Черкаський національний університет імені Богдана Хмельницького

Кафедра програмного забезпечення автоматизованих

систем

КУРСОВА РОБОТА

з дисципліни «Програмування та алгоритмічні мови»

НА ТЕМУ «Калькулятор витратних матеріалів для ремонту в різних видах приміщень»

Студента 2 курсу, групи КС-212

Галузі знань 12«Інформаційні технології»

Спеціальність 121«Інженерія програмного забезпечення»

Гончар А.С.

Керівник: викладач

Мисник Б.В.

Національна шкала:

Кількість балів: Оцінка: ECTS

Члени комісії

(підпис)	(прізвище та ініціали)
(підпис)	(прізвище та ініціали)
(підпис)	(прізвище та ініціали)

м. Черкаси – 2022 рік

Зміст

Вступ

Ця робота присвячена калькулятору витратних матеріалів для ремонту в різних видах приміщень. Ця тема в наш час є досить актуальною. За допомогою цієї програми можна розрахувати скільки матеріалів і яка сума потрібна для нашого міні-проекта.

Актуальність теми

Раніше будь-якій людині, щоб розрахувати кількість витрачених матеріалів потрібно було мати зошит і мати якісь знання з математики. Для цього використовувався зошит, ручка, лінійка і знання формули площі. Спочатку виміряли розмір сторін кімнати і тільки після цього підставивши в формулу визначали площу кімнати. Після обрахунку площі треба було піти в магазин і вибрати матеріал для того, щоб побачити скільки літрів або метрів цього матеріалу потрібно. Після цього люди повертались до дому і розраховували скільки всього матеріалу потрібно. І знову йшли в магазин щоб купити цей матеріал. І на це використовується дуже багато дій. Але в наш час можна скоротити ці дії в рази. Для цього як і раніше треба було визначити розмір кімнати. Але зараз не треба знати якісь формули, можна використати нашу програму для визначення площі. Після визначення площі ми вибераємо матеріали і програма сама вираховує скільки цього матеріалу вам потрібно. Ви можете спитати як визначити ціни матеріалів? Для цього вже не потрібно ходити в магазин. Просто відкриваєте будь-який електроний сайт в вашому телефоні і в ньому дивитесь на велику кількість різних матералів і цін.

Мета роботи

Метою роботи є реалізація програми для розрахунку витратних матеріалів для ремонту в різних видах приміщень. Головною цілью даної програми є розрахунок кількості використаних матеріалів та обрахування цін за використані матеріали.

Основні завдання курсової роботи

- Реалізувати алгоритм для вищитування витрачених матеріалів;
- Реалізувати алгоритм для вищитування цін за витрачені матеріали;
- Побудова блок-схеми для цього алгоритму;
- Реалізувати цей алгоритм;
- Зробити висновок.

Огляд алгоритмів та методи реалізації програми

1.1 Принцип роботи програми

Дана програма є не важкою, але слід її виконувати послідовно. розпишемо програму по діям.

1. Вищитуємо площу кімнати.

2. Вибираємо тип кімнати і матеріали для ремонту.

3. Вибираємо тип ремонту на вибір дається (стіна, стеля і підлога)

4. В цій дії повторюються усі 3 дії до того моменту коли користувач перейде до кінцевої дії обрахунок результатів.

5. В останій дії обраховується кількість матеріалів для даного типу ремонту і обраховується кінцева сума.

1.2 Метод знаходження площі

Принцип роботи даного алгоритму є не дуже важкий. У стандартних приміщень кімнати мають форму прямокутників. Отже за основу усіх кімнат візьмемо прямоктуник, приклад рис. 1.1. Для визначення площі нам потрібно знати дві сторони, нехай вони будуть мати назву A і B. Спершу вводимо значення A і значення B. Значення A і B (обов’язково вводяться в сантиметрах). Після вводу двух значень, підставляємо їх в формулу яка описана нище. Після обрахунку виводиться значення площі (S) в квадратних сантиметрах і можемо розуміти наскільки велика кімната. Після цього можна цього переходити до наступної дії.

$$S = A \cdot B$$

де S – площа [квадратні одиниці вимірювання];

A – ширина [звичайні одиниці вимірювання];

B –довжина [звичайні одиниці вимірювання].

