

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій проектування та прикладної математики

(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

на тему: «Розробка інформаційної системи моделювання технічного стану
будівельних конструкцій»

ЛУК`ЯНЕНКО МИХАЙЛО ВОЛОДИМИРОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій проектування та прикладної математики

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТППМ
д.т.н., професор Бородавка Є.В.

„___” _____ 2025 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

на тему: "Розробка інформаційної системи моделювання технічного стану
будівельних конструкцій"

Виконав: студент II-го курсу, групи ІСТм-24

Спеціальності: 126 «Інформаційні системи і
технології»

(шифр і назва спеціальності)

Лук`яненко М.В.
(прізвище та ініціали)

Керівник д.т.н., проф. Терентьєв О.О.
(прізвище та ініціали)

Рецензент к.т.н., доц. Шабала Є.Є.
(прізвище та ініціали)

Київ, 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій проектування та ПМ

Освітній рівень: «магістр за ОПІ»

Спеціальність: 126. «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТПМ
д.т.н., професор Бородавка Є.В.

„___” _____ 2025 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

Лук`яненко Михайло Володимирович

1. Тема роботи: Розробка інформаційної системи моделювання технічного стану будівельних конструкцій
затверджена наказом ректора КНУБА № 1619/23/25 від «25» вересня 2025 р.
2. Керівник роботи: Терентьев Олександр Олександрович, д.т.н, професор кафедри інформаційних технологій проектування і прикладної математики
3. Строк подання студентом роботи до захисту: грудень 2025 рік
4. Зміст пояснювальної записки за розділами:
 - Р.1. Загальна характеристика категорій стану будівельних конструкцій
 - Р.2. Концептуальне та функціональне проектування технології
 - Р.3. Математична постановка задачі
 - Р.4. Проектування бази даних та архітектури системи
 - Р.5. Тестовий приклад програми
5. Інформаційні слайди:
 - С.1 Структурна схема ІТ системи діагностики технічного стану будівель
 - С.2. Дерево цілей, дерево функцій системи
 - С.3. Концептуальна та фізична модель бази даних
 - С.3. Схема алгоритму роботи системи
 - С.4. Програмне забезпечення системи. Тестовий приклад програми

6. Календарний план виконання кваліфікаційної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Загальна характеристика категорій стану будівельних конструкцій	Вересень 2025 р.
Р. 2. Концептуальне та функціональне проектування технології	Жовтень 2025 р.
Р. 3. Математична постановка задачі	Жовтень 2025 р.
Р. 4. Проектування бази даних та архітектури системи	Листопад 2025 р.
Р. 5. Тестовий приклад програми	Листопад 2025 р.
Остаточне оформлення роботи	Грудень 2025 р.
Направлення роботи на рецензування, перевірку на плагіат	Грудень 2025 р.
Попередній захист роботи на кафедрі	Грудень 2025 р.

7. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1.	к.т.н., доц. Шабала Є.Є.	02.12.2025 р.	
Розділ 2.			
Розділ 3.			
Розділ 4.			
Розділ 5.			
Розділ 6.			

8. Дата видачі завдання: 22 вересня 2025 року

Керівник

(підпис)

Терентьєв О.О.

(прізвище та ініціали)

Магістрант

(підпис)

Лук`яненко М.В

(прізвище та ініціали)

АНОТАЦІЯ

«Розробка інформаційної системи моделювання технічного стану будівельних конструкцій».

Кваліфікаційна робота магістра за спеціальністю: 126. «Інформаційні системи і технології» – Київський національний університет будівництва та архітектури. – Київ, 2025.

Магістерська робота присвячена вирішенню актуального завдання автоматизації роботи експертів у сфері оцінки технічного стану будівельних об'єктів. У роботі запропоновано технологію автоматизації оцінки технічного стану мостових конструкцій, що ґрунтується на обчисленні площі пошкоджених ділянок поверхонь будівельних конструкцій.

Ключові слова: мостова конструкція, автоматизація, оцінка технічного стану, САПР.

SUMMARY

"Development of information technology for modeling the technical condition of building structures."

Master's attestation master's degree in specialty: 126. "Information systems and technologies". - Kyiv National University of Civil Engineering and Architecture - Kiev, 2025.

Master's thesis is devoted to solving the urgent problem of automation of the work of experts in the field of evaluation of the technical state of the objects. The paper presents the automation technology for technical condition assessment of bridge structures based on the calculation of the area of the damaged surface areas of building structures.

Keywords: bridge construction, automation, evaluation of technical condition, CAD.

ЗМІСТ

Вступ

Розділ 1. Загальна характеристика категорій технічного стану будівельних конструкцій

- 1.1 Основні поняття та визначення
- 1.2 Види мостів та умови їх функціонування
- 1.3 Нормативні вимоги до надійності і безпеки мостових конструкцій
- 1.4 Методи і засоби визначення технічного стану мостів
- 1.5 Аналіз типових інформаційних технологій діагностики технічного стану
- 1.6 Постановка задачі

Розділ 2. Концептуальне та функціональне проектування технології

- 2.1 Дерево цілей системи
- 2.2 Дерево функцій системи
- 2.3 Концептуальна модель системи
- 2.4 Діаграми потоків даних
- 2.5 Розробка інтерфейсу користувача

Розділ 3. Математична постановка задачі

- 3.1 Вибір алгоритму побудови опуклої оболонки
- 3.2 Визначення площі пошкоджених ділянок
- 3.3 Технологія оцінки технічного стану

Розділ 4. Проектування бази даних та архітектури системи

- 4.1 Визначення сутностей бази даних системи
- 4.2 Концептуальна модель бази даних
- 4.3 Логічна модель бази даних
- 4.4 Фізична модель бази даних
- 4.5 Діаграма основних класів системи
- 4.6 Опис алгоритму функціонування системи

Розділ 5. Контрольний приклад роботи програми

5.1 Аналіз інструментальних засобів розробки системи

5.2 Аутентифікація та авторизація

5.3 Форма вибору конструктивного елемента

5.4 Форма оцінки технічного стану

Висновки

Список літератури

ВСТУП

Актуальність теми. Технічний стан конструктивної системи споруди в цілому визначають на підставі встановленого технічного стану окремих конструктивних компонентів з врахуванням їх значимості і категорій відповідальності. В оцінюванні технічного стану споруди та її компонентів береться до уваги вплив систем технічного обладнання споруди, а також взаємний вплив споруди та навколишнього природного і техногенного середовища. Терміни планових обстежень технічного стану споруд встановлюються з умови, що до наступного обстеження технічний стан споруди буде підтримуватись на рівні, необхідному для забезпечення роботоспроможності та безпечності об'єкта. У період між плановими обстеженнями споруди досить часто виникають потреби у позапланових обстеженнях для додаткового визначення її технічного стану. Причинами позапланових обстежень є: виявлення нових значних дефектів і пошкоджень; виявлення або прогнозування небезпечних змін в умовах експлуатації споруди, які змінюють передбачені проектом навантаження, впливи, інженерно-геологічну, гідрогеологічну або іншу ситуацію чи конструктивну схему споруди; планування заходів з відновлення експлуатаційної придатності об'єктів, їх адаптації до зміни умов використання, консервації (розконсервації) або ліквідації споруд.

Проведення позапланового обстеження потребує значних матеріальних витрат. Тому проблема знаходження альтернативного, економічнішого способу проведення позапланових обстежень є **актуальною**.

Метою роботи є розробка інформаційної технології, в якій застосовується фото обладнання і математичні алгоритми для оцінки технічного стану будівельного об'єкту.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати предметну область, порівняти існуючі технології, взяти до уваги їх плюси і мінуси;
- розробити дерево цілей та функцій, концептуальну модель системи;
- побудувати діаграми потоків даних;

- обрати алгоритм побудови опуклої оболонки навколо пошкоджених ділянок, метод знаходження їх площі;
- запроектувати базу даних;
- розробити інтерфейс користувача та протестувати програмний продукт.

Об'єктом дослідження є технічний стан будівельних конструкцій.

Предметом дослідження є інформаційна технологія оцінки технічного стану будівельних конструкцій.

Методи дослідження: математичні (системний аналіз, теорія ймовірностей) і механіки руйнувань (метод аналогій і побудови опуклої оболонки).

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА КАТЕГОРІЙ ТЕХНІЧНОГО СТАНУ БУДІВЕЛЬНИХ КОНСТРУКЦІЙ

1.1 Основні поняття та визначення

Будівельними несучими конструкціями промислових і цивільних будинків та інженерних споруд називаються конструкції, розміри перерізів яких визначаються розрахунком. Це основна їх відмінність від архітектурних конструкцій або частин будівель, розміри перерізів яких призначаються згідно архітектурним, теплотехнічним або іншим спеціальним вимогам.

Сучасні будівельні конструкції повинні задовольняти наступним вимогам: експлуатаційним, екологічним, технічним, економічним, виробничим, естетичним та ін [1, 2].

Будинок – результат будівництва, що являє собою об'ємну будівельну систему, має надземну і (чи) підземну частини, включає у собі приміщення, мережі інженерно-технічного забезпечення і системи інженерно-технічного забезпечення і призначений для проживання і (чи) діяльності людей, розміщення виробництва, зберігання продукції або утримання тварин.

Споруда – результат будівництва, що являє собою об'ємну, площинну чи лінійну будівельну систему, має наземну, надземну і (чи) підземну частини, складається з несучих, а в окремих випадках і огорожуючих будівельних конструкцій і призначену для виконання різноманітних виробничих процесів, зберігання продукції, тимчасового перебування людей, переміщення вантажів [3].

Унікальні будівлі і споруди – споруди, де в проектній документації передбачена хоча б одна з таких характеристик:

- використання конструкцій і конструктивних систем, що потребує нестандартних методів розрахунку, або розробки спеціальних методів розрахунку, або потребують експериментальної перевірки на фізичних моделях, і навіть на територіях, сейсмічність яких перевищує 9 балів;
- висота понад сто м;

- проліт понад сто м;
- виліт консолей більш 20 м;
- заглиблення підземної частини нижче планувальної позначки понад 10 метрів.

До унікальних будівель і споруд слід зарахувати, також, видовищні, спортивні, культові споруди, виставкові павільйони, багатофункціональні офісні, торговельно-розважальні комплекси тощо, з максимальним розрахунковим перебуванням понад тисячу людина всередині об'єкта або 10000 людей поблизу об'єкта [4].

Життєвий цикл будівлі та споруди – період, протягом якого здійснюються інженерні пошуки, проектування, будівництво (зокрема консервація), експлуатація (зокрема поточні ремонти), реконструкція, перегляд, знос будівлі споруди.

Вплив – явище, що викликає зміну напружено-деформованого стану будівельних конструкцій і (чи) основ будівлі чи споруди [5].

Навантаження – механічна сила, прикладена до будівельних конструкцій і основ будівлі та визначає їх напружено-деформований стан.

Визначальний елемент - елемент, руйнування якого призводить до руйнування або загрози руйнування споруди в цілому.

Дефект - відхилення якості, форми, фактичних розмірів елементів та конструкцій від вимог нормативної або проектної документації, яке виникає при проектуванні, виготовленні, транспортуванні, будівництві або в процесі експлуатації.

Експлуатаційний стан - технічний стан, що описується добіркою загальноприйнятих числових і неформальних лінгвістичних характеристик.

Допоміжний елемент - елемент, який не входить до складу несучих конструкцій моста. Руйнація такого елемента не призводить до зміни напружено-деформованого стану споруди, а тільки ускладнює її експлуатацію (наприклад, бар'єрна огорожа, перила, оглядові пристрої).

Довговічність - здатність споруди зберігати протягом певного часу роботоспроможний стан при встановленій системі технічного обслуговування. Довговічність визначається в роках.

Другорядний елемент - елемент, руйнація якого не призводить до руйнації моста в цілому, але є серйозною загрозою руйнації при подальшій його експлуатації (наприклад, діафрагми збірної залізобетонної прогонової будови).

Деградація - знос, втрата початкових механічних, фізико-хімічних і естетичних характеристик. Рівень деградації в функції часу може прогнозуватись відповідними моделями.

Знос споруди - деградація елементів споруди в процесі експлуатації. Полягає в погіршенні початкових проектних показників, таких як несуча здатність, вантажопідйомність, невідповідності розрахункових параметрів споруди сучасним вимогам (наприклад, за габаритом тощо).

Експлуатація споруди - технічне використання споруди згідно з призначенням, догляд та збереження технічного стану, передбаченого проектом.

Експлуатаційне утримання - вжиття необхідних заходів із збереження стану конструкцій, за якого вони здатні виконувати функції, передбачені проектом, з параметрами, що визначені проектом та вимогами нормативних документів з експлуатації споруд.

Нормальні умови експлуатації – розрахований під час проектування стан будівлі споруди, у якому відсутні будь-які чинники, що перешкоджають здійсненню функціональних чи технологічних процесів.

Капітальний ремонт - комплекс будівельно-монтажних робіт, який полягає у відновленні, посиленні або заміні елементів моста, що втратили свої проектні показники.

Міст – транспортна споруда призначена для пропуску через перешкоду потоків автомобільного, залізничного транспорту, пішоходів, інших комунікацій. У залежності від функціонального призначення та типу

перешкоди мости мають свої специфічні назви (наприклад, шляхопровід, віадук, естакада).

