ToLower() != "дім" && res.ToLower() != "квартира")
            {
                Console.WriteLine("Надана інформація некоректна. Вкажіть тип житла: дім чи квартира");
                return AddHouseType();
            }
        }
    }

```

```

    return res;
}

```

```

Console.WriteLine("*-----Створюєм профіль з житлом-----*");
string ownerName = AddOwnerName();
string houseType = AddHouseType();

```

```

Random random = new Random();
int id = random.Next(10, 100);

```

```

Console.Write("\nВведіть ваш район: ");
string district = Console.ReadLine();

```

```

Console.Write("\nВведіть площу: ");
double area = double.Parse(Console.ReadLine());

```

```

Console.Write("\nВведіть поверх (або кількість поверхів в вашому
будинку): ");
int floor = int.Parse(Console.ReadLine());

```

```

List<string> info = new List<string>();
Console.WriteLine("\nВведіть переваги вашого житла (через кому): ");
string[] infoArray = Console.ReadLine().Split(',');
info.AddRange(infoArray);

```

```

List<string> requirements = new List<string>();
Console.WriteLine("\nВведіть власні вимоги до житла (через кому): ");
string[] requirementsArray = Console.ReadLine().Split(',');

```

```
requirements.AddRange(requirementsArray);
```

```
House newHouse = new House(id, ownerName, district, area, floor,
houseType, info, requirements);
```

```
database.Add(newHouse);
```

```
Console.WriteLine("\nЖитло додане! Дякуєм.");
```

```
Thread.Sleep(2000);
```

```
Console.Clear();
```

```
if (Working)
```

```
{
```

```
    Menu(newHouse);
```

```
}
```

```
}
```

```
static void Menu(House house)
```

```
{
```

```
    while (true)
```

```
{
```

```
    Console.WriteLine("*-----Головне меню-----*");
```

```
    Show.Menu(new string[] { "Додати житло", "Видалити житло",
"Пошук житла", "Переглянути надіслані пропозиції", "Вихід" });
```

```
    int choice = Show.Options(5);
```

```
    Console.Clear();
```



```

switch (choice)
{
    case 1:
        AddNewHouse(false);
        break;
    case 2:
        DeleteHouse(house.ID);
        break;
    case 3:
        HousesSearch(house);
        break;
    case 4:
        ViewRequests(house);
        break;
    case 5:
        Console.WriteLine("До побачення!");
        return;
}

Console.Clear();

}

}

static void DeleteHouse(int HouseId)
{
    Console.WriteLine("*-----Видалення житла-----*");
    Console.WriteLine("Оберіть житло, яке бажаєте видалити. Для повернення до попередньої сторінки натисніть 0.");
}

```

```
List<House> houses = database.GetAll();

for (int i = 0; i < houses.Count; i++)
{
    House house = houses[i];
    Console.WriteLine($"{i + 1 }.{house.OwnerName} - {house.District} -
{house.ID}");
}

int choice = Show.Options(houses.Count);

if (choice == 0) { return; }

House selectedHouse = houses[choice - 1];

if (selectedHouse.ID == HouseId)
{
    Console.WriteLine("\nЦе ваше житло, вийдіть з профілю, перш ніж
видаляти його.");
    Thread.Sleep(2000);
    return;
}

if (selectedHouse.IfInvited())
{
    House invited = OptionID(selectedHouse.Requests);
    invited.Requests = -1;
    invited.HasBeenInvited = false;
    database.Update(invited);
}
```

```

        database.Delete(selectedHouse.ID);

        Console.WriteLine($"Добре! Житло з району:{selectedHouse.District}
видалено.");
    }

    static void HousesSearch(House house)
    {
        Console.WriteLine("*-----Пошук житла-----*");

        List<House> houses = database.GetAll();

        if (houses.Count == 0)
        {
            Console.WriteLine("На жаль, зареєстрованого житла поки що немає,
тому пошук неможливий");
            Thread.Sleep(2000);
            return;
        }

        Console.WriteLine("Оберіть номер житла для створення запрошення на
обмін.\nДля повернення до попередньої сторінки натисніть 0:");

        for (int i = 0; i < houses.Count; i++)
        {
            House h_option = houses[i];
            Console.WriteLine($" \n{i + 1}. Ім'я власника: {h_option.OwnerName};
Район: {h_option.District}; Тип житла: {h_option.HouseType}; Площа:
{h_option.Area}; Етаж: {h_option.Floor}.");
        }
    }

```

```

int choice = Show.Options(houses.Count);
if (choice == 0) return;
House selectedHouse = houses[choice - 1];

house.Requests = selectedHouse.ID;
selectedHouse.Requests = house.ID;
selectedHouse.HasBeenInvited = true;

Console.WriteLine($"{selectedHouse.District}\nЗапрошення відправлено до житла у район:
{selectedHouse.District}!");
database.Update(house);
database.Update(selectedHouse);
Thread.Sleep(2000);
}

static House OptionID(int ID)
{
    List<House> all = database.GetAll();
    foreach (House selectedHouse in all)
    {
        if (selectedHouse.ID == ID)
        {
            return selectedHouse;
        }
    }
    return null;
}

```

```

static void ViewRequests(House house)
{
    static void CancelRequest(House house, House houseRequest)
    {
        house.Requests = -1;
        houseRequest.Requests = -1;
        house.HasBeenInvited = false;
        houseRequest.HasBeenInvited = false;

        database.Update(houseRequest);
        database.Update(house);
    }

    Console.WriteLine("*-----Запит, який ви надіслали-----*");
    House houseRequest = OptionID(house.Requests);

    if (houseRequest == null)
    {
        Console.WriteLine("Ви поки що не надіслали жодного запиту.");
        Thread.Sleep(2000);
        return;
    }

    Console.WriteLine($"Ім'я власника: {houseRequest.OwnerName};
    Район: {houseRequest.District}; Тип житла: {houseRequest.HouseType}; Площа:
    {houseRequest.Area}; Етаж: {houseRequest.Floor}.\n");
    if (!house.HasBeenInvited)
    {
        Show.Menu(new string[] { "Скасувати", "Вихід" });
        int choice = Show.Options(2);
    }
}

```

```

switch (choice)
{
    case 1:
        CancelRequest(house, houseRequest);
        Console.WriteLine("\nЗапит скасовано");
        Thread.Sleep(2000);
        return;
    case 2:
        return;
}
}
}
}

```

```

class Show

```

```

{
    public static void Menu(string[] options)
    {

        for (int i = 0; i < options.Length; i++)
        {
            Console.WriteLine($"{i + 1}. {options[i]}");
        }
    }
}

```

```

    public static int Options(int option)
    {
        int choice;
        Console.WriteLine("");
    }
}

```

```
while (!int.TryParse(Console.ReadLine(), out choice) || choice < 1 || choice
> option)
{
    if (choice == 0) return 0;

    Console.WriteLine("Команда не вірна");
    Console.WriteLine("");
}
return choice;
}
}
```