

ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ З ОЦІНКИ НЕРУХОМОСТІ»

Студента 4 курсу, 10 групи
спеціальності
122 «Комп'ютерні науки»

Красюка Максима
Сергійовича

підпис студента

Науковий керівник
Кандидат технічних наук, доцент

Томашевська
Тетяна
Володимирівна

підпис керівника

Гарант освітньої програми
кандидат технічних наук, професор

Демідов Павло
Георгійович

підпис керівника

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____

Затверджую
Пурський О.І.
« 12» грудня 2022 р.

Завдання на випускн кваліфікаційну роботу студенту

Красюк Максим Сергійович
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)
«Розробка інформаційної системи з оцінки нерухомості»
Затверджена наказом ректора від «09» грудня 2022 р. № 3332
2. Строк здачі студентом закінченої роботи 30 травня 2023 року
3. Цільова установка та вихідні дані до роботи
Мета роботи: є розробка програмного забезпечення визначення оцінки нерухомості підприємства та аналізі роботи системи.
Об'єкт дослідження: процеси розробки інформаційної системи з оцінювання нерухомості.
Предмет дослідження: засоби розробки інформаційної системи з оцінювання нерухомості.
4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

1	Томашевська Т.В.	15.12.2022 р.	15.12.2022 р.
2	Томашевська Т.В.	15.12.2022 р.	15.12.2022 р.
3	Томашевська Т.В.	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ОСНОВНІ ПІДХОДИ ДО АВТОМАТИЗОВАНОГО УПРАВЛІННЯ РИЗИКАМИ ПІДПРИЄМСТВА З ОЦІНКИ НЕРУХОМОСТІ

1.1. Сутність, поняття, загальна проблематика управління оцінкою нерухомості.

1.2. Основні методи оцінки нерухомості.

1.3. Програмні засоби оцінки нерухомості.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДПРИЄМСТВА ОЦІНКИ НЕРУХОМОСТІ

2.1. Постановка задачі.

2.2. Система вхідних показників визначення оцінки нерухомості.

2.3. Моделювання визначення фінансової стійкості нерухомості.

2.4. Обґрунтування вибору мови програмування та технологій зберігання даних

2.5. Проект системи

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДПРИЄМСТВА З ОЦІНКИ НЕРУХОМОСТІ

3.1. Архітектура програмного забезпечення системи

3.2. Тестування роботи програми

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	04.10.2022	04.10.2022
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	15.12.2022	15.12.2022
3	<i>Вступ</i>	03.02.2023	03.02.2023
4	<i>РОЗДІЛ 1. Теоретичні аспекти механізмів оцінювання рівня соціально-економічного розвитку</i>	28.02.2023	28.02.2023
5	<i>РОЗДІЛ 2. Математична модель визначення інтегрального показника рівня соціально-економічного розвитку</i>	06.04.2023	06.04.2023
6	<i>РОЗДІЛ 3. Автоматизована система оцінювання показників соціально-економічного розвитку регіонів України</i>	12.05.2023	12.05.2023
7	<i>Висновки</i>	15.05.2023	15.05.2023
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	30.05.2023	30.05.2023
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	31.05.2023 -01.06.2023	31.05.2023 -01.06.2023
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	02.06.2023	02.06.2023
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	05.06.2023	05.06.2023
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2022 р.

Керівник випускної кваліфікаційної роботи (проекту)

Томашевська Т.В.

(прізвище, ініціали, підпис)

Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

Завдання прийняв студент-дипломник

Красюк М. С.

(прізвище, ініціали, підпис)

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ОСНОВНІ ПІДХОДИ ДО АВТОМАТИЗОВАНОГО УПРАВЛІННЯ РИЗИКАМИ ПІДПРИЄМСТВА З ОЦІНКИ НЕРУХОМОСТІ	10
1.1 Сутність, поняття, загальна проблематика управління оцінкою нерухомості.....	10
1.2 Основні методи оцінки нерухомості.....	12
1.3 Програмні засоби оцінки нерухомості.....	16
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДПРИЄМСТВА ОЦІНКИ НЕРУХОМОСТІ	23
2.1. Постановка задачі.....	23
2.2. Система вхідних показників визначення оцінки нерухомості	24
2.3. Моделювання визначення фінансової стійкості нерухомості.....	26
2.4. Обґрунтування вибору мови програмування та технологій зберігання даних.....	31
2.5. Проект системи.....	33
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДПРИЄМСТВА З ОЦІНКИ НЕРУХОМОСТІ	37
3.1. Архітектура програмного забезпечення системи.....	37
3.2 Тестування роботи програми	43
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	48
ДОДАТОК	51

ВСТУП

Розробка ефективних інформаційних систем є актуальною задачею в сучасному світі, особливо в контексті оцінки нерухомості. Значущі зміни на ринку нерухомості та постійне зростання інтересу до покупки, продажу або оренди нерухомих об'єктів створюють потребу у зручних та надійних інструментах для визначення їх ринкової вартості.

У цьому контексті інформаційна система з оцінки нерухомості виступає як ключовий елемент, що допомагає фахівцям з оцінки встановлювати об'єктивну вартість нерухомих об'єктів. Основною метою цієї інформаційної системи є надання швидкого, точного та надійного засобу для здійснення оцінки нерухомості, що враховує різноманітні фактори, такі як розмір, стан, розташування об'єкта, ринкові тенденції та інші.

Одним із ключових завдань інформаційної системи з оцінки нерухомості є збір, аналіз та обробка різноманітних даних, пов'язаних з нерухомістю. Вона використовує різні джерела, включаючи публічні бази даних, агентства нерухомості та організації з реєстрації прав власності, для отримання необхідної інформації про характеристики об'єктів, географічні дані, ринкові тенденції та інші фактори, що впливають на їх вартість.

Крім того, інформаційна система забезпечує зручне зберігання та оновлення даних про нерухомість, що дозволяє відстежувати зміни їх вартості в часі та забезпечує доступ до актуальної інформації для користувачів системи. Також система забезпечує важливий аспект безпеки, забезпечуючи захист даних та конфіденційність інформації про нерухомість.

Використання інформаційних комп'ютерних систем дозволяє автоматизувати різні аспекти діяльності підприємства, що призводить до покращення управління і підвищення ефективності. Наприклад, такі системи можуть бути використані для оптимізації процесів планування і аналізу, контролю і вимірювання продуктивності, а також для підвищення якості прийняття рішень. Завдяки цим системам, підприємець може швидко

отримувати необхідну інформацію та аналізувати її на різних гаджетах, таких як телефон, планшет або ноутбук.

Зокрема, важливим аспектом використання інформаційних систем є їх застосування для оцінки нерухомості. За допомогою комп'ютера та спеціалізованого програмного забезпечення можна значно скоротити час, потрібний для отримання необхідної інформації про проект. Крім того, ці системи дозволяють зберігати і оновлювати дані про нерухомість, що спрощує їх подальше використання і аналіз.

Також важливо відзначити, що інформаційні системи забезпечують зв'язок між підприємствами та їх партнерами і клієнтами в мережевому режимі. Це дозволяє підприємствам швидко обмінюватися інформацією, координувати спільні проекти та покращувати комунікацію зі своїми зацікавленими сторонами.

Мета даного дослідження – є розробка програмного забезпечення визначення оцінки нерухомості підприємства та аналізі роботи системи.

Для досягнення мети були поставлені такі **завдання**:

1. Аналіз процесу оцінювання нерухомості та визначення його особливостей в сучасних умовах
2. Визначення параметрів оцінювання нерухомості
3. Проектування інформаційної системи оцінювання нерухомості
4. Розробка інформаційної системи оцінювання нерухомості та її тестування

Об'єктом дослідження є процеси розробки інформаційної системи з оцінювання нерухомості.

Предметом дослідження є засоби розробки інформаційної системи з оцінювання нерухомості.

Методи дослідження: Теоретичною основою дослідження є загальнонауковий аналітичний метод, а також системний підхід і праці провідних вчених з проблем оцінювання нерухомості. Інформаційну базу

дослідження становлять статистичні дані агентств нерухомості. Для практичного вирішення поставлених задач використовувалися такі методи:

- загальнонауковий аналітичний метод;
- методи математичного моделювання для оцінки нерухомості;
- методи алгоритмічного програмування, для створення інформаційної системи оцінювання нерухомості.

Практичне значення. Отримані результати, можуть бути використані агентствами нерухомості в своїй практичній діяльності, що значно підвищить ефективність їх роботи.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел із 26 найменувань, додатків і містить 49 сторінки основного тексту, 5 рисунків і 4 таблиці.

РОЗДІЛ 1.

ОСНОВНІ ПІДХОДИ ДО АВТОМАТИЗОВАНОГО УПРАВЛІННЯ РИЗИКАМИ ПІДПРИЄМСТВА З ОЦІНКИ НЕРУХОМОСТІ

1.1. Сутність, поняття, загальна проблематика управління оцінкою нерухомості.

Управління ризиками включає аналіз, оцінку, контроль та зменшення ризиків, які можуть вплинути на досягнення мети підприємства. Ризики можуть походити з різних джерел, таких як фінансові, економічні, правові, технологічні, природні або соціальні фактори. Підприємство має виявляти, аналізувати та управляти цими ризиками, щоб забезпечити стійкість своєї діяльності та мінімізувати втрати.

Основні кроки управління ризиками включають:

- Ідентифікація ризиків: Цей крок передбачає виявлення потенційних ризиків, які можуть вплинути на підприємство. Це може включати оцінку зовнішнього середовища, внутрішніх процесів, фінансового стану, конкурентної ситуації тощо.
- Оцінка ризиків: Після ідентифікації ризиків їх потрібно оцінити з точки зору ймовірності виникнення та впливу на підприємство. Це дозволяє визначити пріоритети управління ризиками та визначити необхідні заходи для кожного ризику.
- Розробка стратегій управління ризиками: На основі оцінки ризиків підприємство розробляє стратегії та плани дій для кожного ризику. Ці стратегії можуть включати уникнення ризику, зменшення ризику, передачу ризику за допомогою страхування або фінансових інструментів, а також прийняття ризику в певних випадках.
- Реалізація стратегій та моніторинг ризиків: Після розробки стратегій управління ризиками їх необхідно реалізувати і встановити механізми

моніторингу ризиків. Це включає в себе встановлення системи контролю, звітності та відповідальності за управління ризиками.