Надійність - здатність споруди виконувати задані функції в певних умовах експлуатації, зберігаючи протягом встановленого часу нормативні експлуатаційні показники. Надійність визначається ймовірністю того, що не буде досягнуто жодного з розрахункових граничних станів.

Несуча здатність - здатність елемента (конструкції) сприймати граничне зусилля.

Обстеження - процес отримання якісних та кількісних показників експлуатаційної придатності споруди, елементів та конструкцій шляхом їх візуального огляду, інструментальних вимірювань, польових та лабораторних випробувань матеріалів конструкцій.

Пошкодження - дефект, відхилення параметрів і/або характеристик елементів та конструкцій від передбачених проектом, що виникли в результаті природного старіння матеріалів та механічних, фізико-хімічних, теплотехнічних впливів у процесі транспортування, монтажу або експлуатації.

Реконструкція - комплекс будівельно-монтажних робіт, спрямованих на відновлення або перебудову споруди з наданням їй потрібних експлуатаційних характеристик.

Ремонт - комплекс будівельно-монтажних робіт, спрямованих на відновлення проектних параметрів споруди.

Ресурс - строк роботи елемента в роках, впродовж якого надійність і експлуатаційна придатність задовольняють нормативні вимоги.

Характеристика безпеки - числовий показник надійності, математично зв'язаний з ймовірністю того, що не буде досягнуто граничного стану. Так, наприклад, надійності $P = 0,9998$ відповідає характеристика безпеки $\beta = 3,8$.

Фізичний знос – погіршення технічних та експлуатаційних показників будинку або споруди, що викликане об'єктивними причинами.

Моральний знос – поступове (у часі) відхилення основних експлуатаційних показників від рівня технічних вимог експлуатації будинків та споруд.

Поточний технічний стан будинків та споруд – технічний стан будинків та споруд на момент їх обстеження чи проведення етапу моніторингу.

Прийнято, що елементи моста протягом життєвого циклу перебувають послідовно в одному з п'яти експлуатаційних станів [6].

Таблиця 1.1 Класифікація експлуатаційних станів елементів

Експлуа- таційний стан	Назва експлуатаційного стану	Узагальнена характеристика стану
Стан 1	Справний	Елемент відповідає всім вимогам проекту та чинних норм експлуатації
Стан 2	Обмежено справний	Елемент частково не відповідає вимогам проекту, проте не порушуються вимоги ані першої, ані другої груп граничних станів
Стан 3	Працездатний	Елемент частково не відповідає вимогам проекту, проте не порушуються вимоги першої групи граничних станів. Можливе часткове порушення вимог другої групи граничних станів, якщо це не обмежує нормального функціонування споруди
Стан 4	Обмежено працездатний	Можливе часткове порушення вимог першої групи граничних станів. Порушуються вимоги другої групи граничних станів. Споруда експлуатується в обмеженому режимі і вимагає спеціального контролю за станом її елементів

Стан 5	Непрацездатний	Елемент не відповідає вимогам першої групи граничних станів і з'ясовується неможливість їх задоволення.
--------	----------------	---

1.2 Види мостів та умови їх функціонування

Як правило, мости складаються з прогонових конструкцій і опор. Прогонові конструкції служать для прийняття навантажень і передачі їх опорам, на них може розташовуватися проїжджа частина, пішохідний перехід, трубопроводтощо. Опори переносять навантаження з прогонових конструкцій на основу моста.

Прогонові конструкції складаються з тримальних конструкцій: балок, ферм, діафрагм (поперечних балок) і власне плити проїжджої частини. Статична схема прогонових конструкцій може бути арковою, балковою, рамною, вантовою або комбінованою; вона визначає тип моста за конструкцією. Зазвичай прогонові конструкції прямолінійні, проте у разі необхідності (наприклад, при будівництві естакад і дорожніх розв'язок) їм надають складну форму: спіралеподібні, кільцеві тощо. Форми опор також можуть бути найрізноманітнішими. Проміжні опори називаються биками, берегові — устоями. Останні служать для з'єднання моста з підхідними насипами [7].

Матеріалами для спорудження мостів слугують метал (сталь і алюмінієві сплави), залізобетон, бетон, природний камінь, дерево, мотузки.

Мости можуть бути самих різних форм і розмірів. Існує кілька класифікацій цих споруд. Самими старовинними типами мостів є арочні і висячі. Але крім них є й інші види мостів, найпоширенішими серед них є балочні мости, балочної-консольні, вантові.

Арочні мости мають самі різні форми і не завжди напівкруглі. Пологу форму вперше вигадали у Китаї, і цей винахід став справжнім переворотом у

цій галузі будівництва. Перші мости були споруджені ще в античності, а в часи Римської імперії мостобудування отримало широке поширення. Деякі мости, побудовані в ті часи, збереглися до сьогоднішніх днів. Сьогодні можна виділити близько шести категорій класифікації цих споруд. Вони відрізняються за своїм призначенням. Звичайно, в першу чергу мости діляться на:

- автодорожні (призначені для проїзду автомобілів і проходу пішоходів);
- міські (призначені для всіх видів міського транспорту і пішоходів);
- залізничні (для поїздів);
- пішохідні та спеціальні (для кабелів, труб і так далі).

Крім цього мости можуть бути оснащені різною кількістю смуг для руху (1, 2, 4, 8 смуг).

Окремо варто сказати про види мостових опор. Вони можуть бути жорсткими або плавучими.

Міст, побудований на жорстких опорах, весь тиск від прогонових будов передає на фундамент і на ту поверхню, на якій він стоїть. Якщо ж говорити про плавучі опори, то ваговий тиск передається на воду. Окремо варто сказати про типи прогонової будови мостів. Вони можуть бути нерухомими або розвідними. Нерозвідні мости дозволяють не зупиняти рух транспорту і пішоходів, а розвідні мости дозволяють зменшити висоту моста.

Варто згадати і про довжину мостів. Вони можуть бути малими, тоді їх довжина не перевищує двадцяти п'яти метрів, середніми, з довжиною до ста метрів і великими з довжиною більше ста метрів. Ще однією особливістю конструкції, по якій можна відрізнити види мостів є статична схема основних несучих конструкцій будов прольотів. Мости можуть бути розпірними, балковими і комбінованими [8]. У кожної системи можна знайти і достоїнства, і недоліки.

За призначенням мости поділяються на:

- залізничні;
- автомобільні;

- метромости;
- пішохідні;
- водні шляхопроводи (мости для кораблів з низькою ватерлінією);
- комбіновані.

За матеріалом основних конструкцій:

- кам'яні;
- залізобетонні;
- сталеві;
- стале залізобетонні;
- дерев'яні.

За видом:

- мости;
- шляхороводи;
- віадуки;
- естакади;
- розвідні мости;
- наплавні мости;
- мости-транспортери.

За довжиною:

- малі (до 25 м);
- середні (25-100 м);
- великі (понад 100 м).

За терміном служби:

- тимчасові (до 10 років);
- постійні.

Балкові системи - найпростіший вид мостів. Призначені для перекриття невеликих прогонів. Прогонові будови - балки, що перекривають відстань між опорами. Основна відмінна риса балкової системи полягає в тому, що з прогонових будов на опори передаються тільки вертикальні навантаження, відсутні горизонтальні.

Розрізна система — складається з ряду балок, причому одна балка перекриває один прогін. Система статично визначена і може застосовуватися за будь-яких типів ґрунтів. Недоліки: велика кількість деформаційних швів і обов'язкова наявність двох опорних частин на кожній проміжній опорі.

Нерозрізна система — одна балка прогонової будови перекриває кілька прогонів або одразу все. Таким чином, прогонова будова нерозрізної системи розраховується як багатоопорна статично невизначена балка з використанням методу сил, методу переміщень або інших методів розрахунку статично невизначених систем, застосовуваних у будівельній механіці. Нерозрізна система відрізняється меншою, ніж у розрізної, кількістю деформаційних швів і меншою будівельною висотою. Недолік такої системи — чутливість до ґрунтів.

Консольна система — складається з двох типів балок. Одні балки спираються на дві опори і мають консольні звиси. Інші балки називаються підвісними, оскільки спираються на сусідні балки. З'єднання балок здійснюється за допомогою шарнірів. Перевагою консольної системи є її статична визначеність, а отже, легкість розрахунку і нечутливість до ґрунтів. До недоліків системи можна віднести велику кількість і складність деформаційних швів шарнірного типу, а також порушення комфортності проїзду в зоні шарнірів. В наш час мости такої системи споруджуються рідко.

Температурно-нерозрізна система складається з двоопорних балок, об'єднаних в ланцюг за допомогою верхньої сполучної плити. Під дією вертикальних навантажень така система працює як розрізна, а під дією горизонтальних — як нерозрізна. Її перевагою є менша кількість деформаційних швів, а недоліком — обов'язкова наявність двох опорних частин на кожній проміжній опорі.

Розпірні системи - відрізняються від балкових тим, що навантаження, які передаються з прогонових конструкцій опорі, мають не тільки вертикальну, а й горизонтальну складову, звану в будівельній механіці

розпором. Виділяють кілька різновидів розпірних систем, які досить сильно відрізняються один від одного.

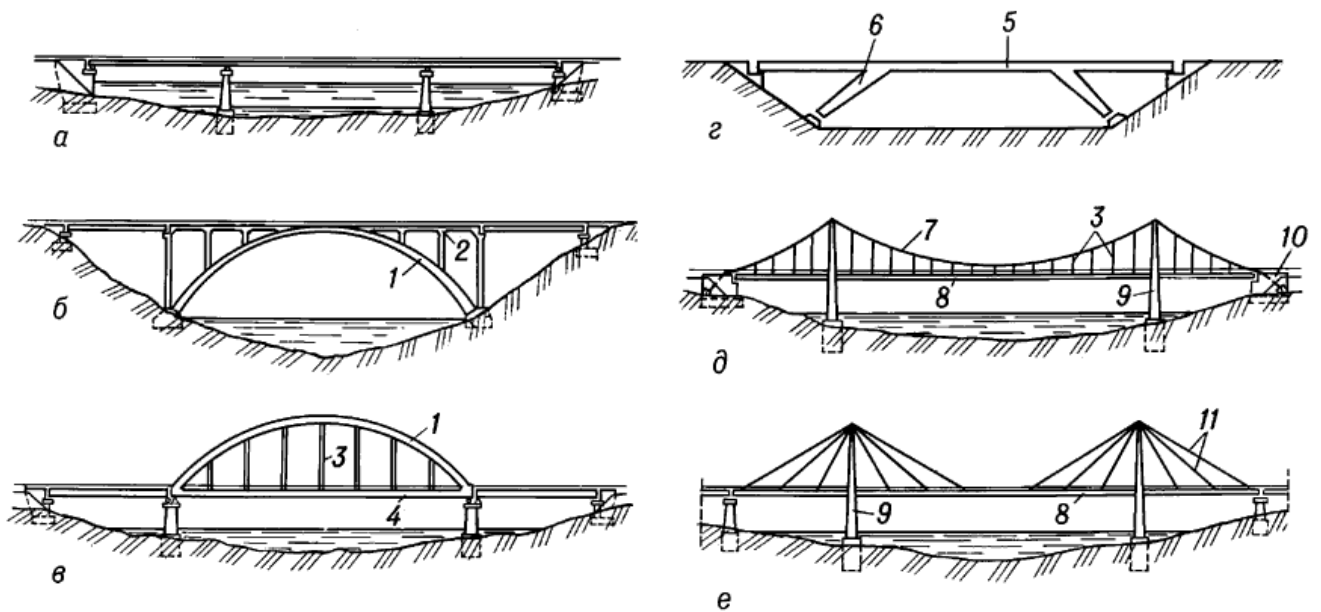
Рамна система — складається з рам, стійки яких виконують роль опор, а ригелі — роль прогонових конструкцій. За формою рами можуть бути Т-подібними, П-подібними, а також мати дві похилі стійки та консольні звиси (спеціальної назви не мають). Перевагами рамної системи є невелика будівельна висота і збільшений в порівнянні з балковими системами підмостовий простір. Все це робить рамні конструкції зручними для шляхопроводів і естакад. Також дана система може застосовуватися в гірських умовах через те, що там в силу особливостей рельєфу не можна знизити рівень проїзду. Недоліками рамної системи є складність будівництва і чутливість до ґрунтів. Такі системи в наш час рідко застосовують через високу вартість і специфічність.

Підвісний міст — міст, в якому основна тримальна конструкція виконана з гнучких елементів (канатів, ланцюгів тощо), що працюють на розтяг, а проїжджа частина підвішена. Підвісними є усі найбільші за довжиною і висотою прогони мостів світу.

Вантовий міст — різновид висячих мостів: роль основної тримальної конструкції виконує вантова ферма, виконана з прямолінійних сталевих канатів. Ванти прикріплені до пілонів — високих стійок, що встановлюється безпосередньо на опорах. Пілони в основному розташовуються вертикально, але не виключене й похиле їх розташування. До вантів кріпиться балка жорсткості, на якій розташовується мостове полотно. Ванти розташовуються під кутом нахилу до горизонталі не менше 30 градусів, інакше в них виникають великі зусилля і жорсткість помітно зменшується. Балку жорсткості краще виконувати з коробчатого перетину, оскільки це поліпшує її роботу на кручення від тимчасових навантажень і від дії вітру. Найчастіше вантова система застосовується при перекритті глибоких рік і в міських умовах.