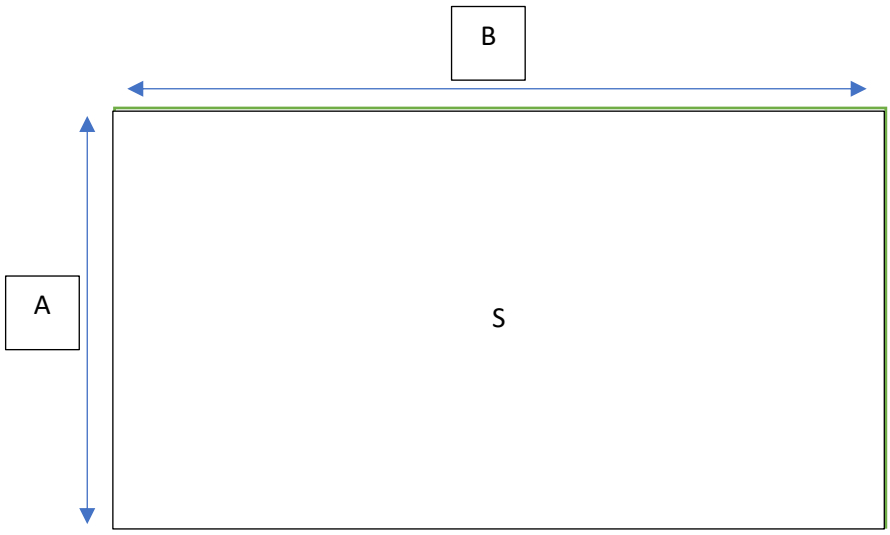

Рис. 1.1 Зображення кімнати з гори

1.3 Метод для розрахунку робіт

Коли параметри площі відомі слід обрати тип кімнати. На вибір дається три типи кімнат. Перший це кухня, другий це ванна, третій це звичайна кімната. Спершу вибирається вид ремонту. Тобто на вибір дається три типи ремонту(стеля, підлога і стіни).

При виборі стелі виповняється лише одна її функція – покраска, приклад стелі зображено на рис. 1.1. В цьому виду ремонту нам треба звернутись до площі і вибрати середній росхід кілограм на метр квадратний.

При виборі підлоги можна використати декілька видів робіт, приклад підлоги зображено на рис. 1.1. В цьому виду ремонту нам треба звернутись до площі і так визначити скільки потрібно матеріалів для підлоги.

В виборі стіни треба ввести в нашу програму ще одне значення D. Після того як в програму ввели нове значення, оглядаємо рис. 1.2. На цьому рисунку зображено 2 площі (2 стіни). Для обрхунку площі двух використаємо формулу яка подана нище. Залишилось обрахувати кількость матеріалів для цього звертаємося до площі 4 стін і за допомого.

$$S = 2 \cdot (A \cdot D) + 2 \cdot (B \cdot D)$$

де S – площа 4 стін;

A – ширина;

B – довжина;

D – висота [звичайні одиниці вимірювання].

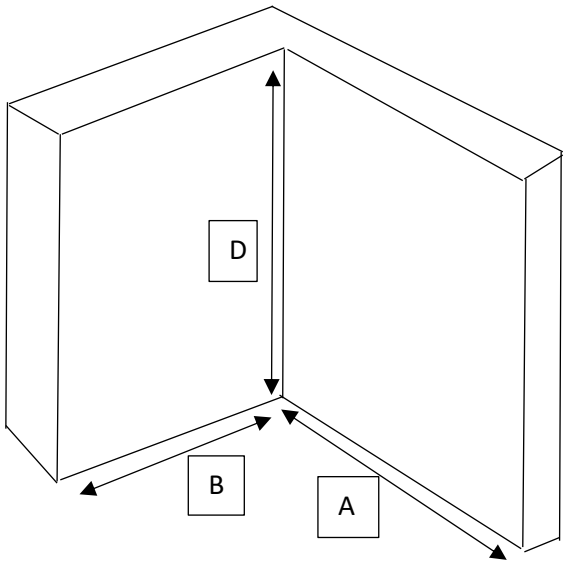

Рис. 1.2 Зображення стін

Якщо людина не вибирає один із 3 типів ремонту, то вище згадані алгоритми не використовуються.

1.4 Метод для розрахунку цін

Після вибору кімнат для ремонту і обрахування загальной кількості матеріалів, наша програма переходить в наступний етап. Цей етап називається кошторис, прикладом його є таблиця згадана нище. Головна суть цього алгоритму розрахувати правильну кількість матеріалів і ціну на вибрані типи ремонту. Операючись на данний вид розрахунку можна закінчувати перерахування принципів роботи моєї програми.