- Оновлення та удосконалення процесів: Управління ризиками є постійним процесом, і воно повинно адаптуватися до змін у внутрішньому та зовнішньому середовищі підприємства. Підприємство повинно оновлювати свої стратегії, процедури та методи управління ризиками для забезпечення ефективного контролю та зменшення ризиків.
- Виявлення ризиків: Виявлення ризиків - це процес визначення, аналізу та оцінки потенційних ризиків, які можуть виникнути у майбутньому. Це може включати проведення аналізу SWOT (аналіз сильних і слабких сторін підприємства, можливостей та загроз), аналіз статистичних даних, консультації з експертами тощо. Виявлення ризиків допомагає підприємству зрозуміти свої слабкі місця і прийняти заходи для їх зменшення.
- Категоризація ризиків: Ризики можна категоризувати за різними ознаками, наприклад за джерелом походження, впливом на підприємство, ступенем невизначеності тощо. Категоризація ризиків допомагає підприємству краще розуміти їх характеристики та розподілити ресурси для ефективного управління ними.
- Управління кризовими ситуаціями: Управління ризиками також включає готовність до управління кризовими ситуаціями. Кризові ситуації можуть виникати з-за непередбачених обставин, таких як природні катастрофи, економічні кризи, технологічні аварії тощо. Підприємство повинно мати плани негайних дій, комунікаційні стратегії та процедури для керування кризовими ситуаціями та забезпечення відновлення діяльності після них.
- Культура ризиків: Успішне управління ризиками вимагає створення культури ризиків у всьому підприємстві. Це означає, що всі працівники повинні бути освічені щодо ризиків, активно виявляти і повідомляти

про потенційні ризики, а також брати участь у впровадженні стратегій управління ризиками. Культура ризиків допомагає підприємству бути гнучким, прогнозованим і реагувати на зміни в економічному та бізнес-середовищі.

- Використання технологій: Сучасні технології, такі як аналітика даних, штучний інтелект, машинне навчання та автоматизація процесів, можуть значно полегшити управління ризиками. Вони допомагають збирати та аналізувати великі обсяги даних, передбачати ризики, виявляти аномалії та автоматизувати процеси контролю ризиків.

Управління ризиками є важливою складовою успішного управління підприємством, дозволяючи зменшити негативний вплив ризиків на бізнес та забезпечити стійкість та стабільність його розвитку.

1.2. Основні методи оцінки нерухомості.

У сучасних умовах ринку важливо мати можливість передбачити фінансову ситуацію підприємства в майбутньому. В економічній науці з'явилися різноманітні розробки і методи аналізу та прогнозу діяльності підприємств. Особливо цікаві є методи визначення ймовірності банкрутства та рівня надійності підприємств.

Необхідність таких розрахунків обумовлена тим, що в умовах вільного підприємництва часто виникають питання щодо прогнозу фінансової стійкості на довгострокову перспективу і оцінки ризику банкрутства підприємства в реальних економічних умовах.

Для розв'язання цих питань можна використовувати економіко-статистичні методи, які базуються на розробці моделей, що враховують різні фактори, що впливають на платоспроможність та фінансову стійкість підприємства.

Перевагами такої системи індикаторів можливого банкрутства є системний та комплексний підхід до аналізу, а недоліками - складність

прийняття рішення в умовах багатокритеріальної задачі, інформативність розрахованих показників та суб'єктивність прогнозних рішень.

На рис.1. представлена класифікація існуючих підходів для оцінки та прогнозування ймовірності банкрутства підприємств.

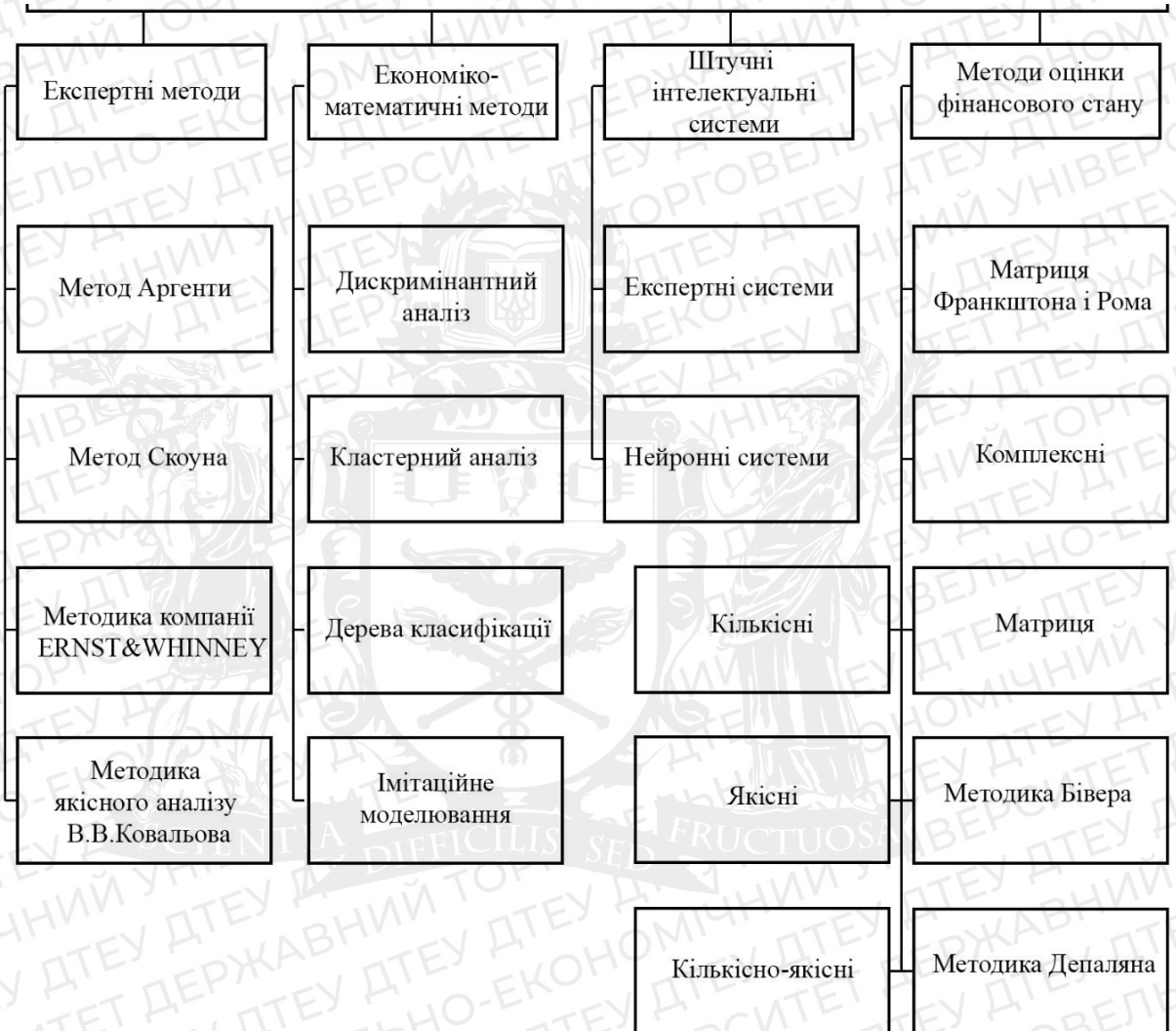


Рис. 1.1. Класифікація існуючих підходів до оцінки та прогнозування оцінки нерухомості

На рис.1 видно, що існує чотири основних підходи для оцінки й прогнозування ймовірності банкрутства підприємств з оцінки нерухомості:

1. Експертні методи.
2. Економіко-математичні методи.
3. Штучні інтелектуальні системи.
4. Методи оцінки фінансового стану.

В сучасних ринкових умовах велике значення має прогнозування фінансової ситуації в майбутньому. Економічна наука пропонує різні методи і підходи для аналізу та прогнозу діяльності підприємств з метою визначення ступеня їх надійності та ризику банкрутства.

Один із підходів базується на якісній оцінці ймовірності банкрутства підприємств. Відомими методиками цього підходу є метод Аргенті (А-рахунок), метод Скоуна, методика компанії ERNST&WHINNEY, методика якісного аналізу В.В. Ковальова. Вони засновані на оцінці різних якісних факторів, що впливають на ймовірність банкрутства підприємств.

Другий підхід використовує економіко-математичні моделі для прогнозування ймовірності банкрутства. До таких методів належать дискримінантний аналіз, кластерний аналіз, метод дерев класифікації та імітаційне моделювання. Наприклад, кластерний аналіз дозволяє класифікувати підприємства залежно від їх фінансової стабільності та ризику банкрутства.

При використанні цих методів для прогнозування ймовірності банкрутства враховуються різні фінансові показники підприємства, аналізуються дані за декілька років, що дозволяє врахувати часовий фактор. Кожен з цих методів має свої переваги, такі як графічне подання результатів, легкість інтерпретації та ієрархічність обчислень.

Також імітаційне моделювання використовується для прогнозування ймовірності банкрутства на основі аналізу результатів. Цей підхід дозволяє оцінити ризик банкрутства підприємства у майбутньому. За допомогою цього методу менш фінансово стійкі підприємства мають більшу ймовірність збанкрутувати в найближчий період, ніж більш стійкі підприємства.

Існує кілька економіко-математичних методів, що застосовуються для прогнозування банкрутства підприємств і оцінки їх фінансового стану. Одним із найбільш поширених методів є дискримінантний аналіз, який базується на побудові дискримінантної функції. Деякі відомі методи, що використовуються в практиці, включають:

1. Модель оцінки ймовірності банкрутства на основі Z-критерію Е. Альтмана.
2. Модель оцінки фінансового стану Р. Ліса.
3. Метод рейтингового числа.
4. R-модель з оцінки нерухомості.
5. Прогнозна модель Дж. Таффлера.
6. Модель Д. Чессера.
7. Узагальнена модель з оцінки нерухомості.
8. Галузеві моделі з оцінки нерухомості.
9. Модель PAS-коефіцієнта.

Крім того, штучні інтелектуальні системи, такі як експертні системи та системи нейромережових обчислень, також використовуються для прогнозування ймовірності банкрутства підприємств. Експертні системи діагностичного характеру призначені для виявлення причин незадовільного стану підприємства та моделювання механізму мислення людини. Системи нейромережових обчислень базуються на штучних нейронних мережах і використовуються для діагностики банкрутства підприємств та прийняття фінансових рішень.

У сфері оцінки нерухомості підприємства також застосовуються різні методики. Деякі з них включають:

- Система показників У. Бівера.
- Метод "credit-men" Ж. Дєпаляна.
- Методика оцінки нерухомості підприємства І. Бланка.
- Методика, заснована на розрахунку коефіцієнта фінансування важколіквідних активів.
- Модель фінансової рівноваги підприємства Ж. Франшо та І. Романе.

Також використовуються системи коефіцієнтів, такі як коефіцієнт поточної ліквідності, коефіцієнт забезпеченості власними коштами та коефіцієнт відновлення (втрати) платоспроможності.

Незважаючи на широкий спектр методів, жодна з них не може претендувати на універсальність. Вибір методу залежить від конкретних умов і завдань аналізу, а також від досвіду та кваліфікації фахівців.

1.3. Програмні засоби оцінки нерухомості

Загальною метою програмних засобів оцінки нерухомості є надання користувачам об'єктивної оцінки вартості нерухомості на основі різних факторів. Ці засоби можуть використовувати різні підходи та методи, але основна ідея полягає в тому, щоб зібрати, аналізувати та інтерпретувати дані, що впливають на ціну нерухомості.

Ось декілька конкретних аспектів програмних засобів оцінки нерухомості:

1. Збір даних: Програмні засоби зазвичай отримують доступ до різних джерел даних, таких як публічні реєстри, бази даних про продажі нерухомості, географічні інформаційні системи та інші джерела. Вони збирають інформацію про ціни продажу схожих об'єктів, характеристики нерухомості, місцезнаходження, регіональні фактори та інші релевантні дані.