Арковий міст. Основними тримальними конструкціями є арки або склепіння. Арка — криволінійний брус, у якого поперечний розмір менше висоти. Склепіння — криволінійний брус, у якого ширина перерізу значно більше висоти. Аркові мости можуть бути з їздою поверху, понизу та посередині. Опори аркових мостів завжди масивні, оскільки повинні бути розраховані і на сприйняття розпору. При великих прогонах арки завжди економічніші балкових конструкцій, але тільки стосовно прогонових конструкцій. Аркові мости характерні для гірських умов, оскільки дозволяють перекрити більший прогін, ніж балки, а в умовах гірського рельєфу спорудження додаткових опор не виправдане. Також специфічна область застосування аркових мостів обумовлена тим, що вони вимагають великого підмостового простору, особливо з їздою поверху, що призводить до подорожчання й ускладнення будівництва насипних підходів, які можуть сягати висоти 20 м. Крім того зростає ймовірність зсуву на таких насипах у початковий період їх експлуатації. Часто аркові мости будують в міських умовах з міркувань краси [9].

Понтонний міст або наплавний міст — тимчасові мости на плавучих опорах. Основні схеми мостових конструкцій зображені на рисунку 1.1.



а - балкова; б - аркова; в - комбінована; г - рамна; д - висяча; е - вантова; 1 - арка; 2 – над аркова надбудова ; 3 - підвіска; 4 - затяжка; 5 - ригель; 6 – похила стійка;

7 - кабель; 8 - балка жорсткості; 9 - пілон; 10 анкерна опора; 11 – ванти.

Рисунок 1.1 Статичні схеми мостів

Іноді окремо виділяють також горбаті мости, які відрізняються своєю формою, - вони істотно вигнуті вгору.

1.3 Нормативні вимоги до надійності і безпеки мостових конструкцій

Міст та мостовий перехід розглядаються як система з семи груп конструктивних елементів [6]:

- елементи проїзної частини;
- елементи прогонової будови;
- опори та опорні частини;
- фундаменти;
- підмостове русло;
- регуляційні споруди;
- підходи.

Основою для класифікації елементів моста є:

- первинна технічна документація моста;
- дані експлуатаційної документації;
- аналіз історії експлуатації;
- дані детального обстеження всієї споруди та її елементів;
- визначення реальної міцності матеріалів елементів споруди;
- виконання перевірочних розрахунків вантажопідйомності;
- вирахування реальної характеристики безпеки елементів;
- перевірка моста, за необхідності, випробувальним навантаженням.

Кожному з експлуатаційних станів відповідає надійність, яка визначена у відповідності з моделлю деградації.

Таблиця 1.2 Верхні значення надійності елементів

Експлуатаційний стан	Назва експлуатаційного стану	Надійність за першою групою граничних танів, P_i	Характеристика безпеки β_i
1	2	3	4
Стан 1	Справний	0,999844	3,80
Стан 2	Обмежено справний	0,998363	2,95

Продовження таблиці 1.2

Експлуатаційний стан	Назва експлуатаційного стану	Надійність за першою групою граничних танів, P_i	Характеристика безпеки β_i
1	2	3	4
Стан 3	Працездатний	0,992461	2,43
Стан 4	Обмежено працездатний	0,979771	2,05
Стан 5	Непрацездатний	0,958351	1,74

У всіх випадках класифікації, коли є сумніви стосовно того, до якого із двох сусідніх експлуатаційних станів належить віднести елемент, рекомендується прийняти нижчий. Для елементів прогонових будов у такому випадку остаточне рішення приймається за результатами визначення їх вантажопідйомності або реальної характеристики безпеки.

Для класифікації стану прогонових будов мостів аналітично вираховується реальна на час обстеження характеристика безпеки β .

Для кожного з експлуатаційних станів визначено необхідні експлуатаційні заходи та рівень зносу [6].

Таблиця 1.3 Деградація елементів та експлуатаційні заходи

Експлуатаційний стан	Характеристика безпеки β	Знос елемента, %	Експлуатаційні заходи
1	2	3	4
Стан 1 Справний	3,8	0-3	Ведуться планові обстеження та догляд
Стан 2	3,0	3-8	Ведуться планові обстеження, догляд та

Обмежено справний			поточні ремонти без обмеження руху
Стан 3 Працездатний	2,4	8-27	Ведуться планові обстеження, скорочуються терміни між періодичними оглядами, виконуються поточні ремонти. За необхідності, обмежується швидкість руху

Продовження таблиці 1.3

Експлуатаційний стан	Характеристика безпеки β	Знос елемента, %	Експлуатаційні заходи
1	2	3	4
Стан 4 Обмежено працездатний	2,1	27-42	Ведуться обстеження за спеціальним графіком, виконується капітальний ремонт. Відповідно до дефектів конструкцій обмежується рух транспортних засобів за вагою, швидкістю та габаритними параметрами. За необхідності, розробляються спеціальні заходи забезпечення безаварійної експлуатації мосту
Стан 5 Непрацездатний	1,7	42-65	Ведеться постійний нагляд та контроль за виконанням обмежень руху з залученням спеціалізованої організації. Терміново вирішується питання про реконструкцію споруди або ж про її закриття.

1.4 Методи і засоби визначення технічного стану

Оцінювання і прогнозування технічного стану мостів та їх елементів визначається на основі даних нагляду і спостережень, результатів обстежень та/або випробувань [5].

Алгоритм оцінки і прогнозування технічного стану складається з таких основних кроків:

- збір вихідних даних для оцінки і прогнозування технічного стану елементів моста;
- визначення стану елементів моста за класифікаційними таблицями експлуатаційних станів;
- уточнення стану елементів за обчисленням вантажопідйомності;
- визначення стану елементів за реальною характеристикою безпеки елементів;
- призначення експлуатаційних заходів;
- прогнозування строку безпечної експлуатації елементів (визначення залишкового ресурсу);
- оцінювання технічного стану моста в цілому для рангування споруди за потребою експлуатаційних заходів.

Визначення стану прогонових будов мостів за результатами обчислення їх вантажопідйомності.

Визначення вантажопідйомності за наведеним нижче алгоритмом є обов'язковою регламентною процедурою і служить цілям уточнення класифікації експлуатаційного стану елемента.

Вантажопідйомність визначається відносно нормативних тимчасових рухомих навантажень:

- а) колон навантажень Н-30, що встановлюються на лінії впливу зусиль та в поперечному перерізі згідно з нормами СН 200-62;
- б) автомобільного навантаження за схемою АК згідно з чинними нормами навантаження класу К згідно з ДБН В.1.2-15;
- в) одиничного колісного транспортного засобу НК-80 або НК-100 згідно з ДБН В.1.2-15.

Розрахунок вантажопідйомності прогонових будов виконується на основі реальних розмірів елементів споруди, механічних характеристик матеріалів та опису наявних дефектів, зафіксованих у результаті обстеження.

Вантажопідйомність прогонових будов встановлюють порівнянням зусиль у перерізах елементів від нормативних тимчасових рухомих

навантажень з граничними значеннями, вираженими у відповідних еквівалентних навантаженнях. Має задовольнятися нерівність:

$$q_e \geq p_e, \quad (1.1)$$

де q_e – граничне значення еквівалентного навантаження;

p_e – еквівалентне навантаження від нормативних тимчасових рухомих навантажень.

Якщо нерівність (1) не задовольняється тобто $p_e > q_e$, то обчислюється величина зниження вантажопідйомності δ (у відсотках), за якою класифікується експлуатаційний стан.

Значення δ вираховують за формулою:

$$\delta = \left(1 - \frac{q_e}{p_e}\right) 100 \quad (1.2)$$

Значення граничного еквівалентного навантаження вираховують за формулами:

За першою групою граничних станів:

$$q_e = \frac{S_{\text{ГР}}^{\text{I}} - \sum_{i=1}^k y_i g_i \omega}{\Omega_{\text{I}}} \quad (1.3)$$

За другою групою граничних станів:

$$q_e = \frac{S_{\text{ГР}}^{\text{II}} - \sum_{i=1}^k g_i \omega}{\Omega_{\text{II}}}, \quad (1.4)$$

де $S_{\text{ГР}}^{\text{I}}, S_{\text{ГР}}^{\text{II}}$ – граничне зусилля в елементі, обчислене за I та II групами граничних станів відповідно;

g_i – нормативне значення інтенсивності постійного навантаження;

y_i – коефіцієнт надійності за постійним навантаженням;

ω – сумарна площа ділянок ліній впливу;

$\Omega_{\text{I}}, \Omega_{\text{II}}$ – розрахункові площі ліній впливу відповідних зусиль;

k – кількість видів постійного навантаження;

Для інтегральної оцінки технічного стану споруди введено два показники: експлуатаційний стан елементів моста та формалізовану експертну оцінку споруди в цілому.

Експлуатаційна оцінка моста в цілому є узагальненою характеристикою експлуатаційної придатності за станом всіх його елементів.

Класифікується експлуатаційний стан моста як найнижчий із показників експлуатаційного стану його елементів:

- прогонових будов;
- опор;
- фундаментів.

Кількісним показником експертного визначення технічного стану споруди (мостового переходу в цілому) є формалізована експертна оцінка (рейтинг), що служить:

- для ранжування мостів у рамках певної дорожньої мережі за потреби ремонту або реконструкції;
- для планування на цій основі видатків на ремонт, реконструкцію або будівництво нового моста.

Експертна експлуатаційна оцінка служить у системі управління мостами показником потреби виконання експлуатаційних заходів:

- встановлення режиму утримання споруди;
- встановлення термінів та видів ремонту;
- призначення параметрів посилення, поширення;
- прийняття рішення щодо необхідності та доцільності заміни, реконструкції або капітального ремонту.

Експертна експлуатаційна оцінка технічного стану споруди визначається за шкалою безрозмірних коефіцієнтів E у 100 балів.

Експертна експлуатаційна оцінка є середньозваженим значенням визначення експлуатаційного стану груп конструктивних елементів споруди:

- елементи проїзної частини;
- елементи прогонової будови;
- опори та опорні частини;
- фундаменти;

- підмостове русло;
- регуляційні споруди;
- підходи.

Експлуатаційний стан групи елементів приймається за станом найбільш слабкого елемента в групі.

Обчислюється експертна оцінка E технічного стану споруди за формулою:

$$E = \frac{80(5 - \sum_{i=1}^7 \alpha_i D_i)}{4} + 20, \quad (1.5)$$

де D_i – номер експлуатаційного стану групи конструктивних елементів споруди згідно з класифікаційною таблицею 1.1.

α_i - коефіцієнти впливу стану i -го елемента на загальний стан споруди (нормалізовані коефіцієнти ваги), $i = 1, 2, \dots, 7$.

Значення вагових коефіцієнтів наведено в таблиці 1.4.

У залежності від рейтингу споруди, таблицею 1.5 визначається потреба у виконанні експлуатаційних заходів.

Таблиця 1.4 Вагові коефіцієнти у формулі експертної оцінки технічного стану споруди

Елемент		Проїзна частина	Прогонова будова	Опора та опорні частини	Фундамент	Підмостове русло	Регуляційна споруда	Підходи
1		2	3	4	5	6	7	8
Коефіцієнт впливу α_i	Міст через водну перешкоду	0,06	0,38	0,19	0,13	0,11	0,08	0,05
	Міст без регуляційних споруд	0,07	0,41	0,21	0,14	0,12	—	0,05

	Шляхопровід естакада, віадук	0,07	0,46	0,23	0,27	–	–	0,07
--	------------------------------------	------	------	------	------	---	---	------

Таблиця 1.5 Потреба у виконанні експлуатаційних заходів

Експлуатаційний стан	Рейтинг балів, від-до	Експлуатаційні заходи
1	2	3
Стан 1 Справний	100-95	Ведуться планові обстеження та догляд
Стан 2 Обмежено справний	94-80	Ведуться планові обстеження, догляд та поточні ремонти без обмеження руху
Стан 3 Працездатний	79-60	Ведуться планові обстеження, скорочуються терміни між періодичними оглядами, виконуються поточні ремонти. За необхідності, обмежується швидкість руху
Стан 4 Обмежено працездатний	59-40	Ведуться обстеження за спеціальним графіком, виконується капітальний ремонт. Відповідно до дефектів конструкцій обмежується рух транспортних засобів за вагою, швидкістю та габаритними параметрами.
Стан 5 Непрацездатний	39-20	Ведеться постійний нагляд та контроль за виконанням обмежень руху з залученням спеціалізованої організації.

1.5 Аналіз типових інформаційних технологій діагностики технічного стану

Існують наступні інформаційні технології діагностики технічного стану будівель: експертні системи, нечіткі системи, нечіткі нейронні мережі або гібридні мережі, інформаційно-довідкові системи, інтелектуальні системи підтримки прийняття рішень та інші [10].