Номер	Найменування	од. вимірювання	Ціна за м2	Вартість (грн.)
Стеля (вид роботи)				
1	Зробити ремонт стелі в кухні (краска)	10 м2	10 грн.	100
2	Зробити ремонт стелі в ваній (краска)	11 м2	11 грн.	121
Пілога (вид роботи)				
3	Зробити ремонт підлога в кухні (плитка)	10 м2	15 грн.	150
Стіни (вид роботи)				
4	Зробити ремонт стіни в кімнаті (шпалери)	10 м2	10 грн.	100
5	Зробити ремонт стіни в кухні (шпалери)	10 м2	11 грн.	121

Підсумок першого розділу

В першому розділі розглянуто роботу „Калькулятора витратних матеріалів для ремонту в різних видах приміщень”, метод для обрахунки площі в якому ввели формулу площі та алгоритм для розрахунку робіт в якому було введено нове значення висоти. І на кінець показали метод для обрахунку цін за допомогою кошторису в форматі гугл-таблиць.

Проектування програмного продукту

2.1 Алгоритми і методи для розрахунку робіт

2.1.1 Метод для обрахунку цілої частини одиниць певного матеріалу

Для використання цього методу треба створити новий клас Class Renovation. Задачою даного класу буде обрахування, скільки треба витрати одиниць певного матеріалу, щоб покрити ним певну поверхню (стіни, стеля, підлога). Даний клас мусить обрахувати кількість матеріалів, якщо число не ціле він мусить округлити до цілого значення. Округляється тільки до певного типу матеріалу(фарба), якщо це фарба то округляємо, якщо ні то залишаємо без змін. (рис. 2.1)