2. Аналіз даних: Після збору даних програмні засоби проводять аналіз, використовуючи різні методи та моделі оцінки. Це може включати порівняння з аналогічними об'єктами, використання статистичних моделей, аналіз ринкових тенденцій та інші методи.

3. Врахування факторів: Програмні засоби можуть враховувати широкий спектр факторів, що впливають на вартість нерухомості. Це можуть бути такі фактори, як розмір та площа нерухомості, розташування, близькість до зручностей (школи, магазини, громадський транспорт), стан будинку, рівень попиту на ринку, макроекономічні фактори та багато інших.

4. Виведення результатів: Після аналізу даних програмні засоби генерують орієнтовану вартість нерухомості або надають користувачеві рейтинг, оцінку або звіт. Цей результат може бути представлений у вигляді числової оцінки, діапазону вартості або графічного відображення.

Розглянемо програмний спосіб оцінки нерухомості на прикладі сайту evaluation.spfu.gov.ua (Фонд державного майна України)



Сайт <https://evaluation.spfu.gov.ua/> є офіційним веб-порталом Державного фонду майна України для проведення оцінки нерухомості. Цей сайт надає можливість здійснювати онлайн-оцінку різних типів об'єктів нерухомості, таких як квартири, будинки, земельні ділянки і комерційні приміщення.

1. Для початку оцінки нерухомості потрібно увійти в особистий кабінет

2. Створюємо нову довідку

Довідка про оцінку майна

Інформація про об'єкт оцінки *

Адреса об'єкту оцінки *

Відомості про оцінку *

Поля, позначені * - обов'язкові для заповнення

Зберегти Зберегти та відправити в ФДМУ Закрити

3. Заповнюємо всі поля

Довідка про оцінку майна

Статус звіту: Зареєстровано

Пароль пошуку: РКГЗРЦ243672

Дата створення звіту: 31.05.2023

Дата передачі звіту ФДМУ: 31.05.2023

Інформація про об'єкт оцінки *

Заявник код: 3691902076

Заявник назва: Красюк Максим Сергійович

Назва об'єкта оцінки: Дім

Вид об'єкта оцінки: житлова нерухомість

Тип об'єкта житлової нерухомості: квартира в багатоповерховій житловій будівлі

Загальна площа, кв м: 54

Площа житлових приміщень, кв м: 25

Рік введення в експлуатацію будинку/будівлі: 2012

Рік забудови: забудова з 2011

Поверховість будинку/будівлі: 25

Поверх у будинку/будівлі: 5

Кількість кімнат, шт: 1

Адреса об'єкту оцінки *

Зона населеного пункту*: серединна

Поштова адреса об'єкта оцінки*: 02130, М.КИЇВ, р. ДНІПРОВСЬКИЙ, вул. Воскресенська, буд. 12а, кв. 20

Відомості про оцінку *

Офіційний курс грн до дол США на дату оцінки: 36.57

Розмір (загальний або частки), що оцінюється, кв м*: 54

Оцінюється частка

Оціночна вартість об'єкта оцінки (повна), грн: 29007222.28

Оціночна вартість поліпшень (повна), грн: 2900722.28

Оціночна вартість земельної ділянки (повна), грн:

Поля, позначені * - обов'язкові для заповнення

Друк Закрити

4. Після заповнення анкети зберігаємо та відправляємо в ФДМУ.
5. Тепер в нас є довідка з оцінкою нашої нерухомості

**Довідка
про оціночну вартість об'єкта нерухомості**

1. Заявник (прізвище, власне ім'я та по батькові (за наявності) або найменування юридичної особи)

Красюк Максим Сергійович

3	6	9	1	9	0	2	0	7	6
---	---	---	---	---	---	---	---	---	---

(Код ЄДРПОУ, реєстраційний номер облікової картки платника податків або серія та номер паспорта (для фізичних осіб, які через свої релігійні переконання відмовляються від прийняття реєстраційного номера облікової картки платника податків та офіційно повідомили про це відповідний контролюючий орган і мають відмітку у паспорті))

2. Вид об'єкта оцінки(1)

житлова нерухомість

3. Тип об'єкта оцінки(2)

квартира в багатопверховій житловій будівлі

4. Назва об'єкта

Дім (РКГЗРЦ243672)

5. Поштова адреса

02130, М.КИЇВ, р. ДНІПРОВСЬКИЙ, вул. Воскресенська, буд. 12а, кв. 20

6. Загальна площа, кв. м

54.00

7. Площа житлових приміщень(3), кв. м

25.00

8. Рік забудови(4)

збудова з 2011

9. Рік введення в експлуатацію будинку/будівлі

2012

10. Поверховість будинку/будівлі

25

11. Поверх у будинку/будівлі(5)

5

12. Кількість кімнат у квартирі(6)

1

13. Розмір частки, що оцінюється(7), кв.м

54.00

14. Кадастровий номер земельної ділянки(8)

15. Площа земельної ділянки(9), кв.м

16. Категорія(10)

17. Цільове призначення(10)

18. Функціональне призначення(11)

19. Коефіцієнт готовності об'єкта(12), %

20. Оціночна вартість об'єкта оцінки, грн

2900722.28

20.1 Оціночна вартість поліпшень(13), грн.

2900722.28

20.2 Оціночна вартість земельної ділянки(13), грн.

Тепер розглянемо оцінку нерухомості з боку покупця на прикладі сайту DIM.RIA.

Як працює оцінка нерухомості від DIM.RIA:

- Система розрахунку.

Розрахунок здійснюється за алгоритмом DIM.RIA на основі розташування і характеристик нерухомості та аналогічних об'єктів

- Точність оцінки

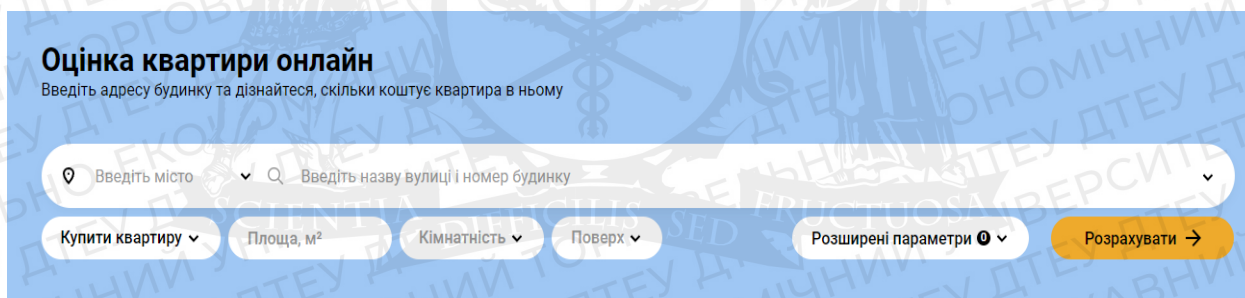
Оцінка є орієнтовною, оскільки ціна залежить від низки факторів, таких як стан житла, планування, вид з вікон тощо

- Для кого оцінка

Онлайн-оцінка допоможе визначити орієнтовну вартість усім, хто продає і купує нерухомість

Для того щоб почати оцінку нерухомості потрібно:

1. Заходимо на сайт та бачимо онлайн калькулятор



The screenshot shows the 'Оцінка квартири онлайн' (Online apartment valuation) interface. It features a search bar with two input fields: 'Введіть місто' (Enter city) and 'Введіть назву вулиці і номер будинку' (Enter street name and building number). Below the search bar are several filter buttons: 'Купити квартиру' (Buy apartment), 'Площа, м²' (Area, m²), 'Кімнатність' (Number of rooms), and 'Поверх' (Floor). There is also a 'Розширені параметри' (Advanced parameters) button and a prominent orange 'Розрахувати' (Calculate) button with a right-pointing arrow.



2. Заповнюємо необхідні поля які нас цікавлять

Оцінка квартири онлайн

Введіть адресу будинку та дізнайтеся, скільки коштує квартира в ному

📍 Київ 🔍 вулиця Воскресенська

Купити квартиру ▾ 55 м² 1 кімната ▾ Не перший / не останній ▾ Розширені параметри ⚙️ ▾ Розрахувати →

Тип стін: Не вибрано Цегла Панель Утеплена панель **Моноліт** Блок СІП панель

Дерево Інше **Згорнути**

Опалення: Не вибрано **Централізоване** Індивідуальне Комбіноване Без опалення

Ремонт: Не вибрано Житловий стан Косметичний ремонт **Євроремонт** Дизайнерський проект

Без обробки Чистова обробка Аварійний стан **Згорнути**

Планування: Не вибрано **Ізольовані кімнати** Двостороння Вільне планування Суміжна, прохідна

Студія Малосімейка, гостинка Smart-квартира Багаторівнева Пентхаус **Згорнути**

Техніка та меблі: Не вибрано **Присутні** Відсутні

Тип будівлі: Не вибрано **Сучасна забудова (економ, комфорт)** Житло підвищеного комфорту (бізнес, еліт)

Забудова радянських часів Стара забудова (до 1917 р.) **Згорнути**

Очистити всі фільтри **Розрахувати →**

3. Розраховуємо нашу нерухомість і нам видають результат

Результат оцінювання продажу 1к квартири 55 м² на вулиці Воскресенська

Скопіювати посилання ↗

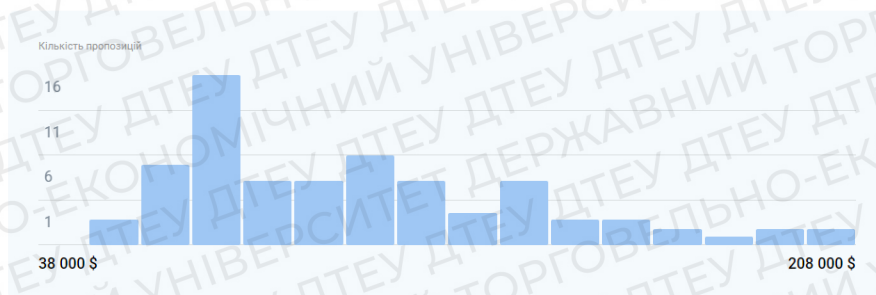
на основі аналізу 94 пропозицій зі схожими параметрами

🏠 Середня ціна за об'єкт
101 439 \$*

🏠 Середня ціна за м²
1 844 \$

грн \$ €

➔ Дивитись 74 пропозицій на DIM.RIA



Висновки до розділу 1

Автоматизоване управління ризиками підприємства з оцінки нерухомості вимагає застосування різних підходів та стратегій. Ідентифікація ризиків дозволяє виявити потенційні загрози, а аналіз ризиків допомагає оцінити їх ймовірність та вплив на діяльність підприємства.

Ефективне управління ризиками передбачає розробку та впровадження відповідних стратегій та заходів, спрямованих на мінімізацію ризиків та забезпечення безпеки підприємства. Це може включати впровадження систем моніторингу, створення запасних планів дій, залучення страхових послуг та інші заходи.

Автоматизоване управління ризиками сприяє забезпеченню стійкості та успішності діяльності підприємства з оцінки нерухомості, допомагає уникнути негативних наслідків, зберегти ресурси та забезпечити високу якість оцінки нерухомих об'єктів.

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДПРИЄМСТВА ОЦІНКИ НЕРУХОМОСТІ

2.1. Постановка задачі.