На рисунку 1.2 представлена структурна схема інформаційних технологій діагностики технічного стану будівель.

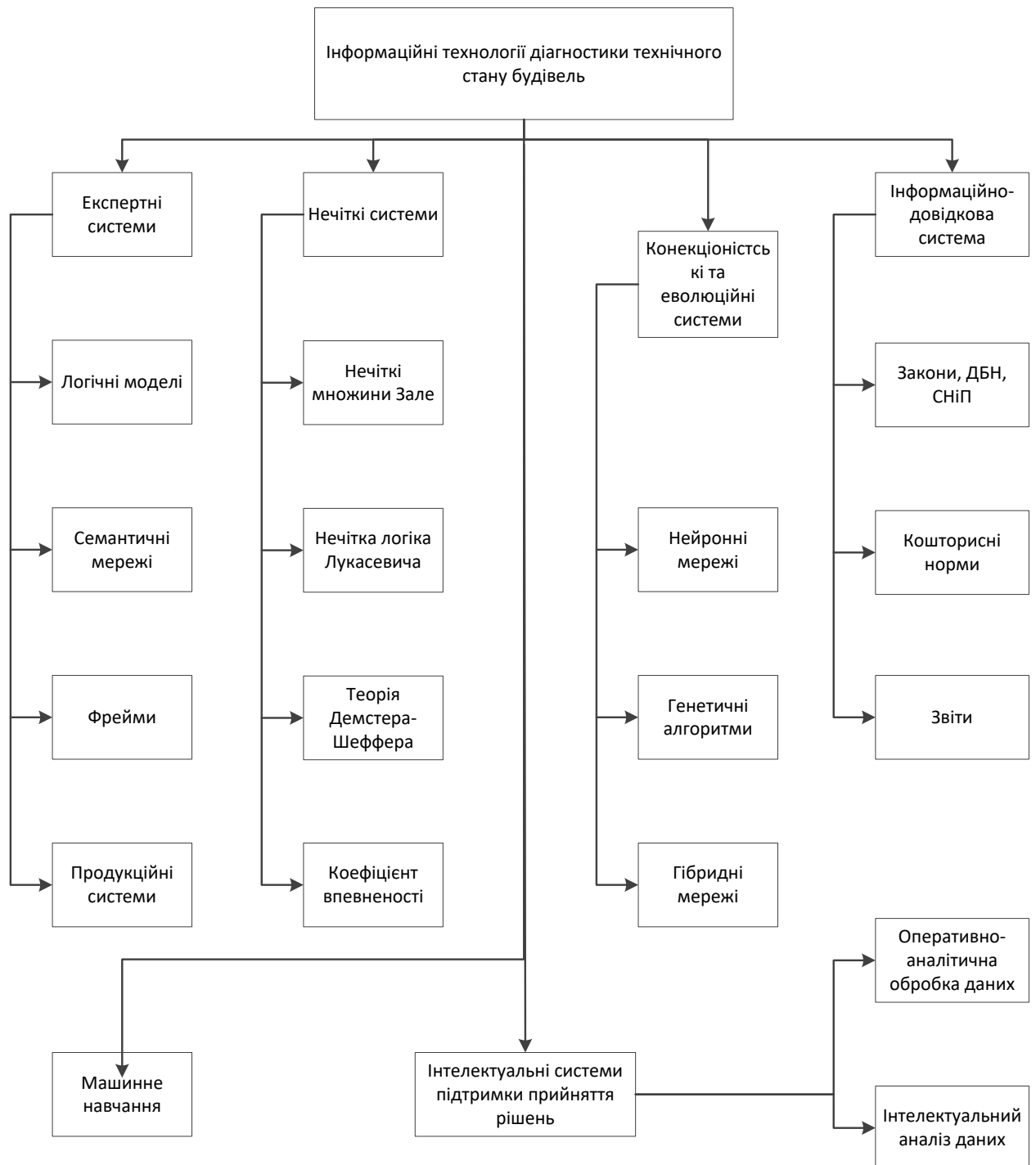


Рисунок 1.2 Структурна схема інформаційних технологій діагностики технічного стану будівель

Кожна з цих технологій має свої особливості. Переваги та недоліки представлених технологій наведено в таблиці 1.6.

Таблиця 1.6 Порівняльна характеристика методів інформаційних технологій

Метод	Переваги	Недоліки
1	2	3
Експертні системи	Створення експертних методів та моделей обстеження техстану будівель; проведення досліджень інформаційних та експертних систем; розробка бази знань та нечітких правил; дослідження ланцюгів логічного виведення для діагно-стики техстану	Складність представлення знань експерта щодо обстеження тех.-нічного стану будівель; складність організації навчання на досвіді спеціаліста
Нечіткі системи	Дозволяють вирішувати задачі інформаційної технології для підтримки діагностики техстану будівель, розробка бази знань для будівель, розробка системи не чіткого виведення; дослідження та реалізація на основі апарату нечіткої логіки моделей діагностики техстану будівель.	Суб'єктивність обстеження технічного стану будівель; не дозволяє навчатися
Нечіткі нейронні мережі або гібридні мережі	Можливість отримання інформації щодо обстеження техстану будівель у формі деякого прогнозу; побудова нейронних мереж обстеження тех.-стану будівель здійснюється за допомогою їх навчання на основі наявної і доступної інформації; розробляти і	Уявлення знань щодо обстеження технічного стану будівель в спеціальному вигляді, який може суттєво відрізнятися від можливої змістовної інтерпретації існуючих взаємозв'язків і відносин

	представляти моделі систем у формі правил нечітких продукцій	
--	--	--

Продовження таблиці 1.6

Метод	Переваги	Недоліки
1	2	3
Генетичні алгоритми	Використовують для пошуку оптимума декілька точок одночасно, а не переходять від точки до точки, що дозволяє уникнути небезпеки попадання у локальний оптимум; не потрібно в процесі роботи ніякої додаткової інформації, що збільшує швидкість роботи алгоритму; використовують детерміновані і ймовірнісні правила	Не працюють з символічною інформацією
Інформаційно - довідкова система	Наявність законів, нормативних документів, кошторисних норм, СНіПів, ДБНів, положень	Відсутність розробки і застосування інформаційної системи, що дозволяє в автоматичному режимі узагальнювати і аналізувати інформацію; вибирати оптимальну технологію за наявності баз даних та способах їх виконання
Машинне навчання	Рішення задач навчання базується або на лабіринтній моделі, або передбачає пошук напрямку руху в лабіринті можливих варіантів, або встановлювання асоціативних зв'язків у нейроподібних структурах	Проблеми узагальнення накопичення досвіду вирішення задач і застосування цих проблем у вирішення нових задач

Продовження таблиці 1.6

Метод	Переваги	Недоліки
1	2	3
Інтелектуальні системи підтримки прийняття рішень	Використання можливостей інформаційних систем обстеження технічного стану будівель; сучасні бази даних включають до свого складу цілий ряд механізмів і технологій, які підвищують інтелектуальні можливості; інтелектуальні інформаційні системи обстеження технічного стану	Складність представлення різнобічної інформації

В результаті проведеного аналізу інформаційних технологій, для подальшої роботи було обрано інформаційно-довідкову систему, яка вміщує всі нормативні документи, необхідні для оцінки технічного стану будівель та споруд.

1.6 Постановка задачі

На даний час не існує спеціалізованої системи оцінки технічного стану будівельних конструкцій, яка узагальнює позитивні фактори кожної інформаційної технології. Об'єднання декількох локальних систем в єдину велику систему дозволить оперативно вирішувати задачі діагностики технічного стану. Розробка такої системи забезпечить зниження трудомісткості та вартості обстеження конструкцій. Водночас вона забезпечить більш якісний результат проведених оцінок технічного стану будівельних об'єктів. Метою роботи є розробка однієї з локальних підсистем, а саме – інформаційно-довідкової складової.

Результатом створення системи повинно бути досягнення таких *цілей*:

1. Розробити систему взаємодії з користувачем.
2. Забезпечити можливість вибору стандартних конструктивних елементів з бази даних.
3. Забезпечити можливість вибору типових дефектів для кожного з конструктивних елементів.
4. Розробити систему моделювання конструкції.
5. Розробити систему оцінки технічного стану.
6. Розробити систему генерування звітів про технічний стан конструкції, що обстежувалась.
7. Забезпечити занесення результатів роботи до бази даних.

Система повинна складатися з наступних модулів:

- модуль введення даних;
- модуль взаємодії з користувачем;
- модуль управління базою даних;
- модуль моделювання конструкції;
- модуль оцінки технічного стану;
- модуль генерації звітів;
- модуль збереження даних;

В системі використовуються такі бази даних:

- БД типових конструктивних елементів;
- БД ДБН України;
- БД дефектів;
- БД звітів;

Для технічного обслуговування системи достатньо одного працівника з незначною попередньою підготовкою.

Вимоги до надійності та безпеки:

- забезпечення резервного копіювання даних;
- надійна автентифікація та авторизація користувачів.

Вимоги до ергономіки і технічної естетики:

- забезпечення доступу через інтерфейс для всіх користувачів системи;
- інтуїтивність та ергономічність дизайну користувацького інтерфейсу.

2 КОНЦЕПТУАЛЬНЕ ТА ФУНКЦІОНАЛЬНЕ ПРОЕКТУВАННЯ ТЕХНОЛОГІЇ

2.1 Дерево цілей системи

Слід зауважити, що на практиці, як правило, існує кілька цілей і тому важливо, окрім визначення головної мети, не упустити деякі з суттєвих серед інших. Для цього застосовують метод побудови дерева цілей, що був запропонований ще 1957 року групою американських учених та успішно використаний в ряді військових та промислових програм у США, а нині є повсякденним інструментом практично будь-якого сучасного менеджера [11].

Під деревом цілей розуміють ієрархічну деревоподібну структуру, яка отримується поділом загальної цілі на підцілі, а їх, у свою чергу, — на детальніші складові — нові підцілі, функції тощо. Якщо всі ці елементи зобразити графічно, то одержимо «дерево цілей», повернуте «короною» донизу. При цьому головну ціль розміщують на найвищому рівні.

Перевагою цього методу є те, що він уможливорює поділ складного завдання, яке важко формалізувати, на сукупність простіших завдань, для розв'язання яких існують перевірені прийоми і методи. Послідовний поділ розв'язуваної проблеми на її частини — підпроблеми — є важливим етапом системного аналізу проблем. Поділ продовжують доти, доки не отримають прості, звичні, очевидні завдання, які можна розв'язати відомими методами.

Метод побудови дерева цілей являє собою один із найрозповсюдженіших та найефективніших способів аналізу слабо структурованих завдань, що стоять перед економічними об'єктами. Він допомагає знаходити найкращі шляхи та засоби вирішення існуючих проблем. Деревоподібні ієрархічні структури використовуються і при дослідженні та удосконаленні організаційних структур.

Для формалізації поставленої задачі та досягнення поставлених вимог будуюмо дерево цілей (рис. 2.1).

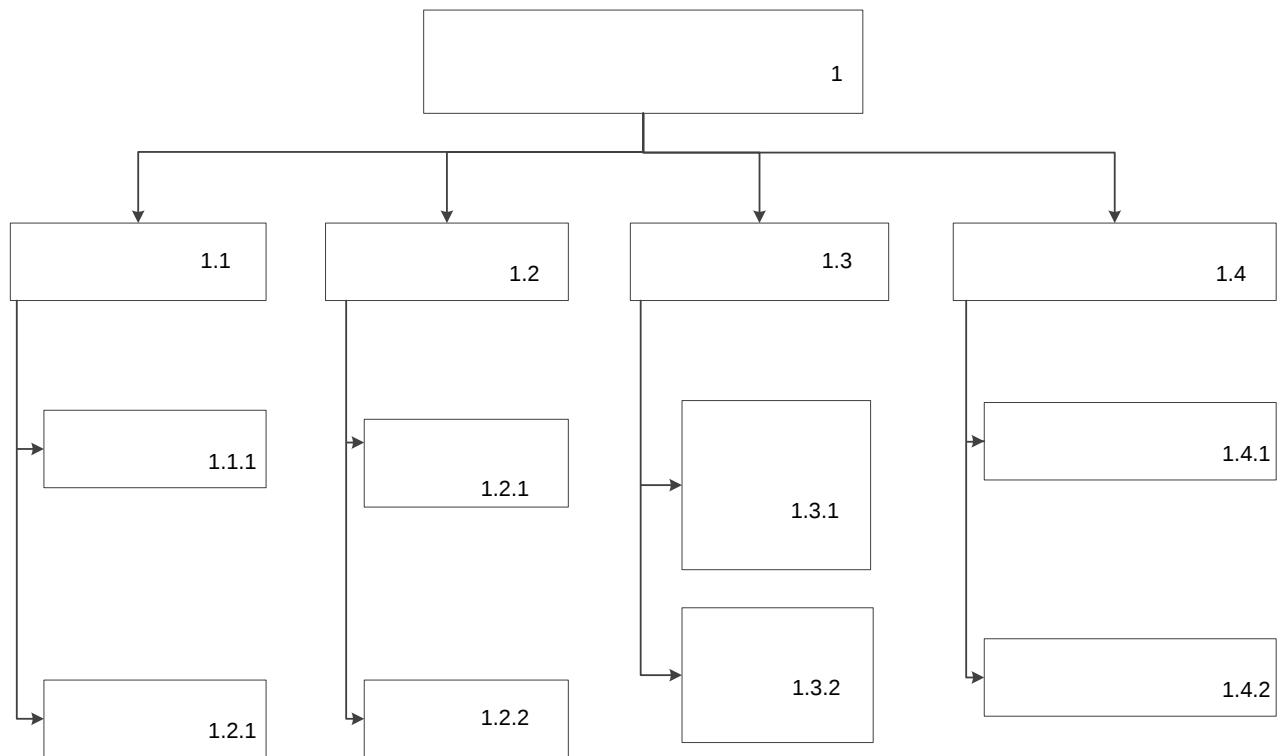


Рисунок 2.1 Дерево цілей системи

Опис вузлів дерева:

1. Розробити інформаційну технологію моделювання технічного стану будівельних конструкцій.
 - 1.1.Розробити систему введення даних.
 - 1.1.1. Забезпечити можливість вибору типових елементів мостової конструкції з БД.
 - 1.1.2. Забезпечити можливість вибору дефектів, що присутні в конструктивному елементі конструкції з БД.
 - 1.2.Розробити систему взаємодії з користувачем.
 - 1.2.1. Дозволити користувачу вказувати величину дефектів в конструктивних елементах.
 - 1.2.2. Дозволити користувачу задавати параметри кожного конструктивного елементу конструкції.
 - 1.3.Розробити систему моделювання конструкції та оцінки технічного стану.