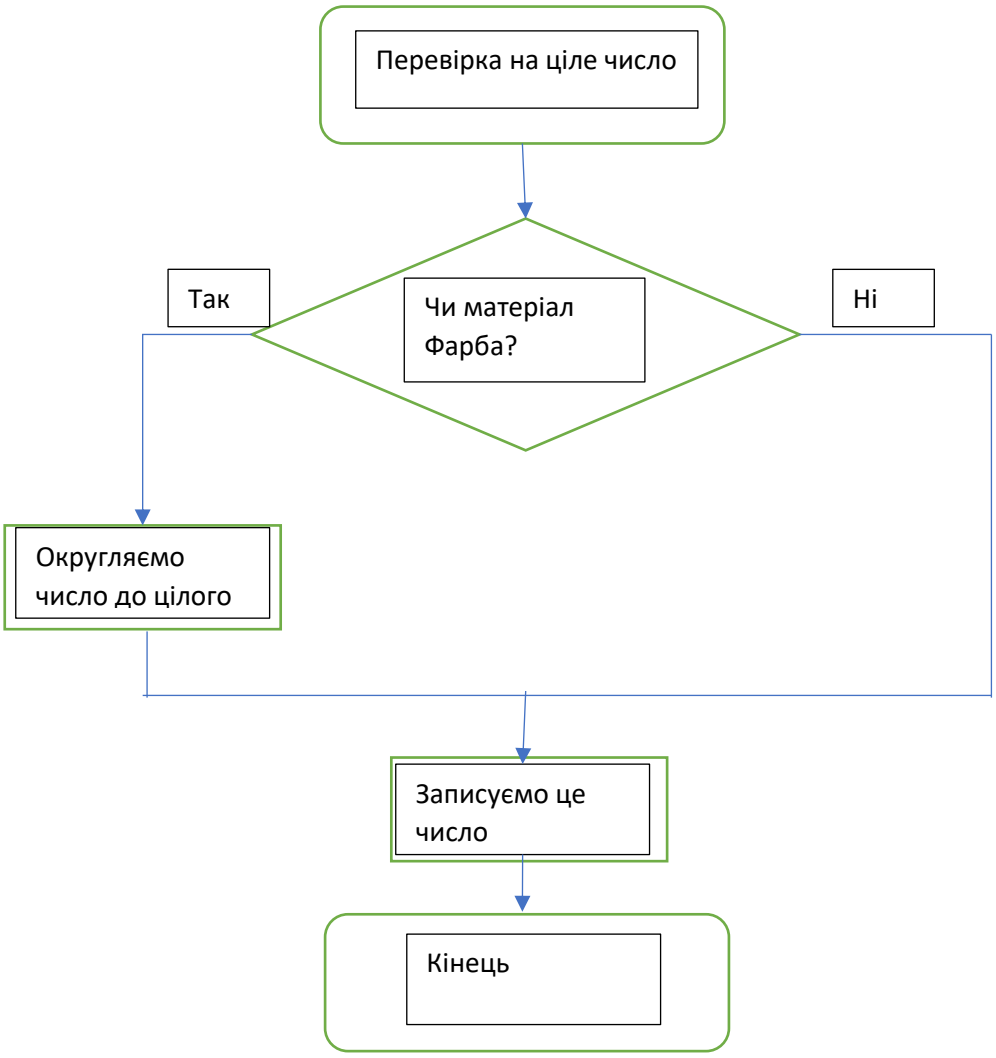

Рис. 2.1 – Блок-схема для обрахунку цілої частини

2.1.2 Метод для зберігання інформації про поверхню

Для використання цього методу треба створити новий клас Class Room. Задачою даного класу буде збереження інформацій про поверхню однієї кімнати. Даний клас мусить мати доступ до будь якого виду поверхонь, мати коротку іформацію про кімнати(висота, ширина, довжина). І мусить повертати масив поверхонь, попередньо визначивши параметри (стіна, стеля і підлогу) для кожної окремо.

2.1.3 Метод для обрахування площі

Для використання цього методу треба створити новий клас Class Surface. Задачою даного класу буде зберігання інформації про одну поверхню, та розраховування її площі.

2.1.4 Метод для перерахування та зберігання поверхонь

Для використання цього методу треба створити Enum SurfaceType і SurfaceOrder. Enum SurfaceType потрібен для перерахування поверхонь (Wall, Roof, Floor). А ось Enum SurfaceOrder потрібен для перерахування порядку зберігання поверхонь (це необхідно, бо поверхні можна записувати в будь-якому порядку, таким чином ми точно знаємо де конкретно в пам’яті лежить підлога, де стеля і стіна).

2.1.5 Алгоритм розміщення матеріалу на певній поверхні

Для використання цього алгоритму треба звернуться до классу Class Material. В цьому алгоритмі треба перевірити чи може бути тут цей матеріал. Наприклад лінолеум не може бути на стелі або шпалери не можуть бути на полу. (рис. 2.2)

Назва типу матеріалу	Чи можна покрити стіни?	Чи можна покрити стелю?	Чи можна покрити підлогу
Wallpaper	Так	Ні	Ні
Linoleum	Ні	Ні	Так
Tile	Так	Ні	Так
Paint	Так	Так	Ні