Проектування інформаційної системи підприємства оцінки нерухомості включає розробку та впровадження програмного забезпечення та інфраструктури, що дозволить здійснювати оцінку вартості нерухомості, зберігати та обробляти відповідні дані. Нижче наведено загальну процедуру проектування інформаційної системи підприємства оцінки нерухомості:

1. Визначення вимог: Спілкування з зацікавленими сторонами, включаючи менеджерів підприємства оцінки нерухомості, оцінювачів, фахівців з нерухомості та користувачів, щоб з'ясувати їх потреби та вимоги до системи. Визначте функціональність, яку потрібно реалізувати, а також технічні, безпекові та інші вимоги.

2. Аналіз процесів: Розібратися з поточними процесами оцінки нерухомості у підприємстві. Визначити ключові етапи та взаємозв'язки між ними. Це можуть бути етапи збору даних, аналізу ринку, оцінки об'єкта, формування звіту та інші.

3. Розробка бази даних: Створити базу даних, яка буде використовуватися для зберігання інформації про нерухомість, клієнтів, ринкових даних тощо. Визначте необхідні таблиці, поля та взаємозв'язки між ними. Розробіть схему бази даних та специфікації таблиць.

4. Розробка програмного забезпечення: Створіть програмне забезпечення для реалізації функціональності системи. Це може включати модулі для введення даних, обробки даних, аналізу, генерації звітів, розрахунку вартості тощо. Використовуйте відповідні технології програмування та дотримуйтесь кращих практик розробки програмного забезпечення.

Основні вимоги до програмного забезпечення з оцінки нерухомості

- **Функціональність:** Програмне забезпечення повинно мати набір функцій, які дозволяють користувачам здійснювати оцінку нерухомості. Це може включати розрахунок ринкової вартості, оцінку потенційного доходу, аналіз ризиків та інші релевантні аспекти.
- **Використання даних:** Програмне забезпечення повинно бути здатним обробляти та аналізувати різноманітні дані, які впливають на оцінку нерухомості. Це можуть бути дані про ринкові ціни на нерухомість, характеристики об'єкта, історичні дані про продажі та інші релевантні джерела.
- **Надійність:** Програмне забезпечення повинно бути стабільним і надійним у роботі, забезпечуючи точність та достовірність результатів оцінки. Воно повинно мати вбудовані механізми для перевірки та валідації даних, а також уникати помилок під час розрахунків.
- **Зручність використання:** Програмне забезпечення повинно мати інтуїтивний і простий у використанні інтерфейс користувача. Воно повинно бути зрозумілим та доступним для фахівців у галузі оцінки нерухомості, а також може включати функції, що полегшують введення даних та взаємодію з іншими системами.

2.2. Система вхідних показників визначення оцінки нерухомості.

Система вхідних показників визначення оцінки нерухомості - це набір факторів, що використовуються для визначення вартості нерухомості або оцінки її ринкової вартості. Ці показники зазвичай враховуються експертами-оцінювачами, банками, страховими компаніями та іншими зацікавленими сторонами при визначенні цінності власності.

Основні вхідні показники, що використовуються при оцінці нерухомості, можуть включати наступні:

1. Розташування:

- Географічне розташування нерухомості, включаючи район, місто або село.
- Доступність до основних доріг, громадського транспорту, зупинок, залізничних станцій тощо.
- Наявність навколишньої інфраструктури та сервісів, таких як магазини, ресторани, школи, лікарні, парки, спортивні об'єкти тощо.
- Розмір та площа:
- Загальна площа нерухомості, включаючи будівлю та земельну ділянку.
- Кількість кімнат, ванних кімнат, кухонь та інших функціональних приміщень.
- Кількість поверхів та загальна площа кожного поверху.

2. Стан нерухомості:

- Старість будівлі та рівень зносу.
- Наявність будівельних дефектів, таких як тріщини, вологість, проблеми зі зовнішньою обробкою, системами опалення, вентиляцією, електропостачанням тощо.
- Потреба в ремонті або модернізації.

3. Побудова та матеріали:

- Тип будівлі, наприклад, одноповерховий будинок, багатоповерховий житловий комплекс, котедж тощо.
- Матеріали конструкції, такі як цегла, дерево, бетон, метал тощо.
- Якість будівельних матеріалів та кінцевого оброблення.

4. Планування:

- Функціональність планування, ефективне використання простору, зручність розташування приміщень.
- Наявність додаткових особливостей, таких як гардеробні кімнати, спальні східці, великі вікна, балкони, тераси тощо.

5. Ринкові фактори:

- Аналіз ринкових умов та попиту на нерухомість в конкретному регіоні.
- Рівень конкуренції на ринку та наявність подібних об'єктів нерухомості.
- Ринкові тенденції та очікувані зміни в цінах.
- Орендні можливості:
- Приблизна орендна ставка, яку можна отримати від нерухомості, якщо вона використовується в оренду.
- Популярність району серед орендних клієнтів.

6. Історичні дані:

- Аналіз історичних цін на нерухомість в даному районі та місцевості.
- Дані про попередні транзакції, включаючи ціни продажу та додаткові умови.
- Врахування всіх цих вхідних показників дозволяє зробити більш об'єктивну оцінку вартості нерухомості, але варто зауважити, що точні показники та їх вагомість можуть різнитися в залежності від ринкових умов, регіону та специфіки нерухомості.

На основі загальних принципів оцінки нерухомості можна розглянути узагальнену формулу, яка враховує деякі основні фактори:

$$\text{Оцінка нерухомості} = (\text{Площа} * \text{Ціна за квадратний метр}) + \text{Фактор 1} + \text{Фактор 2} + \dots + \text{Фактор N}$$

У цій формулі:

- Площа відображає загальну площу нерухомості.
- Ціна за квадратний метр відображає середню ціну ринку для даного типу нерухомості в конкретному районі або регіоні.

Фактори 1, 2, ..., N відображають додаткові фактори, які можуть впливати на вартість нерухомості, такі як розташування, стан нерухомості, планування, ринкові умови тощо. Кожен фактор може мати свою власну формулу або метод оцінки, який враховує його вагу і вплив на загальну вартість.

2.3. Моделювання визначення фінансової стійкості нерухомості.

Так, модель Е. Альтмана, відома як "Z-показник Альтмана" або "Альтманський Z-показник", є однією з найвідоміших моделей оцінки ймовірності банкрутства підприємств. Ця модель була розроблена Едвардом Альтманом в 1960-х роках і базується на дискримінантному аналізі.

Модель Альтмана використовує п'ять фінансових показників для прогнозування ризику банкрутства підприємства. Ці показники вимірюють різні аспекти фінансового стану підприємства і використовуються для обчислення інтегрального Z-показника.

Формула для розрахунку Z-показника Альтмана має такий вигляд:

$$Z = 1.2 * x_1 + 1.4 * x_2 + 3.3 * x_3 + 0.6 * x_4 + x_5$$

Де:

Z - інтегральний показник рівня загрози банкрутства підприємства.

x₁ - відношення власних оборотних коштів до суми активів.

x₂ - відношення суми нерозподіленого прибутку до суми активів.

x₃ - рівень прибутковості активів.

x₄ - коефіцієнт фінансового ризику.

x₅ - коефіцієнт оборотності активів.

Значення Z-показника можуть бути інтерпретовані для прогнозування ймовірності банкрутства. Зазвичай, чим менше значення Z, тим вище ризик банкрутства.

Важливо зазначити, що модель Альтмана була розроблена для прогнозування банкрутства на основі фінансових даних і не враховує інші

фактори, такі як економічні чи індустриальні зміни. Також, модель була спочатку розроблена для оцінки банкрутства в американському контексті

Модель Z-показника Альтмана, хоча є однією з найбільш відомих і рекомендованих моделей для оцінки ймовірності банкрутства, має кілька серйозних недоліків, які були виявлені сучасними дослідниками.

По-перше, виникають сумніви щодо репрезентативності вибірки, на якій була побудована модель. Починаючи з 1968 року, Едвард Альтман використовував дані з 66 підприємств, враховуючи фінансові дані з 1946 року. Однак, за цей довгий період часу економіка зазнала значних змін, що робить порівняння даних з різних "епох" недоцільним.

По-друге, сам Едвард Альтман підбирав фактори для моделі на основі свого суб'єктивного бачення. Спочатку було відібрано 22 аналітичних коефіцієнта, а потім, на його розсуд, вибрано лише 5 найважливіших. Хоча модель показувала високу точність при тестуванні на підприємствах США (до 98% визначення ймовірності банкрутства), при застосуванні до підприємств різних країн виявлені значні варіації у вагах факторів та шкалі визначення ймовірності банкрутства.

Додатково, Альтман також відзначив, що ваги факторів та шкала моделі можуть значно відрізнятись не лише між країнами, але й у межах однієї країни з року в рік. Це свідчить про нестійкість моделі до змін вихідних даних та відсутність статистичної однорідності вибірки.

Важливо зазначити, що п'ятифакторна модель Z-показника Альтмана не може бути використана для оцінки українських підприємств через сильну кореляцію з показником фондовіддачі, що значно відрізняється в різних галузях економіки. Українські реалії також ускладнюють побудову подібних моделей через відсутність належної статистики про підприємства, які пройшли процедуру банкрутства.

Крім того, в моделі Альтмана не враховуються багато факторів, що впливають на фінансову стійкість підприємств. Також в Україні існує нестабільна та недостатньо розвинена нормативна база щодо банкрутства

підприємств. Українська економіка має свої особливості, які потребують побудови моделей оцінки загрози банкрутства на основі вітчизняних підприємств. Також для порівняння наводжу зведену таблицю основних методів визначення ризику банкрутства підприємства (табл.2.1)

Таблиця 2.1. Порівняння моделей оцінки

Назва моделі	Переваги	Недоліки
Модель Альтмана	простота та можливість застосування за наявності обмеженої інформації; порівняльність показників; можливість поділу аналізованих компаній на потенційних банкрутів і не банкрутів; висока точність розрахунків;	не враховує українські особливості економіки; складність інтерпретації підсумкового значення; залежність точності розрахунків від вихідної інформації; обмеженість сфери застосування; заснована на застарілих даних;
Модель Тафлера-Тішоу	спрощеність розрахунків та висока точність прогнозу ймовірності банкрутства компанії, що пов'язано зі значною кількістю проаналізованих компаній	обмеження сфери застосування (тільки для акціонерних товариств, акції яких активно торгуються на фондовому ринку); складність інтерпретації підсумкового значення; неможливість використання в українських умовах; залежність точності розрахунків від вихідної інформації;

Продовження табл. 2.1

<p>Модель Фулмера</p>	<p>широке охоплення показників; вірогідність: 1 рік – 98%, 2 роки – 81%</p>	<p>розроблена для застосування в банківській практиці; модель була створена з урахуванням специфіки західних країн; необхідна корекція на національну валюту</p>
<p>Модель Терещенка</p>	<p>урахування специфіки діяльності суб'єктів; охоплення незначної кількості показників; доступність інформації, необхідних для розрахунку параметрів моделі.</p>	<p>недостатній рівень обґрунтованості показників та їх нормативних значень; можливість застосування тільки для окремих підприємств; великий інтервал невизначеності.</p>
<p>Модель Спрінггейта</p>	<p>Точність моделі становить більше 90%;</p>	<p>під час розрахунків не були враховані умови розвитку економіки різних країн, орієнтовано на Канаду і США; необхідна корекція на національну валюту, яка, своєю чергою, порушує безперервність; Між змінними спостерігається досить висока кореляція.</p>

Закінчення табл. 2.1

<p>Модель Белікова</p>	<p>Давидової-</p>	<p>простота і швидкість розрахунків; доступність необхідних для розрахунків даних; можливість оцінити фінансовий стан та спрогнозувати банкрутство</p>	<p>не пристосована до українських підприємств; неможливість застосування до малих підприємств; невідповідність методичних прийомів розрахунку показників фінансової звітності; результати є завищеними, так як значний вплив на результуючий показник здійснює прибуток від реалізації.</p>
----------------------------	-------------------	--	---

2.4. Обґрунтування вибору мови програмування та технологій зберігання даних.