- 1.3.1. *Забезпечити доступ до бази нормативних документів.*
- 1.3.2. *Провести моделювання та оцінку технічного стану конструкції.*
- 1.4. *Розробити систему виведення та збереження відпрацьованих даних.*
 - 1.4.1. *Згенерувати звіт відповідно до ДБН про змодельовану конструкцію.*
 - 1.4.2. *Забезпечити занесення результатів роботи системи до БД.*

2.2 Дерево функцій системи

Функції, які повинна виконувати система, в ієрархічному вигляді наведено на рисунку 2.2.

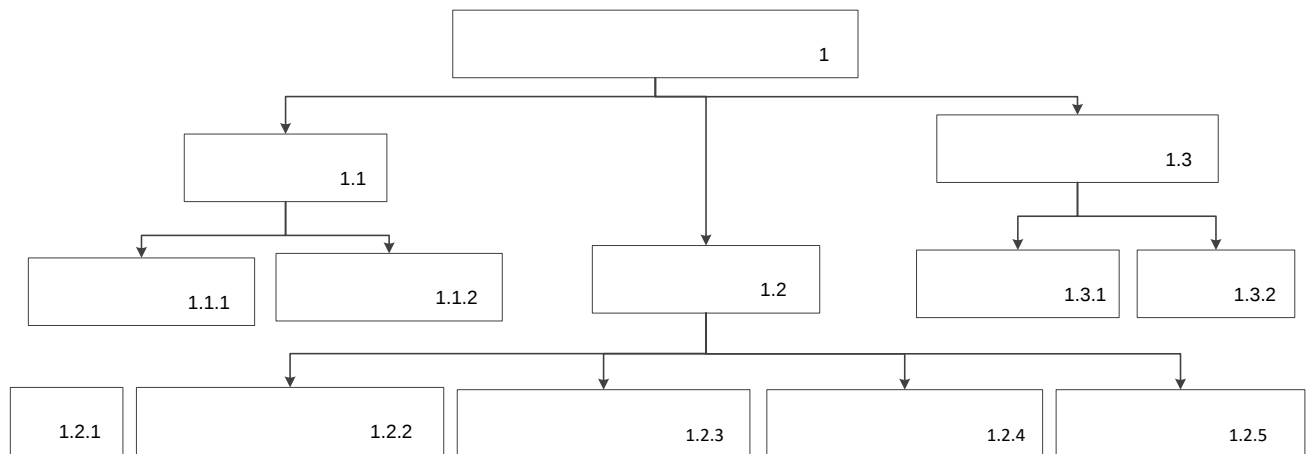


Рисунок 2.2 Дерево функцій системи

- 1. *Автоматизація визначення технічного стану будівельних конструкцій.*
 - 1.1. *Введення вхідних даних.*
 - 1.1.1. *Вибір конструктивних елементів.*
 - 1.1.2. *Вибір дефектів.*
 - 1.2. *Оцінка технічного стану.*
 - 1.2.1. *Завантаження фото.*
 - 1.2.2. *Використання алгоритму Ендрю для оцінки пошкодженої площі покриття.*
 - 1.2.3. *Порівняння результатів оцінки з нормативними значеннями.*
 - 1.2.4. *Порівняння результатів оцінки з граничними значеннями.*

1.2.5. Визначення категорії технічного стану.

1.3. Виведення та збереження вихідних даних.

1.3.1. Генерація звіту про змодельовану конструкцію відповідно до ДБН.

1.3.2. Збереження результатів до БД.

2.3 Концептуальна модель системи

Система повинна складатись з модулів, які забезпечують досягнення поставлених цілей та виконання функціональних можливостей. На рисунку 2.3 зображено концептуальну модель системи [12].

Модуль введення вхідних даних забезпечує виконання функції введення вхідних параметрів, вибору типу конструкції.

Модуль взаємодії з користувачем дозволяє користувачу обирати складові елементи конструкції з бази даних та задавати їх параметри. Також забезпечується можливість вибору дефектів, що присутні в тому чи іншому конструктивному елементі.

Модуль управління базою даних дозволяє отримати доступ до сховища даних, заносити результати роботи програми до бази даних.

Модуль моделювання конструкції – забезпечує створення моделі конструкції відповідно до обраних користувачем елементів.

Модуль оцінки технічного стану – основний модуль системи, який реалізує функцію аналізу елементів конструкції та оцінки їх технічного стану.

Модуль генерації звітів забезпечує створення звітів про оцінку змодельованої конструкції.

Модуль збереження даних забезпечує можливість занесення звітів по оцінених елементах конструкції до бази даних.

Сховище даних – сукупність баз даних для зберігання інформації про типові конструктивні елементи, нормативні вимоги до безпеки їх експлуатації, можливі дефекти, що виникають в конструкціях, категорії технічного стану та їх особливості.



Рисунок 2.3 Концептуальна модель системи

Обмін даними між складовими концептуальної моделі можна представити за допомогою діаграми потоків даних.

2.4 Діаграми потоків даних

Відображення структури системи, у якій зв'язками є інформаційні потоки, можна здійснити за допомогою діаграм потоків даних DFD. Ці діаграми використовують для аналізу та моделювання інформаційних систем з метою мінімізації потоків даних та зменшення їх об'єму, виявлення як дублювання інформації, так і дублювання шляхів її передавання. DFD відображають джерела та споживачів інформації, вид та напрямок передачі інформації, елементи накопичення та процеси перетворення, при цьому використовуються різні засоби відображення елементів (нотації)[11, 12].

Головний процес – «Автоматизація визначення технічного стану будівельних конструкцій» (рис. 2.4). Вхідні дані для виконання процесу надає користувач. В результаті роботи процесу отримуємо звіт про технічний стан змодельованої конструкції, який надається користувачу та заносяться до бази даних.

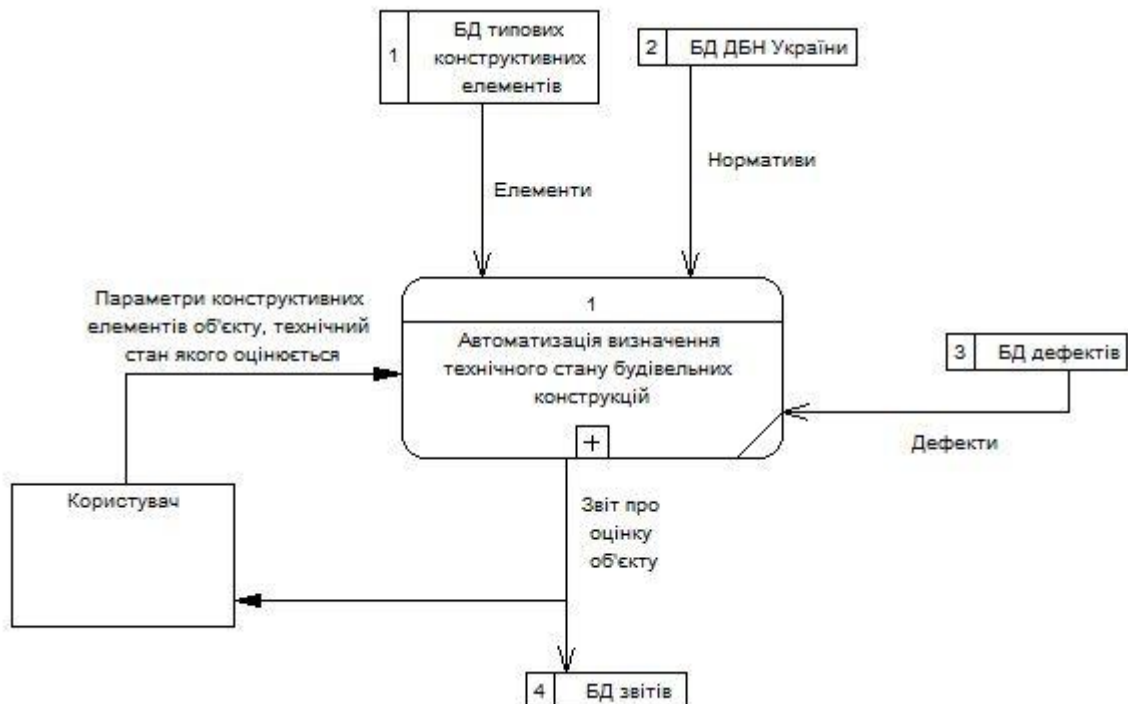


Рисунок 2.4 Процес «Автоматизація визначення технічного стану будівельних конструкцій»

Декомпозиція процесу «Автоматизація визначення технічного стану будівельних конструкцій» зображена на рисунку 2.5. З рисунка видно, що підпроцеси головного процесу повністю відповідають дереву функцій системи (рис. 2.2).

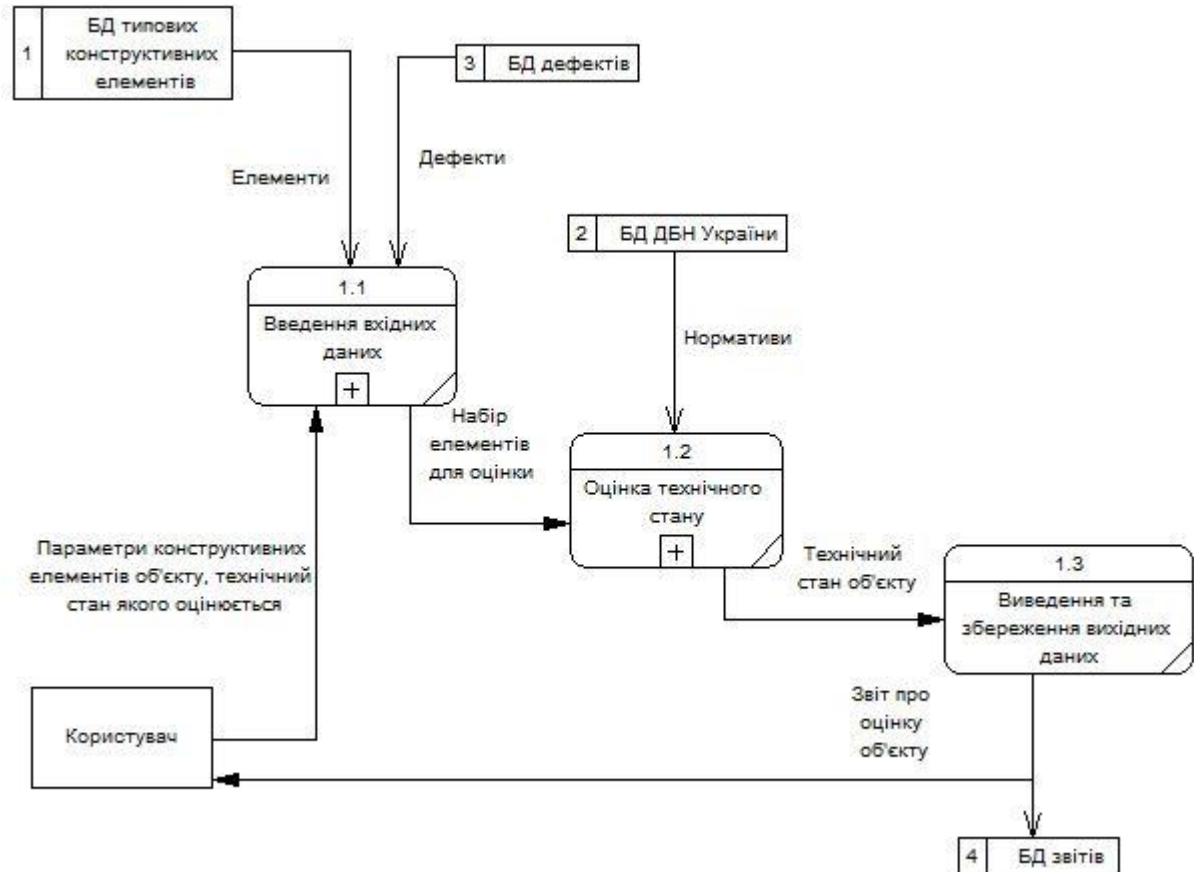


Рисунок 2.5 Декомпозиція процесу «Автоматизація визначення технічного стану будівельних конструкцій»

На рисунку 2.6 зображено декомпозицію процесу «Введення вхідних даних». В якості вхідних даних для виконання процесу виступають параметри конструктивних елементів об'єкту, технічний стан якого оцінюється, а також величина кожного з дефектів, які присутні в елементах. В результаті отримуємо набір конструктивних елементів, технічний стан яких треба оцінити, з присутніми в них дефектами

На рисунку 2.7 зображено одну з гілок дерева функцій системи (див. рис. 2.2), а саме – декомпозицію процесу «Оцінка технічного стану».