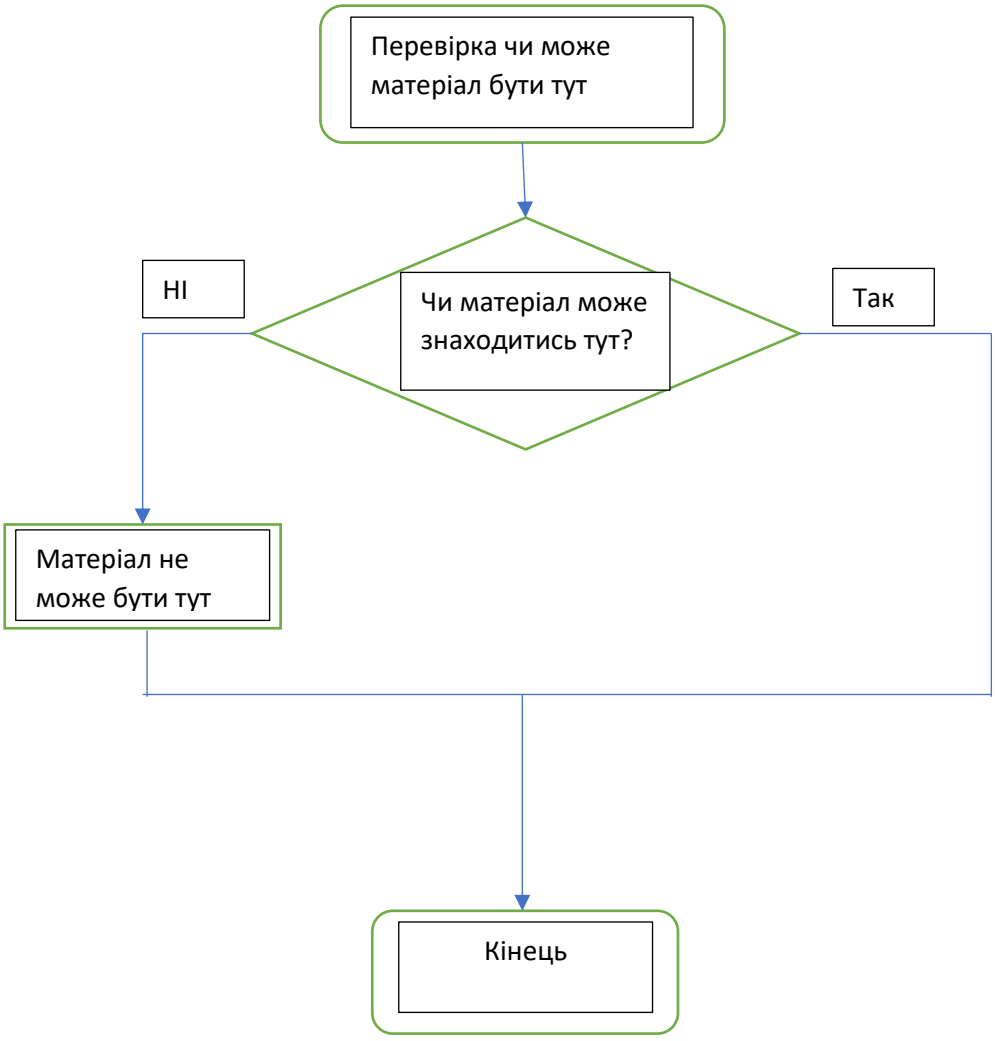

Рис. 2.2 – Блок-схема для перевірки розміщення матеріалу на певній поверхні

2.2 Метод для розрахунку цін

Для використання цього методу теба звернутись до класу Material. В цьому класі реалізованно метод вартостість матеріалів за її кількість. Для того щоб визначити суму треба треба помножити кількість одиниць квадратних на ціну одного квадратного метру матеріалу.

2.3 Реалізація різних класів (матеріалів)

Для шпалер буде створено новий клас Class Wallpaper. Wallpaper це дочірній клас Material, який буде описувати шпалери.

Для лінолеума буде створено новий клас Class Linoleum. Linoleum це дочірній клас Material, який буде описувати покриття підлоги (лінолеум).

Для плитки буде створено новий клас Class Tile. Tile це дочірній клас Material, який буде описвати плитки.

Для фарби буде створено новий клас Class Paint. Paint це дочірній клас Material, який буде описувати фарбу.

Підсумок другого розділу

В другому розділі розглянуто докладно роботу „Калькулятора витратних матеріалів для ремонту в різних видах приміщень”. Реалізовано алгоритми і методи для розрахунку робіт.

Метод перший - обрахунок цілої частини одиниць певного матеріалу, в якому докладно описали в яких випаднах число повино округлятись а в яких ні.

Метод другий - метод для обрахування площі. Головною дією цього метода є зберігання інформації і знаходження площі.

Алгоритм третій - алгоритм розміщення матеріалу на певній поверхні. Головною дією цього алгоритма є перевірка чи може бути тут цей матеріал.

Метод четвертий - метод для розрахунку ціни. Головною дією цього метода розрахунок ціни за матеріали.

І на кінець показали реалізацію різних класів матеріалів (шпалери, лінолеум, плитка і фарба). Було побудовано 2 блок-схеми. Перша для реалізації обрахунку цілої частини. Друга для перевірки розміщення матеріалу на певній поверхні

Реалізація програмного продукту

3.1. Вибір мови програмування

Для даної програми використовувалась мова програмування C#. C# це об’єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. C# був розроблений Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research. Мова підтримує поліморфізм, перевантаження операторів, атрибути, властивості і коментарі у форматі XML. Дуже багато мова перейняла від C++, Delphi, Модула і Smalltalk , переглянувши проблеми раніше переглянутих мов вона виключає різні проблеми наприклад множинне спадкування класів (на відміну від C++).[1]

3.2. Вибір фреймворку та архітектура програми

Для написання використовувалась IDE «Rider». Компанія JetBrains, представила на початку 2016 року нове інтегроване середовище розробки на мові C# під назвою Project Rider. На відміну від подібної Microsoft Visual Studio за функціональності, це середовище є крос-платформним - воно працює не тільки під Windows, але і під OS X і Linux.