Наразі є такі популярні мови для програмування додатків під ОС Windows.

1. **Java** є однією з найпопулярніших мов програмування для розробки настільних додатків. Вона відома своєю надійністю, кросплатформністю та широким спектром бібліотек та фреймворків, які дозволяють розробляти різноманітні за функціональністю додатки.

2. **C#** є мовою програмування, розробленою компанією Microsoft, і використовується для розробки настільних додатків на платформі .NET. Вона має схожий синтаксис з мовою Java та надає доступ до багатого екосистеми бібліотек та інструментів для розробки Windows-додатків.

3. C++ є мовою програмування загального призначення, яка широко використовується для розробки настільних додатків, особливо тих, де вимагається висока продуктивність та контроль над ресурсами. Вона надає широкі можливості для оптимізації та роботи з низькорівневими компонентами системи.

4. Python є інтерпретованою мовою програмування, яка зарекомендувала себе як проста у використанні та читанні. Хоча вона не така швидка, як C++ або Java, Python має велику кількість сторонніх бібліотек та фреймворків, які дозволяють швидко розробляти настільні додатки з графічним інтерфейсом.

Було обрано мову C#, через її наступні переваги:

- ООП. C# використовує потужний об'єктно-орієнтований підхід, що сприяє модульності, повторному використанню коду та спрощенню управління складними проектами.
- Підтримка багатопоточності. C# надає можливість розробляти багатопотокові додатки, що дозволяє ефективно використовувати ресурси комп'ютера та підвищує продуктивність програм.
- Вбудована система безпеки. C# має потужну систему безпеки, що допомагає зменшити ризик вразливостей програм та забезпечує контроль доступу до ресурсів, а також знижує ймовірність неконтрольованого витоку пам'яті.
- Багата стандартна бібліотека та безліч сторонніх бібліотек які з легкістю інтегруються. .NET Framework, який включає в себе стандартну бібліотеку C#, надає широкий спектр готових компонентів та функцій, що спрощують розробку програм та прискорюють процес створення.
- Інтеграція з мовою розмітки XAML. C# має вбудовану підтримку мови розмітки XAML, що дозволяє розробляти багатофункціональні додатки для платформ Windows та мобільних пристроїв.

Існує два популярні формати для обміну та збереження даних XML (eXtensible Markup Language) та JSON (JavaScript Object Notation). Обидва

формати використовуються для структурованого представлення і передачі даних, проте вони мають свої особливості та використовуються в різних сферах.

XML був розроблений у 1998 році як мова розмітки, що дозволяє структуровано представляти дані. XML використовує теги для опису даних та має гнучкі правила створення власних тегів та розмітки. Це надає йому велику потужність та гнучкість, але робить синтаксис більш складним для обробки та розуміння. XML широко використовується для структурованого представлення даних у документації, конфігурації систем та обміну повідомленнями між різними системами.

З іншого боку, JSON є легковаговим форматом обміну даними, що став особливо популярним з появою веб-додатків та використанням JavaScript. JSON використовує простий синтаксис для представлення даних у вигляді пар "ключ-значення" та масивів. Його синтаксис є простим і зрозумілим для розробників, що робить його легким у використанні та обробці. JSON є універсальним форматом обміну даними, який використовується в широкому спектрі додатків, включаючи веб-додатки, мобільні додатки та мікросервіси.

Однією з основних переваг JSON є його компактність. JSON займає менше місця в пам'яті та на мережі порівняно з XML, що робить його ефективнішим для ресурсоемних сценаріїв. Крім того, JSON має простий і зрозумілий синтаксис, що полегшує роботу з ним, особливо для розробників, знайомих з JavaScript.

Однак, XML також має свої переваги. Він дозволяє більшу структуруваність даних та включає вбудовану підтримку просторів імен, схем та валідації даних. XML є більш масштабованим та розширюваним, оскільки дозволяє створювати власні елементи та розширювати існуючі схеми.

У нашому випадку більш привабливим є Json, оскільки дані у цьому форматі займають менше місця та швидше обробляються.

2.5. Проект системи.

При проектуванні системи було створено кілька класів та форм та підключено сторонню бібліотеку Newton.Json. Дерево проекту показано на рис 2.5.

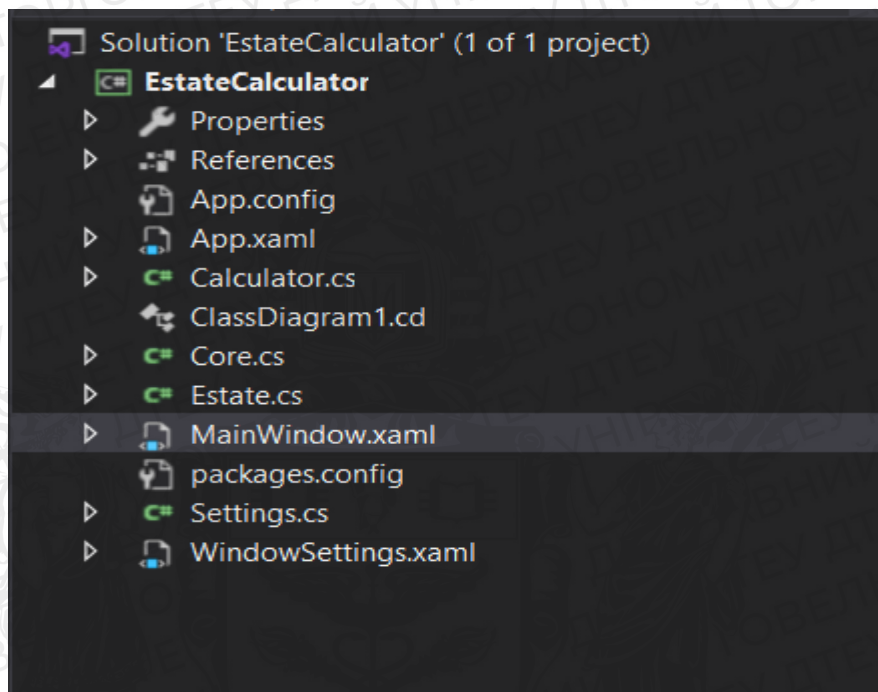


Рис.2.5. Дерево проекту

Тут помітно що створено 4 класи Calculator, Core, Estate та Settings та дві візуальні форми MainWindow та WindowSettings.

Враховуючи задані функціональні вимоги у користувача є лише два основні варіанти використання це налаштування коефіцієнтів та цін за квадратний метр та введення даних про об'єкт нерухомості з подальшим переглядом орієнтовної вартості.



Рис.2.6. Діаграма варіантів використання

Як бачимо з діаграми використання на рис 2.6. у користувача є лише два описані вище варіанти використання. Проте разом з тим вони розширюються та включають інші варіанти використання.

Так налаштування коефіцієнтів та цін включає в себе налаштуванням коефіцієнтів елітності, коефіцієнтів ремонту та коефіцієнтів району та розширюється збереженням налаштувань.

А введення даних про об'єкт розширюється переглядом орієнтовної ціни об'єкту нерухомості.

Розглянемо детальніше один з варіантів використання – введення даних для подальшої оцінки об'єкту нерухомості.

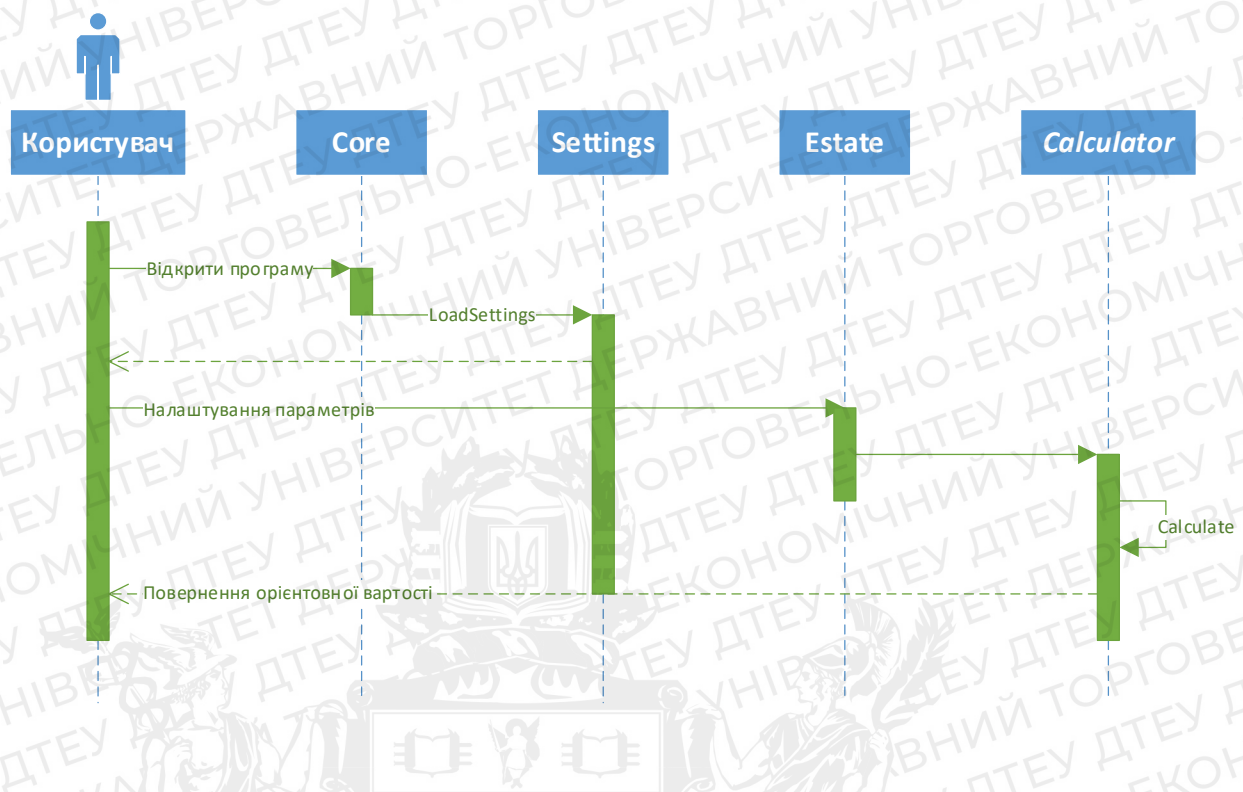


Рис.2.7. Діаграма послідовності

Як бачимо з діаграми послідовності спершу користувач відкриває форму. При цьому управління передається до об'єкту класу Core який підвантажує налаштування з файлу. Після цього користувач налаштовує особливості об'єкту нерухомості і щоразу відбувається зміни параметрів даного об'єкту нерухомості з подальшим викликом класу Calculator який викликає функцію підрахунку орієнтовної вартості. Результати підрахунку у зміненому форматі (відбувається відсікання дробової частини та додавання тексту) виводяться на формі де користувач може їх переглянути та проаналізувати.