На рисунку 2.8 відображено декомпозицію процесу «Виведення та збереження вихідних даних».

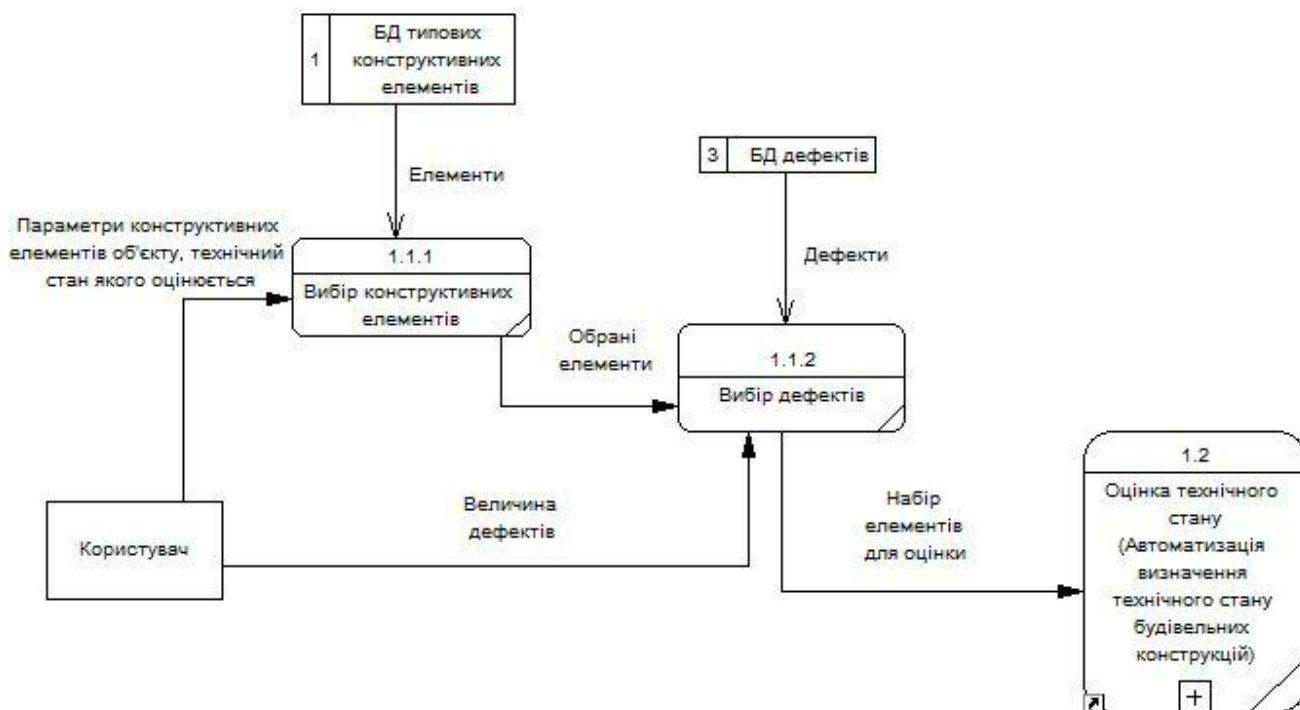


Рисунок 2.6 Декомпозиція процесу «Введення вхідних даних»

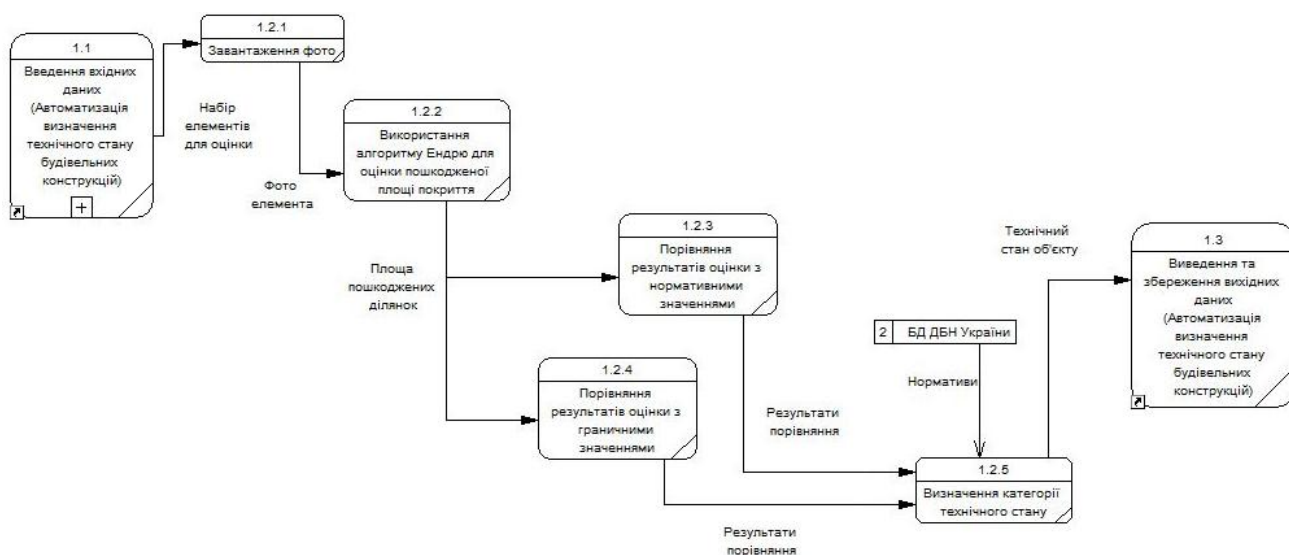


Рисунок 2.7 Декомпозиція процесу «Оцінка технічного стану»

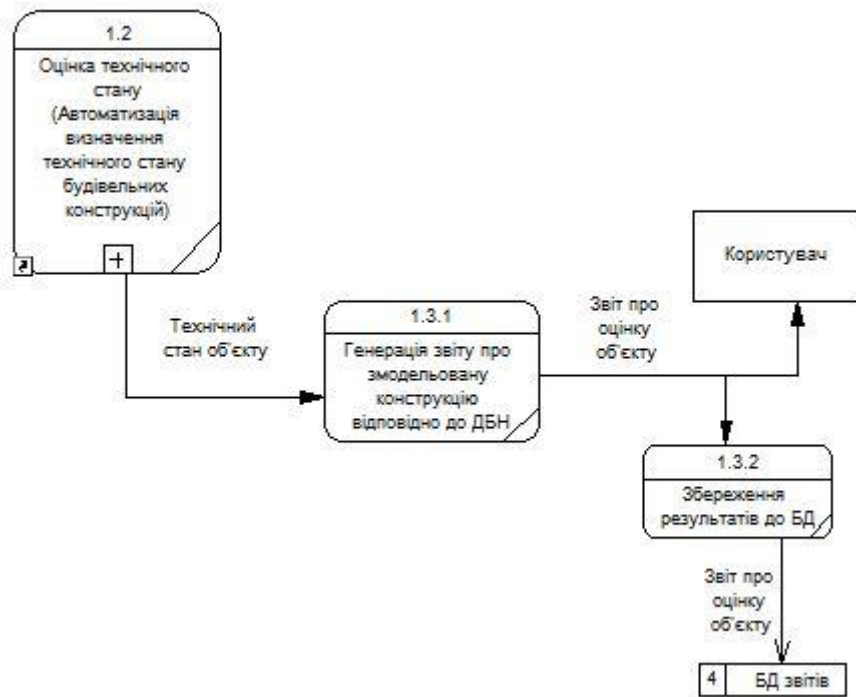


Рисунок 2.8 Декомпозиція процесу «Виведення та збереження вихідних даних»

Отже, діаграма потоків даних відображає виконання функцій, запроектованих у системі, у відповідних процесах, відповідаючи дереву (рис. 2.2). Це значить, що процеси системи призначені для досягнення поставлених цілей(див. рис. 2.1).

2.5 Розробка інтерфейсу користувача

Інтерфейс користувача — сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача; у графічних системах інтерфейс користувача реалізовується багатовіконним режимом, змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їхнім розташуванням, сортуванням елементів вікон, гнучкими налаштуваннями як самих вікон, так і окремих

їхніх елементів (файли, папки, ярлики, шрифти тощо), доступністю багатокористувацьких налаштувань.

Для забезпечення захищеності системи було прийнято рішення розробити можливість авторизації для кожного з користувачів. Схематичне зображення вікна авторизації представлено на рисунку 2.9.

Вікно авторизації користувача. У верхній частині вікна знаходиться заголовок "Application/Window-name". Нижче розташовані поля для введення "Login" та "Password", а також кнопка "Enter".

Рисунок 2.9 Вікно авторизації користувача

Після авторизації користувач отримує доступ до бази типових конструктивних елементів та дефектів рисунок 2.10.

Робоче вікно програми. У верхній частині вікна знаходиться заголовок "Application/Window-name". Нижче розташовані елементи інтерфейсу: ліве меню з чекбоксами "Element type" (Type 1, Type 2, Type 3, Type 4); центральна таблиця "Element name" з заголовком та рядками "Name 1", "Name 2", "Name 3"; правий блок з текстовим полем "E:\column1.jpg", кнопкою "Open" та кнопкою "Upload photo".

Element name
Name 1
Name 2
Name 3

Рисунок 2.10 Робоче вікно програми

На рисунку 2.11 представлено останнє вікно програми, де буде відображатись технічний стан елементів та рекомендації щодо його покращення, якщо це необхідно.

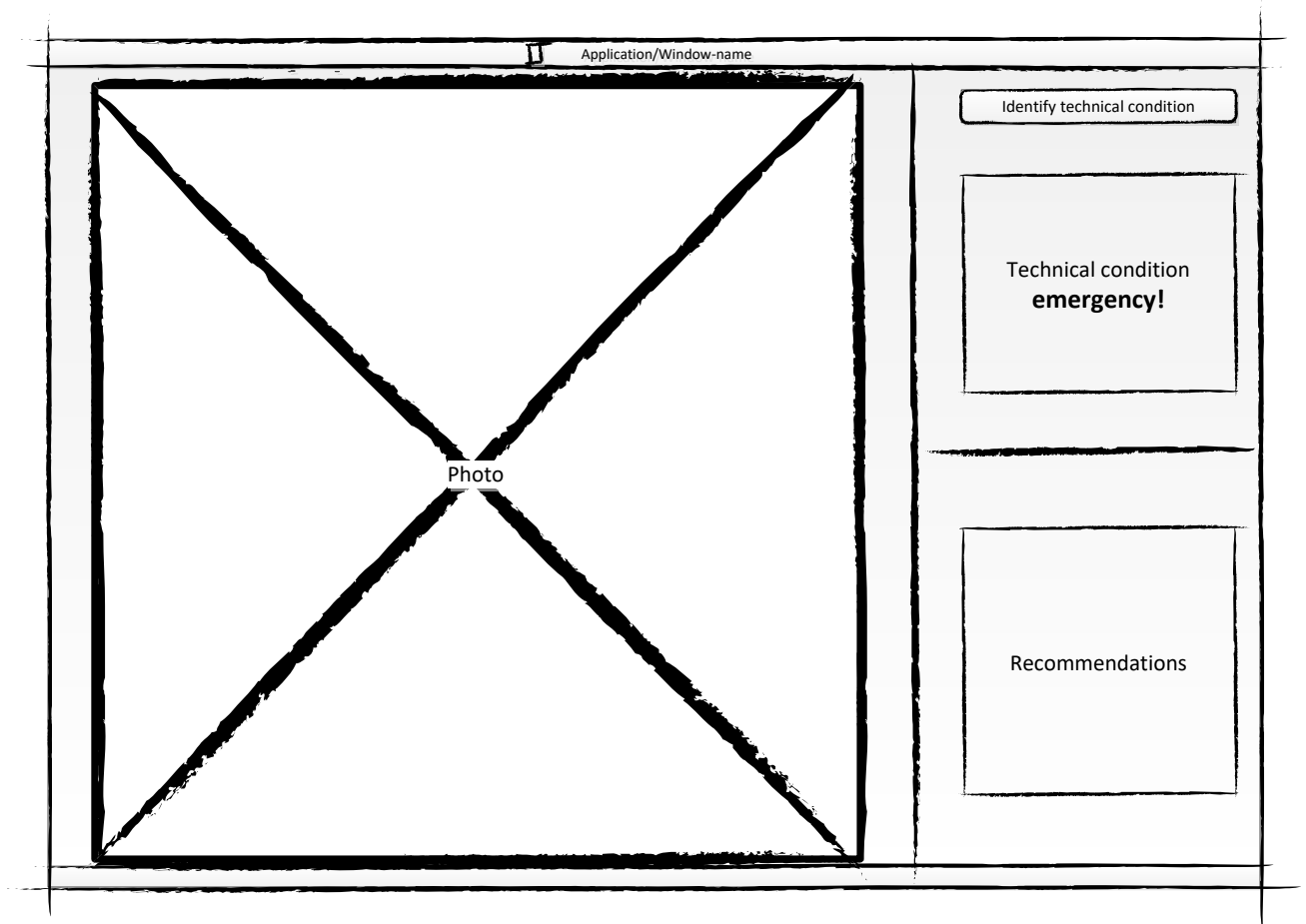


Рисунок 2.11 Вікно для визначення технічного стану

Наведений інтерфейс дуже простий у використанні, але при цьому він дозволяє захистити базу даних та саму систему від несанкціонованого доступу.