Нове інтегроване середовище розробки (Integrated Development Environment) JetBrains дозволяє створювати програми для Windows, веб-додатки та мобільні програми, як і Microsoft Visual Studio. Але, на відміну від Visual Studio, Rider є крос-платформним середовищем, його можна використовувати під Windows, OS X і Linux (тоді як Visual Studio — лише під Windows). Хоча в JetBrains визнають, що версія для Linux поки що не така як для Windows або OS X. [2]

Головною перевагою фреймворка «Avalonia» є те що він використовує для проектування патерн MVVM. Він використовувся для роз’єднання (frontend) від (backend). В данному патерні Model В даному патерні Model усі об’єкти які

будуть якось відображатися або змінюватися. ViewModel в ньому описується логіка використання моделей (об’єктів) та їх модифікації залежно від дій у View частині. [3]

Використовуючи даний патерн, проект виглядає наступним чином (рис. 3.1)

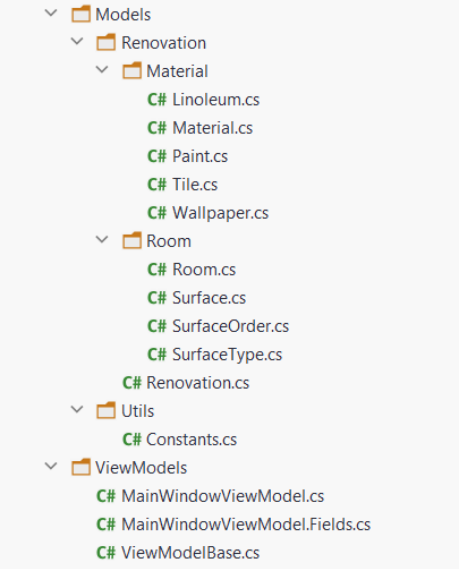

Рис. 3.1. Файлове представлення проекта.

3.3. Проектування інтерфейсу

На даній формі відображено наступні елементи, Name - ім’я кімнати, Height - висота кімнати (в см.), Width - ширина кімнати (в см.), Lenght - довжина кімнати (в см.). Add new room - Кнопка для додавання кімнат(на вибір дається три типи Living room, Kitchen, Bathroom). Delete - видалення кімнати, update room properties - оновлення даних (кімнати, висота, ширина і довжина), update material properties - оновлення (панелі матеріалів). Wall(1-4), Floor, Roof - кнопки для виду ремонту (1,2,3,4-та стіна, підлога, стеля).Add this room to sum - кнопка для вищитування суми і Clear - видалення цієї суми. Так повинен виглядати проект рис. 3.2. При першому відкритті базові дані вже будуть вписані. В цей список входять ім’я кімнат, висота, ширина і довжина. Назва матеріалів, висота , ширина і ціна за метр квадратний. Якщо людина хоче додати кімнату, ій треба вибрати вид кімнати і нажати на кнопку (Add new room), потім перейти в цю кімнату і

вписати параметри кімнати. Після вибору кімнати вибрати матеріал для роботи на вибір їх 4 виду. Після вписати дані про цей матеріал і перевірити чи підходить він для кімнати, якщо ні то повино висвітись що не підходить рис. 3.3. Після того як вибрали усі матеріали, переходимо до кінцевого обрахування Add this room to sum. Якщо все правильно вибрати програма повина виглядати так Рисунок 3.4. Скільки було використано матеріалів і кінцева сума. Якщо щось було не правильно можна видалити кінцеву суму і кількість матеріалів на кнопку Clear.