При більш детальному розгляді діаграми стане зрозуміло, що клас Calculator виділений курсивом. Це тому, що даний клас є статичним і, відповідно, не можна створити об'єкти даного класу. Проте статична функція цього класу грає важливу роль у обчисленні результатів.

Висновки до розділу 2

Проектування інформаційної системи підприємства оцінки нерухомості є важливим етапом, який допомагає створити ефективний та функціональний інструмент для оцінювачів та фахівців з нерухомості. Визначення вимог, аналіз процесів та розробка архітектури системи є ключовими етапами проектування. Крім того, варто забезпечити відповідну інтеграцію з джерелами даних та забезпечити безпеку та захист інформації. Ефективно спроектована інформаційна система допоможе підприємству підвищити ефективність та точність процесу оцінки нерухомості та забезпечить швидкий та зручний доступ до необхідної інформації.



РОЗДІЛ 3.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДПРИЄМСТВА З ОЦІНКИ НЕРУХОМОСТІ

3.1. Архітектура програмного забезпечення системи

У існує безліч архітектурних моделей. Одна з них – клієнт-серверна архітектура, яка вже стала практично класикою. В цьому підході функції системи розділені між клієнтською та серверною сторонами. Клієнтська сторона взаємодіє з користувачем, відображає дані, виконує локальну обробку та передає запити серверу. Серверна сторона, у свою чергу, зберігає дані, виконує бізнес-логіку та обробку запитів від клієнтів. Це забезпечує централізоване керування та покращує масштабованість системи.

Ще одна архітектурна модель – трьохрівнева архітектура, також відома як N-tier архітектура. Вона розбиває систему на три рівні: рівень представлення, бізнес-логіки та рівень доступу до даних. Рівень представлення відповідає за відображення даних користувачеві та обробку його запитів. Бізнес-логіка включає в себе логіку обробки даних, розрахунків та бізнес-правил. Рівень доступу до даних забезпечує зберігання та доступ до даних, включаючи взаємодію з базами даних. Такий підхід дозволяє забезпечити модульність, шаруватість та відокремленість функцій системи.

Мікросервісна архітектура – це ще одна цікава концепція. Вона полягає в розбитті системи на невеликі незалежні сервіси, які працюють окремо і взаємодіють між собою за допомогою легковагих протоколів, наприклад, HTTP або Message Queue. Кожен сервіс має свою власну базу коду, базу даних та виконує конкретну функцію системи. Цей підхід забезпечує легке масштабування окремих сервісів, їх незалежний розвиток та гнучкість в розробці та впровадженні.

Шарова архітектура – це ще один цікавий спосіб розділити систему на логічні шари, кожен з яких виконує певну функцію. Це можуть бути шари

представлення, бізнес-логіки та доступу до даних. Кожен шар має свої власні відповідальності та інтерфейси для взаємодії з іншими шарами. Такий підхід дозволяє забезпечити модульність, легку заміну та розширення окремих шарів системи.

І на останок, сервісно-орієнтована архітектура (SOA). У цій моделі система розбивається на незалежні сервіси, які надають функціональні можливості через інтерфейси. Ці сервіси можуть використовуватися іншими системами за допомогою спеціальних протоколів комунікації. SOA спрямована на створення гнучких і розширюваних систем шляхом збирання сервісів замість монолітної архітектури.

Існує багато інших цікавих архітектурних моделей, і кожна з них має свої переваги та використання. Розуміння цих концепцій допоможе вам розробляти та вдосконалювати системи ефективніше.

Трирівнева архітектура – це підхід до проектування та розробки програмного забезпечення, який розділяє додаток на три логічно пов'язані рівні. Ця архітектура дозволяє покращити модульність, розширюваність та керованість програмного забезпечення.

Основні типи трирівневої архітектури включають:

- Рівень представлення відповідає за взаємодію з користувачем та представлення даних. Він забезпечує інтерфейс користувача, який дозволяє введення даних та отримання результатів. У цьому рівні можуть використовуватись такі компоненти, як веб-інтерфейси, десктопні додатки, мобільні додатки тощо.
- Рівень бізнес логіки містить логіку додатку та обробку даних. Він виконує всі необхідні операції та обчислення для обробки запитів користувача та підготовки відповідей. Бізнес-логіка зазвичай містить правила бізнесу, алгоритми обробки даних та взаємодії з базою даних.
- Рівень даних відповідає за збереження та доступ до даних. Він включає в себе системи управління базами даних, файлові системи або будь-

які інші механізми збереження даних. Цей рівень виконує операції збереження, витягування, оновлення та видалення даних.

Трирівнева архітектура дозволяє розділити відповідальності та функціональність між різними рівнями, що полегшує розробку та підтримку додатків. Вона також сприяє забезпеченню модульності та переносимості коду. Кожен рівень може бути розроблений та модифікований незалежно від інших, що робить систему більш гнучкою та легко змінюваною.

Існує кілька популярних архітектурних шаблонів

- MVC (Model-View-Controller): MVC є одним з найбільш відомих архітектурних шаблонів. У цій моделі додаток розділяється на три основні компоненти:

- Модель (Model): Представляє дані та бізнес-логіку додатку. Вона забезпечує доступ до даних та виконання операцій над ними.

- Представлення (View): Відповідає за візуалізацію даних, як правило, у вигляді графічного інтерфейсу користувача. Вона відображає дані з моделі та отримує вхідні дані від користувача.

- Контролер (Controller): Обробляє вхідні дані від користувача, взаємодіє з моделлю для оновлення даних та керує відображенням у представленні. Контролер служить посередником між моделлю та представленням.

MVC дозволяє розділити логіку додатку на логічно пов'язані компоненти та полегшує розробку та підтримку додатків.

- MVP (Model-View-Presenter): MVP є варіацією архітектурного шаблону MVC, який акцентує увагу на відокремленні представлення від моделі. Основні компоненти MVP включають:

- Модель (Model): Відповідає за обробку даних та бізнес-логіку, аналогічно до шаблону MVC.

- Представлення (View): Відповідає за візуалізацію даних та передачу вхідних подій презентеру.

○ Презентер (Presenter): Приймає вхідні події від представлення, виконує потрібні дії, такі як взаємодія з моделлю, і оновлює представлення згідно результатів.

MVP дозволяє відокремити логіку взаємодії з користувачем від бізнес-логіки, що полегшує тестування та підтримку коду.

• MVVM (Model-View-ViewModel): MVVM є іншою популярною архітектурою, особливо в контексті розробки інтерфейсів користувача.

Основні компоненти MVVM включають:

○ Модель (Model): Аналогічно до інших шаблонів, модель представляє дані та бізнес-логіку.

○ Представлення (View): Відповідає за відображення даних та обробку взаємодії з користувачем.

○ ViewModel: Це проміжний шар між моделлю та представленням. ViewModel надає дані та команди для представлення, а також обробляє взаємодію з користувачем. Він дозволяє зв'язати дані моделі з представленням та дозволяє легше управляти станом представлення.

MVVM забезпечує простий зв'язок між моделлю та представленням через ViewModel, що полегшує розробку інтерфейсів користувача та реагування на зміни даних.

Clean Architecture (чиста архітектура) - це архітектурний шаблон, який ставить акцент на високу роздільну здатність та незалежність компонентів програмного забезпечення. Цей підхід розробки прагне досягти максимальної гнучкості та підтримуваності системи, використовуючи сильну ізоляцію між різними рівнями та компонентами.

У Clean Architecture система розділяється на кілька рівнів абстракції, кожен з яких відповідає за конкретні функції та має свої внутрішні правила.

Основні компоненти Clean Architecture включають:

○ Сутності (Entities): Це основні бізнес-об'єкти, які містять стан та поведінку. Сутності не залежать від жодних інших компонентів і є незалежними від фреймворків або зовнішніх деталей.

- Використання (Use Cases): Вони відповідають за виконання конкретних випадків використання системи. Використання містять бізнес-логіку та координують взаємодію між різними сутностями.
- Інтерфейси (Interfaces): Це контракти або договореності, які визначають спосіб взаємодії між різними компонентами. Інтерфейси можуть бути реалізовані на різних рівнях, але вони визначаються з орієнтацією на використання, а не на деталі реалізації.
- Представлення (Presenters) та Перегляди (Views): Вони відповідають за візуалізацію даних та взаємодію з користувачем. Представлення не повинні містити бізнес-логіку, вони просто передають дані від використань до користувача та навпаки.
- Зовнішні агенти (External Agents): Це компоненти або системи зовнішнього середовища, такі як бази даних, API, фреймворки та інше. Вони повинні бути ізольовані від основної бізнес-логіки та легко замінюватись.

Одна з головних переваг Clean Architecture - це забезпечення роздільної здатності та незалежності компонентів, що полегшує тестування, розширення та підтримку системи. Вона також пропагує використання принципів SOLID, дотримання розподілення відповідальностей та забезпечення високої якості програмного забезпечення.

В нашому випадку найкраще підійде MVP. Модель це дані, класи та бізнес логіка проекту див рис 3.1.

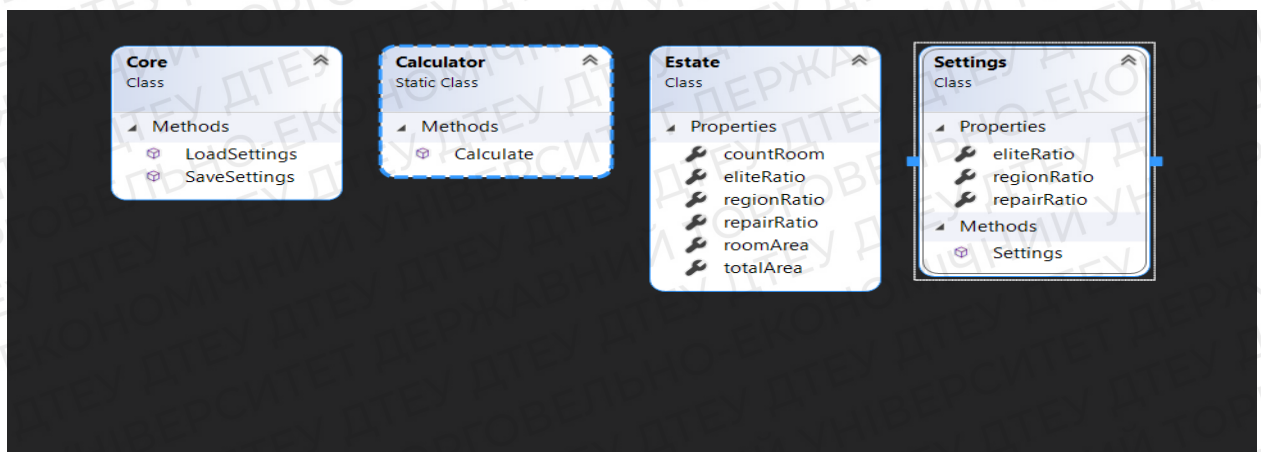


Рис.3.1. Модель даних.