3 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ

З математичної точки зору, поставлена задача зводиться до визначення відношення площі пошкоджених зон до загальної площі поверхні конструктивного елемента, що оцінюється. Будь який дефект поверхні можна описати опуклим багатокутником. Отже необхідно обрати алгоритм побудови опуклої оболонки навколо пошкодженої зони та метод визначення площі опуклого багатокутника.

3.1 Вибір алгоритму побудови опуклої оболонки

Після локалізації дефективних зон, їх необхідно охопити оболонкою. Незважаючи на тип дефекту та його розміри, оболонка буде опуклою.

Відповідно до визначення, *опукла оболонка* - це найменша опукла множина, яка містить S [13].

Нехай в просторі E^d задано k різних точок $p_1, p_2, \dots, p_k$. Множина точок

$$p = a_1 p_1 + a_2 p_2 + \dots + a_k p_k \quad (3.1)$$
$$(a_j \in \mathbb{R}, a_j \geq 0, a_1 + a_2 + \dots + a_k = 1),$$

називається опуклою множиною, яка породжена точками $p_1, p_2, \dots, p_k$, а p називається опуклою комбінацією $p_1, p_2, \dots, p_k$.

Опуклою оболонкою $\text{conv}(L)$ підмножини L називається найменша опукла множина, яка містить L .

Щоб охарактеризувати структуру $\text{conv}(L)$ скінченної множини точок L , необхідно узагальнити поняття опуклого багатокутника та опуклого многогранника.

Поліедральною множиною в E^d називається перетин скінченної множини замкнутих півпросторів.

Поліедральна множина є опуклою, так як півпростір є опуклою множиною і перетин опуклих множин теж є опуклою множиною. В загальному випадку скінченну d - вимірну поліедральну множину називають опуклим d - політопом .

Опукла оболонка скінченної множини точок в E^d є опуклим політопом. Навпаки, кожен опуклий політоп є опуклою оболонкою деякої скінченної множини точок.

Опуклий політоп задається описом його границі, яка складається з граней. Кожна грань опуклого політопа є опуклою множиною; k - грань означає k - вимірну грань.

Позначимо границю опуклої оболонки через $CH(S)$.

Задача OO1: В E^d задано множину S , яка містить N точок. Необхідно побудувати їх опуклу оболонку.

Задача OO2: В E^d задано множину S , яка містить N точок. Необхідно визначити ті з них, які є вершинами опуклої оболонки $\text{conv}(S)$.

Нижня оцінка задачі пошуку опуклої оболонки: для знаходження опуклої оболонки множини з N точок у просторі $d \geq 2$ необхідно $\Omega(N \log N)$ операцій. Задачу сортування можна за N кроків звести до задачі пошуку опуклої оболонки.

Теорема 1: задача сортування зводиться за лінійний час до задачі побудови опуклої оболонки, і для знаходження впорядкованої опуклої оболонки з N точок на площині потрібен час $\Omega(N \log N)$ [15].

Нехай задано N додатніх дійсних чисел $x_1, x_2, \dots, x_N$. Поставимо у відповідність числу x_i точку (x_i, x_i^2) і присвоїмо їй номер i (рис 3.1).

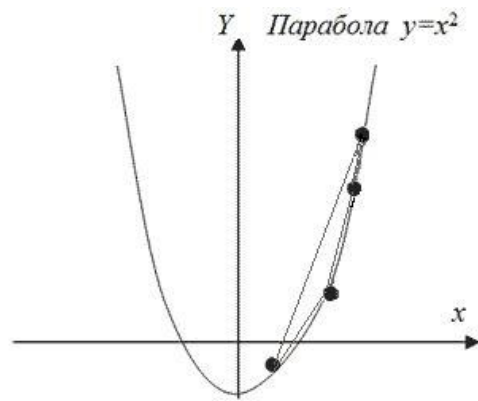


Рисунок 3.1 Ілюстрація доведення теореми 1

Усі ці точки лежать на параболі $y=x^2$. Опукла оболонка цієї множини точок представлена у стандартному вигляді, буде складатися із списку точок, впорядкованого за значенням абсциси. Один перегляд цього списку дозволяє прочитати в потрібному порядку значення x_i . Опукла оболонка буде описана послідовністю точок, x -координати яких будуть упорядковані, тому, маючи опуклу оболонку, можемо отримати впорядкований масив точок $x_1, x_2, \dots, x_N$.

Точка p опуклої множини S називається крайньою, якщо не існує точок $a, b \in S$ таких, що $p \in (a, b)$. Звідси випливає, що для того, щоб знайти випуклу оболонку, потрібно виконати два наступні кроки:

1. Визначити крайні точки.
2. Упорядкувати ці точки так, щоб вони утворили опуклий багатокутник.

Теорема 2: точка p не є крайньою плоскої опуклої множини S тоді і тільки тоді, коли p лежить у деякому трикутнику $p_1p_2p_3$, де $p_1, p_2, p_3 \in S$ (рис. 3.2).

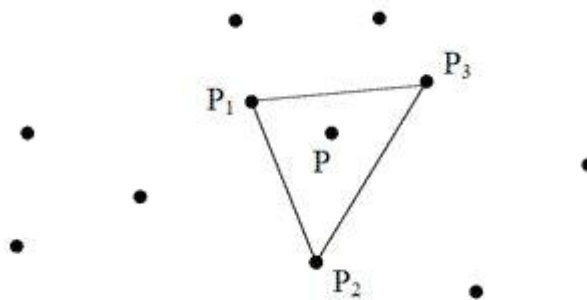


Рисунок 3.2 Ілюстрація теореми 2

За час $O(N^3)$ можна визначити, чи є точка крайньою. Побудова цієї процедури для всіх N точок множини S потребує часу $O(N^4)$.

Якщо відомі крайні точки деякої множини, то опуклу оболонку P можна знайти, вибравши точку q як внутрішню точку оболонки і впорядкувавши потім крайні точки відповідно до полярного кута відносно q (рис. 3.3).

За точку q можна взяти центроїд множини крайніх точок $p_1, p_2, \dots, p_k$: $p = (x_p, y_p)$, $x_p = \sum x_i / k$, $y_p = \sum y_i / k$. Центроїд множини із N точок в k – вимірному просторі може бути тривіально визначений за $O(Nk)$ арифметичних операцій.

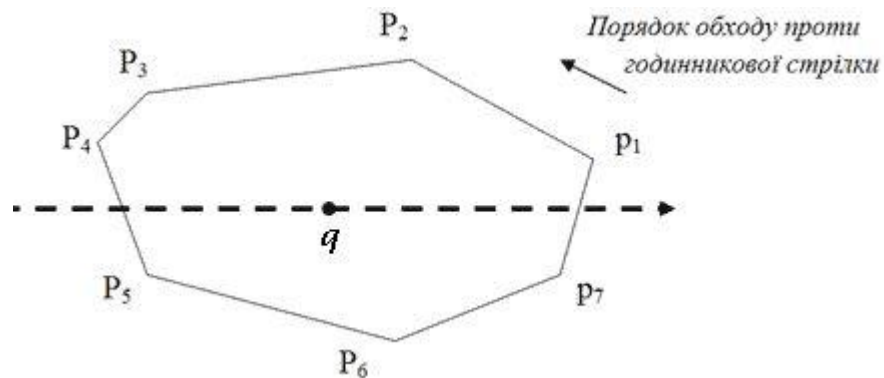


Рисунок 3.3 Вершини многокутника P впорядковані відносно точки q

Для побудови опуклої оболонки існує декілька алгоритмів [13, 14].

Алгоритм Грехема - метод знаходження опуклої оболонки для скінченної множини точок на площині за час $O(n \log n)$. Названий на честь Рональда Грехема, який опублікував первісний варіант алгоритму в 1972 році. Алгоритм знаходить всі вершини опуклої оболонки впорядковані вздовж її межі.

Перший крок в алгоритмі - знайти точку з найменшою y -координатою. Якщо таких декілька, то обираємо серед них точку з найменшою x -координатою. Назвемо її P . Цей крок має складність $O(n)$, де n — кількість точок. Додамо цю точку в стек.

Далі, точки мають бути відсортовані в порядку зростання кута, який вони разом з Р утворюють з віссю x. Будь-який алгоритм сортування підходить. Для пришвидшення обчислень, не обов'язково визначати кут, який ці точки утворюють з віссю x; замість цього достатньо обчислити котангенс цього кута: це монотонно спадна функція на проміжку, що важливий для цього кроку (від 0 до 180 градусів) і може бути обчислена простими арифметичними операціями.

Алгоритм виконується вважаючи, що точки відсортовано згідно з попереднім кроком. Для кожної точки він визначає чи було пересування від двох попередніх точок до цієї точки поворотом ліворуч чи поворотом праворуч. Якщо це був поворот праворуч, тоді передостання точка не є частиною опуклої оболонки і має бути видалена зі стека. Цей процес триває доти доки останні три точки утворюють поворот праворуч. Як тільки поворот ліворуч був отриманий, алгоритм рухається до наступної точки у відсортованому масиві.

Визначення, коли три точки утворюють поворот ліворуч, а коли поворот праворуч не вимагає обчислення кута між двома відрізками, і може бути визначено за допомогою простих арифметичних операцій. Для трьох точок (x_1, y_1) , (x_2, y_2) і (x_3, y_3) , просто обчисліть напрямок векторного добутку двох векторів визначених точками (x_1, y_1) , (x_2, y_2) і (x_1, y_1) , (x_3, y_3) , який характеризується знаком виразу $(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)$.

Якщо результат 0, точки колінеарні; якщо позитивний, точки утворюють поворот ліворуч, інакше поворот праворуч.

Насамкінець алгоритм досягне точки з якої ми починали, тепер він завершився і стек містить точки опуклої оболонки в напрямку зворотному до годинникового.

Алгоритм Джарвіса - алгоритм знаходження опуклої оболонки. Часова складність - $O(n * h)$, де n - кількість точок, h - кількість точок опуклої

оболонки. Тобто, алгоритм найбільш ефективний у випадку малої кількості точок опуклої оболонки.

Нехай шукана опукла оболонка множини $P = \{p_1, p_2, \dots, p_n\}$ точок. У якості початкової беремо крайню ліву точку, якщо їх буде декілька, то виберемо серед них точку з найменшою у-координатою. Нехай знайдена точка – точка p_1 . Точка p_1 напевно є вершиною опуклої оболонки. Далі для кожної точки p_i шукаємо проти годинникової стрілки точку p_{i+1} з найменшим полярним кутом $p_{i-1}p_ip_{i+1}$. Вона і буде наступною вершиною опуклої оболонки. При цьому не обов'язково обчислювати кут – достатньо обчислити векторний добуток між кутами $p_ip'_{i+1}$ та $p_ip''_{i+1}$, де p'_{i+1} – знайдений на даний момент мінімум. Якщо векторний добуток від'ємний, то знайдемо новий мінімум. Якщо рівний нулю, тобто p'_{i+1} та p''_{i+1} лежать на одній прямій, то мінімум – та точка, яка лежить далі від точки P_i . Алгоритм продовжує роботу доки $p_{i+1} \neq p_1$

Алгоритм Кіркпатрика-Зейделя - побудова опуклої оболонки методом «розділяй та володарюй». Алгоритм побудови опуклої оболонки зі швидкістю $O(n \log h)$, де n - кількість вхідних точок, h - кількість точок в опуклій оболонці.

Задана множина S , що складається з N точок.

1. Якщо $N \leq N_0$ (N_0 - деяке невелике ціле число), то побудувати опуклу оболонку одним з відомих методів та зупинитися, інакше перейти до кроку 2.
2. Розіб'ємо початкову множину S довільним чином на дві приблизно рівних за потужністю підмножини S_1 та S_2 (нехай S_1 містить $\lceil \frac{N}{2} \rceil$ точок, а S_2 містить $N - \lceil \frac{N}{2} \rceil$ точок).
3. Рекурсивно знаходимо опуклі оболонки кожної з підмножин.
4. Будуємо опуклу оболонку початкової множини об'єднанням двох опуклих оболонок $CH(S_1)$ і $CH(S_2)$.