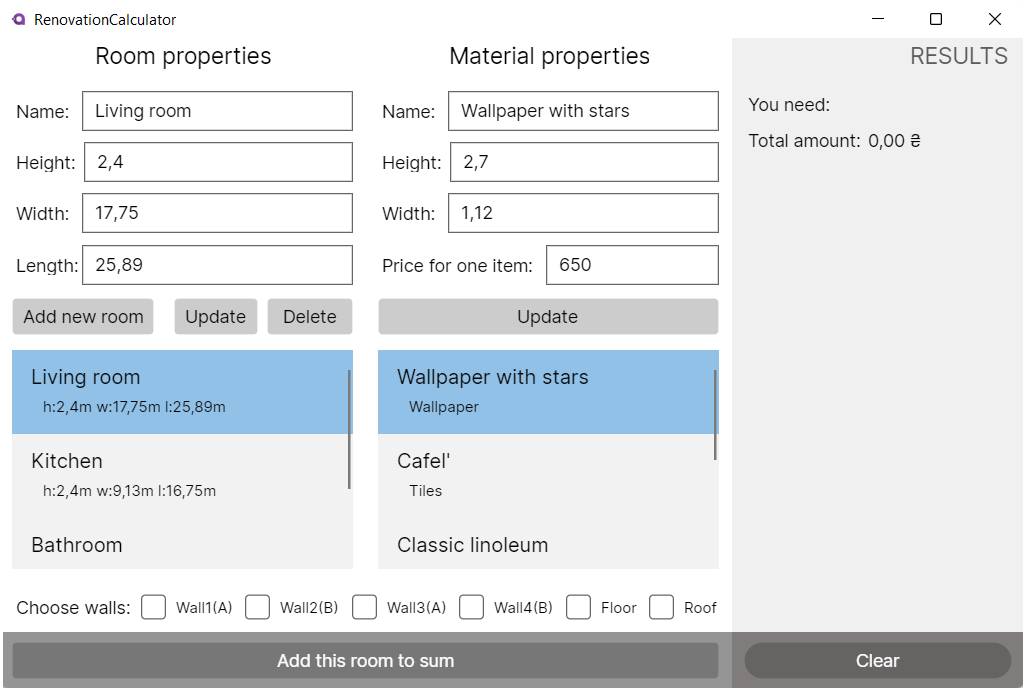

Рис. 3.2. Вигляд проєкта.

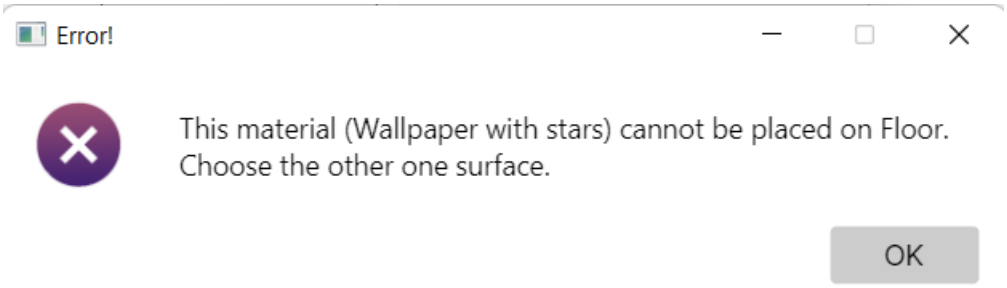

Рис. 3.3. Помилка при виборі матеріалу.

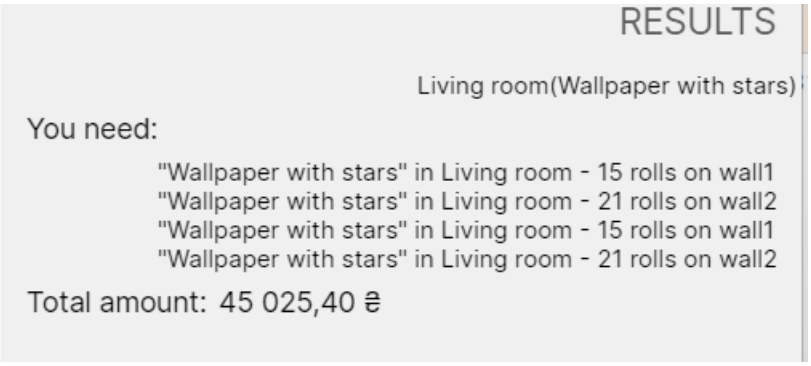

Рис. 3.4.Кінцеве обрахування ремонту.

3.4 Перевірка на необрані дані.

Просимо користувача ввести якісь дані про кімнату, якщо він не вводить програма просить ще раз ввести якісь дані в програму. (рис. 3.5.)

```
// Додаємо кімнату та матеріал, яким будемо покривати вказані поверхні (поверхні отримуємо через checkBoxData)
// (клік по кнопці "Add this room to sum")
1 usage
public void AddRenovationInCollection(string checkBoxData)
{
    // Якщо не вибрана кімната чи матеріал, то просимо користувача їх обрати
    if (SelectedRoom == null || SelectedMaterial == null)
    {
        ShowErrorDialog( message: "Select a room and a material.");
        return;
    }
```

Рис 3.5 перевірка якщо не обрана кімната

3.5 Перевірка чи може бути тут обраний матеріал.