Особливу увагу варто приділити класу Calculator. Це статичний клас з єдиним статичним методом у якому і відбувається обчислення. Самі обчислення відбуваються по формулі що показана на лістингу 3.1.

Лістинг 3.1. Формула обчислення вартості житла

```
public static double Calculate(Estate est, Settings s)
{
    double result;
    result =
        (
            est.roomArea
            * s.regionRatio[est.regionRatio-1]
            + (est.totalArea - est.roomArea)/2
            * s.regionRatio[est.regionRatio-1]
        )
        * s.eliteRatio[est.eliteRatio-1]
        * s.repairRatio[est.repairRatio-1]
        * Math.Pow(1.1,est.countRoom);
    return result;
}
```

Як бачимо згідно формули вартість житла обчислюється по формулі в якій за основу береться житлова площа помножена на вартість квадратного метра в регіоні та нежитлова площа помножена на половину вартості квадратного метра в регіоні. Після цього ці цифри множаться на коефіцієнт за елітність та за якість ремонту і в самому кінці враховується кількість кімнат по формулі 3.1.

$$1,1^{\text{к-сть кімнат}} \quad (3.1)$$

Таким чином чим більше кімнат тим вища вартість об'єкту нерухомості. Дана формула є узагальненою формулою що апроксимована з відкритих джерел. В реальності кожне агенство нерухомості використовує

свою власну формулу яка базується на їхній стратегії на ринку та бізнес логіці.

3.2 Тестування роботи програми

Метою дипломної роботи було розробка програмного забезпечення визначення вартості об'єкту нерухомості.

Були розглянуті питання:

- дослідження сучасних методів створення інформаційних систем;
- огляд існуючих методів дослідження об'єктів нерухомості;
- створити програмне забезпечення.
- проаналізувати отримані результати.

В третій частині було розроблено програмне забезпечення для оцінки нерухомості на базі інтегрованого середовища Visual Studio. На підставі отриманих результатів аналізу були отримані результати, які відповідають поставленим задачам.

Програмне забезпечення системи реалізовано у формі віконного інтерфейсу. На рис 3.2. наведено головне вікно програми, яке дозволяє вводити дані про об'єкт нерухомості.

The screenshot shows a window titled "EstateCalculator" with the following fields and values:

Загальна площа	62
Житлова площа	45
Кількість кімнат	2
Рейтинг району	4
Рейтинг ремонту	3
Елітність	3

Орієнтовна вартість квартири рівна 51788

Налаштування

Рис. 3.2. Головне вікно програми

Як бачимо на рис 3.2. програма передбачає введення площі житлової та нежитлової, яка може бути значенням з плаваючою крапкою, кількість кімнат, яке завжди є цілим значенням та рейтинг району, рейтинг ремонту та рейтинг елітності, які можуть приймати значення від 1 до 5. Це реалізовано за допомогою слайдерів, які налаштовані так аби значення завжди було цілим, а саме значення дублювалося у текстове поле справа з допомогою механізму зв'язування даних (data binding).

Це особливість технології WPF, яка прийшла на зміну WinForms. Всі форми програми мають захист від “дурня”, тобто при введенні невідповідних даних програма не зависне та не вилетить, а також “резиновий” інтерфейс. Див рис 3.3.

В правій частині форми є меню з налаштуваннями яке дозволяє налаштувати всі необхідні коефіцієнти.

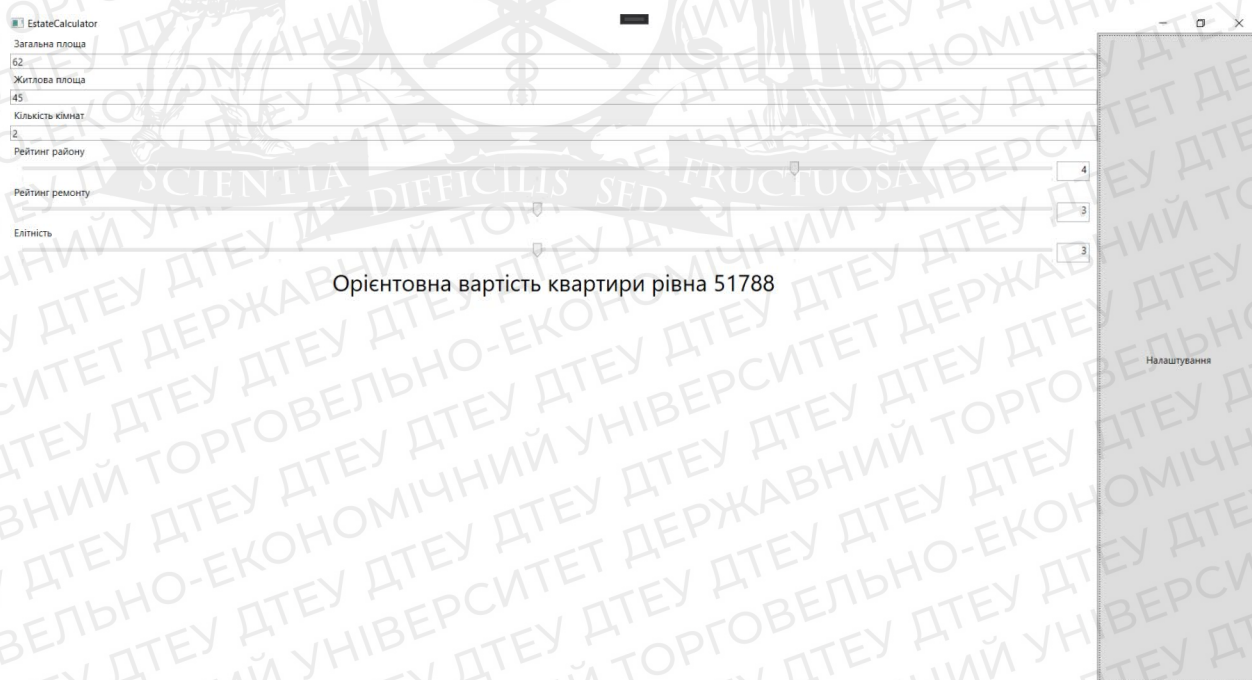


Рис.3.3. Повноекранне відкриття головної форми

Налаштування	
Ціна 1м2 при оцінці району 1	300
Ціна 1м2 при оцінці району 2	350
Ціна 1м2 при оцінці району 3	420
Ціна 1м2 при оцінці району 4	500
Ціна 1м2 при оцінці району 5	600
Коеф ціни при рейтингу ремонту 1	0,8
Коеф ціни при рейтингу ремонту 2	0,9
Коеф ціни при рейтингу ремонту 3	1
Коеф ціни при рейтингу ремонту 4	1,1
Коеф ціни при рейтингу ремонту 5	1,25
Коеф ціни при рейтингу елітності 1	1
Коеф ціни при рейтингу елітності 2	1,3
Коеф ціни при рейтингу елітності 3	1,6
Коеф ціни при рейтингу елітності 4	1,9
Коеф ціни при рейтингу елітності 5	2,2
Зберегти	
Закрити	

Рис. 3.4. Меню налаштувань

Варто зазначити що дані налаштування тестові і для роботи додатку “в полі” необхідно провести додатковий аналіз ринку для введення відповідних даних. Ці налаштування зберігаються між сесіями роботи додатку завдяки серіалізації в json файл, який знаходиться поряд з виконавчим файлом та має назву data.ini. для серіалізації даних та подальшої десеріалізації використовується бібліотека json.

Це безкоштовна бібліотека для роботи з даними Json, окрім серіалізації та десеріалізації дає безліч додаткових функцій для роботи з даними у форматі json. На рис 3.5 показано структуру файлу що зберігає налаштування між сесіями.


```
1 {
2   "regionRatio": [
3     300.0,
4     350.0,
5     420.0,
6     500.0,
7     600.0
8   ],
9   "repairRatio": [
10    0.8,
11    0.9,
12    1.0,
13    1.1,
14    1.25
15  ],
16  "eliteRatio": [
17    1.0,
18    1.3,
19    1.6,
20    1.9,
21    2.2
22  ]
23 }
```

Рис.3.5. Налаштування

Як бачимо файл містить серіалізовані дані з класу що містить три масиви даних кожен по п'ять елементів. Це масиви що відповідають за рейтинг районі, рейтинг ремонту та рейтинг елітності.

ВИСНОВКИ

Розробка ефективних інформаційних систем з оцінки нерухомості значно підвищує актуальність роботи зі застосування діджиталізованих методик до організації роботи відповідних підприємств, що беруть на себе роль оцінки нерухомості.

Одним із підходів для оцінки нерухомості є використання математичних моделей та інформаційних технологій.

В роботі був проведений аналіз інформаційних систем з оцінки нерухомості та визначені основні підходи та потреби до розробки програми.

Розроблено програмне забезпечення на мові C# з використанням формату збереження даних JSON, що дало створити функціональну програму по оцінці нерухомості.

В першому розділі було розглянуто основну проблематику управління оцінкою нерухомості в умовах сучасної економіки, та основні методи оцінки нерухомості в сучасних ринкових умовах для передбачення фінансової ситуації в майбутньому. Для прикладу були наведені існуючі програмні засоби з оцінки нерухомості, метою яких є надання об'єктивної оцінки на підставі різних факторів.

У другому розділі досліджувалися різні методи аналізу підприємств, що сприяло створенню повноцінної картини для розробки алгоритму інформаційної системи. Обрали та обґрунтували більш зручну мову програмування та технологію зберігання даних. Також ми спроектували інформаційну систему, та забезпечили відповідну інтеграцію з джерелами даних.

Створена мною програма повністю відповідає та задовольняє вимогам завдання. За рахунок можливості програмування та підключення макросів можливо створити візуальне та інтуїтивне програмне забезпечення, для використання якого не потрібно мати технічної освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Авдющенко А.С. Обґрунтування можливості використання агрегованого показника для оцінки економічного стану підприємства / А.С. Авдющенко // Науковий вісник №5. – 2010. [Електронний ресурс] – Режим доступу до журналу.: <http://www.nbuu.gov.ua/>.
2. Білик М. Д. Сутність і оцінка фінансового стану підприємств / М.Д. Білик // Фінанси України № 3. – 2015.
3. Барановська О.М. Аналіз фінансового стану як інструмент запобігання кризі та виведення підприємства з неї /О.М. Барановська // Вісник Придніпровської державної академії будівництва та архітектури. – Дніпропетровськ: ПДАБА, 2012. – № 12. [Електронний ресурс]. – Режим доступу до журналу.: http://www.nbuu.gov.ua/portal/natural/Vpabia/2008_12/stat/UDK%20334.htm.pdf.
4. Давидова Г. В. Методика кількісної оцінки ризику банкрутства підприємства / Г. В. Давидова, А. Ю. Беликов // Управління ризиком №3. – 2010.
5. Овдій Л.І., Некрасова Я.А. Оцінка інвестиційної привабливості підприємств за допомогою статистичних моделей / Л.І. Овдій, Я.А. Некрасова// Вісник Хмельницького національного університету №4. – 2010. [Електронний ресурс]. – Режим доступу до журналу.: <http://www.nbuu.gov.ua/>.
6. Офіційний сайт Національної бібліотеки України ім. Вернадського. – [Електронний ресурс]. – Режим доступу: <http://www.nbuu.gov.ua>
7. Терещенко О. О . Дискримінанта модель інтегральної оцінки фінансового стану підприємства / О. О. Терещенко // Економіка України №8. – 2013. – С. 38-44.
8. Терещенко О. О. Фінансова діяльність суб'єктів господарювання: Навч. посібник. / О. О. Терещенко – Київ: КНЕУ. – 2013.

9. Даниленко В.А. Застосування моделей діагностики банкрутства для оцінки фінансової стійкості промислових підприємств України / В.А. Даниленко// Збірник наукових праць “Науковий вісник Академії муніципального управління”, серія “ЕКОНОМІКА” – випуск 7. – 2012 р. [Електронний ресурс]. – Режим доступу до журналу: http://www.nbu.gov.ua/portal/Soc_Gum/Nvamu/Ekon/2009_7/09dvaeu.pdf.
10. Фучеджи В. І. Характеристика методів та моделей діагностики кризового стану підприємства / В. І. Фучеджи [Електронний ресурс] – Режим доступу: <http://ena.lp.edu.ua>
11. Балабанова Л.В., Алачева Т.І. Інформаційне забезпечення обґрунтування управлінських рішень в умовах маркетингової орієнтації підприємства: Монографія. / Л.В. Балабанова // Т.І. Алачева - Донецьк: ДонДУЕТ ім. М. Туган-Барановського, 2013.
12. Бороненкова С.А. Економічний аналіз в управлінні підприємством. / С. А. Бороненкова - М.: Фінанси і статистика, 2013. - 224с.
13. Ванькович Д.В. Удосконалення системи управління фінансовими ресурсами промислових підприємств. / Д. В. Ванькович // Фінанси України. - 2012. - № 7. - С. 44-50.
14. Годін В.В., Корнєєв І.К. Інформаційне забезпечення управлінської діяльності: Підручник. /В.В. Годін // І.К.Корнєєв - М.: Майстерність; Вища школа, 2001. - 240с.
15. Енциклопедія бізнесмена, економіста, менеджера / За ред. Р. Дяківа. - К.: Міжнародна економічна фундація, 2010. - 704с.
16. Інформаційні системи и технології в обліку: Навч. посібник / В.Д. Шквір, А.Г. Загородній, О.С. Височан. - Львів: Видавництво національного університету "Львівська політехніка", 2013. - 268с.
17. Кузьмін О.Є., Георгіаді Н.Г. Формування и использование інформаційної системи управління економічним розвитком підприємства: Монографія. / О.Є. Кузьмін // Н.Г. Георгіаді - Львів: Вид-во Нац. ун-ту "Львівська політехніка", 2016. - 368 с.

18. Джай К. Шим, Джойл Г. Сігел. Основи бюджетування і більше: Довідник по складанню бюджетів / Пер. з англ .; За заг. ред. В.А. Плотникова. - М .: Вершина, 2017. - 368с.
19. Концепція Business Performance Management: початок шляху / Є.Ю. Духонін, Д.В. Ісаєв, Е.Л. Мостовий та ін .; Під ред. Г.В. Генса. - М .: Альпіна Бізнес Букс, 2014. - 269 с.
20. Сухарева Л.А., Петренко С.Н. Контролінг - основа управління бізнесом. / Л.А. Сухарева // С.Н. Петренко - К .: Ельга, Ніка-Центр, 2012. - 208 с.
21. Давидова Г.В. Методика кількісної оцінки ризику банкрутства підприємств [Текст] / Г. В. Давидова // Управління ризиком. - 2010. - №6.
22. Дуброва Т.А. та ін. Аналіз часових рядів і прогнозування в системі «STATISTICA» [Текст]: навч. посібник / Т.А. Дуброва // - М.: Юніті, 2012.
23. Журов В.А. Процес розробки моделей для прогнозування банкрутства підприємств [Текст] / В. А. Журов // Фінансовий менеджмент. - 2017.
24. Коваленко О.В. Діагностика стану підприємства на основі нечітких продукційних систем і дискримінантного аналізу [Текст] / О.В Коваленко // Економічний аналіз: теорія і практика. - 2017. - №14.
25. Халафяна А.А. Факторний, дискримінантний та кластерний аналіз [Текст]: навч. посібник / А.А. Халафяна - М.: Фінанси і статистика, 2017.
26. Хайдаршіна Г.А. Методи оцінки ризику банкрутства підприємства [Текст]: автореф. дис. канд. екон. наук / Г.А. Хайдаршіна. - М., 2012.

ДОДАТКИ

Програмний код реалізації програми

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace EstateCalculator

{

public static class Calculator

{

public static double Calculate(Estate est, Settings s)

{

double result;

result =

(

est.roomArea

* s.regionRatio[est.regionRatio-1]

+ (est.totalArea - est.roomArea)/2

* s.regionRatio[est.regionRatio-1]

)

* s.eliteRatio[est.eliteRatio-1]

* s.repairRatio[est.repairRatio-1]

* Math.Pow(1.1,est.countRoom);

return result;


```
using Newtonsoft.Json;
using System;
using System.IO;

namespace EstateCalculator
{
    class Core
    {
        public Settings LoadSettings()
        {
            string path = Environment.CurrentDirectory + "\\data.ini";
            string json = "";
            string[] SB;
            try
            {
                SB = File.ReadAllLines(path);
                for (int i = 0; i < SB.Length; i++)
                    json += SB[i];
                Settings tmp = JsonConvert.DeserializeObject<Settings>(json);
                return tmp;
            }
            catch (Exception ex)
```

```

    }
    Settings tmp = new Settings();
    return tmp;
}
}

public void SaveSettings(Settings s)
{
    string path = Environment.CurrentDirectory + "\\data.ini";
    string json = JsonConvert.SerializeObject(s, Formatting.Indented);
    File.WriteAllText(path, json);
}
}
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EstateCalculator
{
    public class Estate
    {
        public int regionRatio { get; set; }
        public int repairRatio { get; set; }
    }
}

```



```
public int eliteRatio { get; set; }
public double totalArea { get; set; }
public double roomArea { get; set; }
public double countRoom { get; set; }
}
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EstateCalculator
{
    public class Settings
    {
        public double[] regionRatio { get; set; }
        public double[] repairRatio { get; set; }
        public double[] eliteRatio { get; set; }
        public Settings()
        {
            regionRatio = new double[5];
            repairRatio = new double[5];
            eliteRatio = new double[5];
        }
    }
}
```

```
}  
}  
}  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows;  
using System.Windows.Controls;  
using System.Windows.Data;  
using System.Windows.Documents;  
using System.Windows.Input;  
using System.Windows.Media;  
using System.Windows.Media.Imaging;  
using System.Windows.Navigation;  
using System.Windows.Shapes;  
  
namespace EstateCalculator  
{  
    /// <summary>  
    /// Interaction logic for MainWindow.xaml  
    /// </summary>  
    public partial class MainWindow : Window  
    {
```



```
public Settings s;
public Estate est = new Estate();
public MainWindow()
{
    InitializeComponent();
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    WindowSettings ws = new WindowSettings();
    ws.s = s;
    ws.ShowDialog();
}

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    s = new Core().LoadSettings();
}

private void TotalArea_TextChanged(object sender, TextChangedEventArgs
e)
{
    Calc();
}

public void Calc()
{
```

```

try
{
    est.countRoom = Convert.ToInt32(CountRoom.Text);
    est.totalArea = Convert.ToDouble(TotalArea.Text);
    est.roomArea = Convert.ToDouble(RoomArea.Text);
    est.eliteRatio = Convert.ToInt32(RE.Text);
    est.regionRatio = Convert.ToInt32(RR.Text);
    est.repairRatio = Convert.ToInt32(RP.Text);
    Result.Content="Орієнтовна вартість квартири рівна " +
    Calculator.Calculate(est, s).ToString("F0");
}
catch {}
}

private void RoomArea_TextChanged(object sender, TextChangedEventArgs
e)
{
    Calc();
}

private void CountRoom_TextChanged(object sender,
TextChangedEventArgs e)
{
    Calc();
}

```



```
private void RR_TextChanged(object sender, TextChangedEventArgs e)
{
    Calc();
}
```

```
private void RP_TextChanged(object sender, TextChangedEventArgs e)
{
    Calc();
}
```

```
private void RE_TextChanged(object sender, TextChangedEventArgs e)
{
    Calc();
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
```

using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace EstateCalculator

{

/// <summary>

/// Interaction logic for WindowSettings.xaml

/// </summary>

public partial class WindowSettings : Window

{

public Settings s;

public WindowSettings()

{

InitializeComponent();

}

private void Window_Loaded(object sender, RoutedEventArgs e)

{

RR1.Text = s.regionRatio[0].ToString();

RR2.Text = s.regionRatio[1].ToString();

RR3.Text = s.regionRatio[2].ToString();

RR4.Text = s.regionRatio[3].ToString();

RR5.Text = s.regionRatio[4].ToString();


```
RP1.Text = s.repairRatio[0].ToString();
RP2.Text = s.repairRatio[1].ToString();
RP3.Text = s.repairRatio[2].ToString();
RP4.Text = s.repairRatio[3].ToString();
RP5.Text = s.repairRatio[4].ToString();

RE1.Text = s.eliteRatio[0].ToString();
RE2.Text = s.eliteRatio[1].ToString();
RE3.Text = s.eliteRatio[2].ToString();
RE4.Text = s.eliteRatio[3].ToString();
RE5.Text = s.eliteRatio[4].ToString();
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    try
    {
        s.regionRatio[0] = Convert.ToDouble(RR1.Text);
        s.regionRatio[1] = Convert.ToDouble(RR2.Text);
        s.regionRatio[2] = Convert.ToDouble(RR3.Text);
        s.regionRatio[3] = Convert.ToDouble(RR4.Text);
        s.regionRatio[4] = Convert.ToDouble(RR5.Text);
    }
}
```

```
s.repairRatio[0] = Convert.ToDouble(RP1.Text);
s.repairRatio[1] = Convert.ToDouble(RP2.Text);
s.repairRatio[2] = Convert.ToDouble(RP3.Text);
s.repairRatio[3] = Convert.ToDouble(RP4.Text);
s.repairRatio[4] = Convert.ToDouble(RP5.Text);

s.eliteRatio[0] = Convert.ToDouble(RE1.Text);
s.eliteRatio[1] = Convert.ToDouble(RE2.Text);
s.eliteRatio[2] = Convert.ToDouble(RE3.Text);
s.eliteRatio[3] = Convert.ToDouble(RE4.Text);
s.eliteRatio[4] = Convert.ToDouble(RE5.Text);

new Core().SaveSettings(s);
Close();
}
catch { }
}

private void Button_Click_1(object sender, RoutedEventArgs e)
{
    Close();
}
}
```