Коли була обрана кімната перевірити чи можна уживати матеріал для типу ремонту. Наприклад плитка для стелі. (рис 3.6)

```
// Обраховуємо використані матеріали та оновлюємо загальні витрати.
// Якщо певний матеріал не можна використовувати на певній поверхні,
// то просимо користувача змінити поверхню
try
{
    for (int i = 0; i < wallFlags.Length; i++)
    {
        if (wallFlags[i] && (i == 0 || i == 2))
            UpdateRenovationReport(renovation, SurfaceOrder.Walls1);
        else if (wallFlags[i] && (i == 1 || i == 3))
            UpdateRenovationReport(renovation, SurfaceOrder.Walls2);
        else if (wallFlags[i] && i == 4)
            UpdateRenovationReport(renovation, SurfaceOrder.Floor);
        else if (wallFlags[i] && i == 5)
            UpdateRenovationReport(renovation, SurfaceOrder.Roof);

        if (wallFlags[i] == false)
            falseCounter++;
    }
}
catch (Exception e)
```

Рис 3.6 перевірка чи може бути обраний матеріал

3.6 Перевірка чи обраний вид ремонту.

Спершу перевіряємо чи був обраний ремонт поверхні якщо ні то просимо вибрати поверхню. (рис 3.7)

```
// Якщо не було обрано жодної поверхні, то просимо користувача
// обрати хоча б одну поверхню
if (falseCounter == wallFlags.Length)
{
    ShowErrorDialog( message: "Select at least one surface.");
    return;
}
```

Рис 3.7 перевірка на обрану певерхню

3.7 Перевірка на помилку

Якщо в програмі є якась помилка, то виводиться error.

```
private void ShowErrorDialog(string message)
{
    var dialogBox = MessageBox.Avalonia.MessageBoxManager
        .GetMessageBoxStandardWindow( title: $"Error!", text: message,
            MessageBoxAvaloniaEnums.ButtonEnum.Ok,
            MessageBoxAvaloniaEnums.Icon.Error);
    dialogBox.Show();
}
```

Рис 3.8 перевірка будь-яку помилку

Підсумок третього розділу

В третьому розділі реалізовано роботу „Калькулятора витратних матеріалів для ремонту в різних видах приміщень. Було обрано мову програмування, середовище, фреймворк. Було створено інтерфейс, його повністю було росписано.

Висновок

В результаті даної курсової роботи була розроблена програма Калькулятор витратних матеріалів для ремонту в різних видах приміщень. Перед початком розробки програми був розглянутий принцип роботи цього калькулятора.

Далі було оглянуто проектування програмного продукту. В ньому було побудовано дві блок-схеми. Ще в цьому етапі було розглянуто максимально усі алгоритми і методи які були реалізовані в моїй програмі. В подальшому при написані програми мені не так було важко писати програму, коли усе було розписано.

В наступному кроці треба було обрати фреймворк для проектування моєї програми, моїм вибором стала «Avalonia», через її багатоплатформність. Архітектурним патерном було обрано MVVM для того щоб відокремити візуальну частину від основної логіки програми.

Якщо далі розвивати програму, то пакет-доповнення для фреймоврку «Avalonia» є найкращим вибором, через те що людям які захочуть розвинути дану програму буде краще орієнтуватись в програмі, вони зможуть мати меньше проблем з доповненям до цієї програми. Висновком цього буде спрощення створень нових методів та алгоритмів в цій програмі. Цю програму можна зробити ще краще, добавивши вікна, двері, зробити кімнату не прямокутною. Отже, ця програма підтверджує що в цьому світі ще не все досконале, і навіть такі незначні додавання можуть зробити світ краше.

Список використаних джерел

1. http://y66819tz.beget.tech/c-sharp/ Інформація про C#.

2. https://www.tadviser/index.php опис Rider.

3. https://metanit.com/sharp/wpf/22.1.php опис Паттерна MVVM.

4. https://www.meter.com.ua/uk/calculate.html як повинен виглядати калькулятор