Алгоритм Чена - алгоритм побудови опуклої оболонки скінченної множини точок на площині. Виконується за час $O(n \log h)$, де h - кількість точок опуклої оболонки. Є комбінацією алгоритму, що обчислює опуклу оболонку за час $O(n \log n)$ (наприклад, алгоритм Грехема) з алгоритмом загортання за Джарвісом, який виконується за час $O(nh)$.

Ідея алгоритму Чена полягає у початковому поділі всіх точок на групи по m штук в кожній. Відповідно, кількість груп дорівнює $r = \lfloor \frac{n}{m} \rfloor$. Для кожної з груп будується опукла оболонка скануванням за Грехемом, або якимось іншим, що працює зі швидкістю $O(m \log m)$. Таким чином, час витрачений для всіх груп точок складе $O(rm \log m) = O(n \log m)$.

Далі, починаючи з найнижчої точки (якщо таких декілька, то обираємо з них крайню ліву), для отриманих в результаті розбиття оболонок будується спільна опукла оболонка за алгоритмом Джарвіса. При цьому наступна точка опуклої оболонки знаходиться за час $O(r \log m)$, так як для того, щоб знайти точку з максимальним тангенсом по відношенню до точки, що розглядається, в m - кутнику достатньо затратити $O(\log m)$. Логарифмічний час на пошук, витрачається тому, що точки в m кутнику вже впорядковані за полярним кутом під час виконання алгоритму сканування за Грехемом. У підсумку, на обхід потрібно $O(hr \log m) = O\left(\frac{hn}{m}\right) \log m$ часу.

Алгоритм швидкої оболонки - метод обчислення опуклої оболонки скінченної множини точок на площині. Використовує підхід «розділяй та володарюй», який полягає в тому, що задача розбивається на підзадачі приблизно однакового розміру. Аналогічний метод, використовується в алгоритмі швидкого сортування, звідси така назва.

Алгоритм можна розбити на наступні етапи:

1. Знайти точки з мінімальною і максимальною x координатою, вони зобов'язані бути частиною опуклої оболонки.
2. Використовуючи лінію, утворену двома точками розділити всю множину точок на дві підмножини, які будуть оброблятися рекурсивно.

3. Визначити точку, на одній стороні лінії, з максимальною відстанню від лінії. Знайдені до цього дві точки утворюють з цією точкою трикутник з найбільшою площею.
4. Точки, що лежать всередині цього трикутника не можуть бути частиною опуклої оболонки і, отже, можуть бути проігноровані в наступних кроках.
5. Повторіть попередні два кроки для двох ліній, утвореного трикутника (окрім початкової лінії).
6. Продовжуйте робити так доти, поки більше точок не залишиться, у кінці рекурсії, вибрані точки, складуть опуклу оболонку.

Алгоритм Грехема-Ендрю - алгоритм побудови опуклої оболонки в двовимірному просторі, модифікація алгоритму Грехема.

На відміну від алгоритму Грехема, в якому побудова здійснюється за допомогою стеку, використовує лексикографічне впорядкування точок по координатах, що дозволяє алгоритму не використовувати дійсні числа і тригонометричні функції. Алгоритм окремо обчислює верхню і нижню оболонки з послідовних ланцюгів точок. Фактично, алгоритм Ендрю є окремим випадком алгоритму Грехема, коли центральна точка вибирається нескінченно віддаленою у від'ємному напрямку по вісі ординат, так що в цьому випадку впорядкування по абсцисі збігається з впорядкуванням по полярному куту. Алгоритм Грехема-Ендрю полягає в наступному.

Перший крок: нехай на площині задано множину з N точок. Визначимо спочатку її ліву і праву крайні точки L та R і побудуємо пряму проходить через них.

Другий крок: решта точок розбиваються на дві підмножини (нижню і верхню) залежно від того, по який бік від прямої вони розташовуються - нижче або вище.

Третій крок: для кожної підмножини шукаємо опуклу оболонку використовуючи алгоритм Грехема з умовою, що сортуємо не по полярному куту, а по координаті.

Четвертий крок: поєднання цих двох ламаних дає опуклу оболонку вихідної множини.

Трудомісткість алгоритму $O(n \log n)$.

Проаналізувавши плюси і мінуси кожного з алгоритмів, для розв'язання поставленої задачі було вирішено використовувати алгоритм Грехема-Ендрю.

Вирішальними факторами у виборі алгоритму стали його відносна простота та швидкість роботи.

3.2 Визначення площі пошкоджених ділянок

Для обчислення площі дефективних зон було вирішено застосувати триангуляцію.

Триангуляція в найзагальнішому значенні - це розбиття геометричного об'єкта на симплекси. Наприклад, на площині це розбиття на трикутники, звідки й назва.

Триангуляція T простору R^{n+1} - це розбиття R^{n+1} на $n+1$ -вимірні симплекси, такі що:

1. будь-які два симплекси в T перетинаються в спільній грані, ребру чи вершині, або взагалі не перетинаються;
2. будь-яка обмежена множина в R^{n+1} перетинає скінченну кількість симплексів з T .

Триангуляція дискретної множини точок $P \subset R^{n+1}$ - це розбиття опуклої оболонки точок на симплекси так, що виконується перша умова з попереднього означення.

Триангуляція Делоне є найвідомішим видом триангуляції множини точок.

Триангуляція Делоне для множини точок P на площині (рис 3.4) — це така триангуляція $DT(P)$, що жодна точка множини P не знаходиться всередині описаних довкола трикутників кіл в множині $DT(P)$. Триангуляція Делоне

дозволяє якомога зменшити кількість малих кутів. Цей спосіб триангуляції був винайдений Борисом Делоне в 1934 році [15].

Базуючись на визначенні Делоне, описане коло трикутника утворене трьома точками з вихідної множини точок називається пустим, якщо воно не містить вершин трикутника інших ніж ті три, що його задають (інші точки допускаються тільки на периметрі кола, але не всередині)

Умова Делоне стверджує, що мережа трикутників є триангуляцією Делоне, якщо всі описані кола трикутників пусті. Це є початкове визначення для двовимірного простору. Його можна використовувати для тривимірного простору, якщо використовувати описані сфери замість описаних кіл.

Для множини точок на одній лінії триангуляції Делоне не існує. Для чотирьох точок на одному колі триангуляція Делоне має два випадки, тобто можна розділити цей чотирикутник двома способами, які задовольняють умови Делоне.

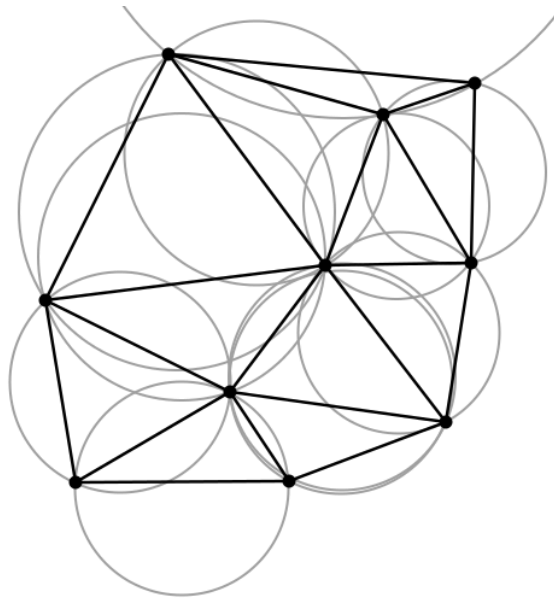


Рисунок 3.4 Триангуляція Делоне

Триангуляція многокутника - це розбиття многокутника на трикутники, що мають спільні ребра з умовою, що множина вершин трикутників співпадає з множиною вершин многокутника.

Гранична триангуляція Делоне - це адаптація триангуляції Делоне від множин точок до многокутників, у загальнішому – до планарних графів.

В методі скінченних елементів триангуляція використовується в якості сітки, що є основою обчислення. В цьому випадку, трикутники повинні утворювати множину в області визначення функції. Відомі багато сіточних технік, що містять уточнення Делоне, наприклад другий алгоритм Чу та алгоритм Руперта.

3.3 Технологія оцінки технічного стану

Вихідними даними для застосування даного підходу служить фотокартка конструктивного елемента, стан якого необхідно визначити. Важливою умовою є те, що на фото має бути зображений об'єкт повністю. Також, для масштабування елемента необхідний його технічний паспорт.

Технологія оцінки технічного стану складається з наступних кроків:

Здійснення візуального обстеження об'єкту, фото зйомка.

Для оцінки технічного стану конструктивного елементу необхідне його фото та геометричні параметри. Фотозйомку проводить фахівець з відповідними навичками. При цьому повинна використовуватись апаратура, що забезпечує високу якість зображення, розширення не менше 1024×768 пікселів. Геометричні параметри об'єкту що оцінюється беруться з його технічного паспорта. На рисунку 3.5 зображено фото елемента для оцінки.



Рисунок 4.1 Фото елемента , технічний стан якого оцінюється

Опис стану конструктивного елемента.

Після отримання потрібних знімків об'єкту, проводиться його опис. Вказується терміновість проведення оцінки. Визначається наскільки великими є проблемні зони. Робиться припущення про стан об'єкту.

Застосування фільтрів для деталізації дефектів.

Проаналізувавши отримані дані, для уточнення та деталізації проблемних зон, застосовують антишумові фільтри, які дозволяють позбутися пилу та бруду на фото(рис. 4.2).



Рисунок 4.2 Фото, очищене від пилу та бруду

Визначення типу дефектів.

Після накладення спеціальних фільтрів, експерт вказує типи дефектів, наявних на зображенні.

Розстановка реперних точок.

Проблемні зони, наявні на зображенні необхідно виділити. Для цього користувач розставляє реперні точки, які локалізують дефекти.

Застосування до пошкоджених ділянок алгоритму Грехема-Ендрю.

Для визначення площі дефективних зон необхідно спочатку їх виділити. Для цього застосовуємо алгоритм обходу Грехема-Ендрю.

Визначення загальних площ пошкоджених зон в залежності від типу дефектів.

Для визначення площі пошкоджених ділянок застосовуємо метод триангуляції Делоне. Після розбиття багатокутника на трикутники за допомогою триангуляції Делоне знаходимо його площу.

4 ПРОЕКТУВАННЯ БАЗИ ДАНИХ ТА АРХІТЕКТУРИ СИСТЕМИ

4.1 Визначення сутностей бази даних системи

В результаті аналізу предметної області та розробки концептуальної моделі системи було визначено такі сутності майбутньої БД: технічний стан, дефект, елемент проїзної частини, регуляційна споруда, підхід, елемент прогонової будови, фундамент, опора, матеріал, виробник. Для збереження результатів оцінки призначені такі додаткові сутності: пошкоджений елемент прогонової будови, пошкоджений елемент проїзної частини, пошкоджена регуляційна споруда, пошкоджений підхід, пошкоджений фундамент, пошкоджена опора. Сутності БД системи представлені в таблиці 4.1.

Таблиця 4.1 Сутності БД та їх атрибути

Назва сутності	Атрибути
1	2
Технічний стан	Код
	Назва
	Опис
Дефект	Код
	Назва
Елемент проїзної частини	Код
	Назва
Регуляційна споруда	Код
	Назва
Підхід	Код

	Назва
--	-------

Продовження таблиці 4.1

Назва сутності	Атрибути
1	2
Елемент прогонової будови	Код
	Назва
Фундамент	Код
	Назва
Опора	Код
	Назва
Матеріал	Код
	Назва
Виробник	Код
	Назва
	Адреса
	Телефон
Пошкоджений елемент проїзної частини	Код
	Площа
Пошкоджена регуляційна споруда	Код
	Площа
Пошкоджений підхід	Код
	Площа
Пошкоджений елемент прогонової	Код

будови	Площа
--------	-------

Продовження таблиці 4.1

Назва сутності	Атрибути
1	2
Пошкоджений фундамент	Код
	Площа
Пошкоджена опора	Код
	Площа

Всі необхідні сутності визначені. Наступним кроком є розробка концептуальної моделі бази даних.

4.2 Концептуальна модель бази даних

Концептуальна модель - це відображення предметної області, для якої розробляється база даних. Побудова концептуальної моделі являє собою процес моделювання смислового наповнення бази даних. Концептуальна модель складається з наступних трьох основних компонентів [16, 17].

1. Сутності. Це елементи реального світу, які можуть існувати незалежно. Сутність представляється у концептуальній моделі прямокутником, в якому зазначено її назву.

2. Атрибути. Атрибути описують сутність. Вони зображуються овалами з зазначенням імен, які прикріплені до сутності.

3. Зв'язки. Зв'язок представляє взаємодію між сутностями. На діаграмі вона зображується ромбом, який з'єднує сутності, що беруть участь у зв'язку. Потужність зв'язку позначається стрілками (у напрямку, де потужність

дорівнює багатьом - подвійна стрілка, а з боку, де вона дорівнює одиниці - одинарна).

Концептуальна модель системи зображена на рисунку 4.1.

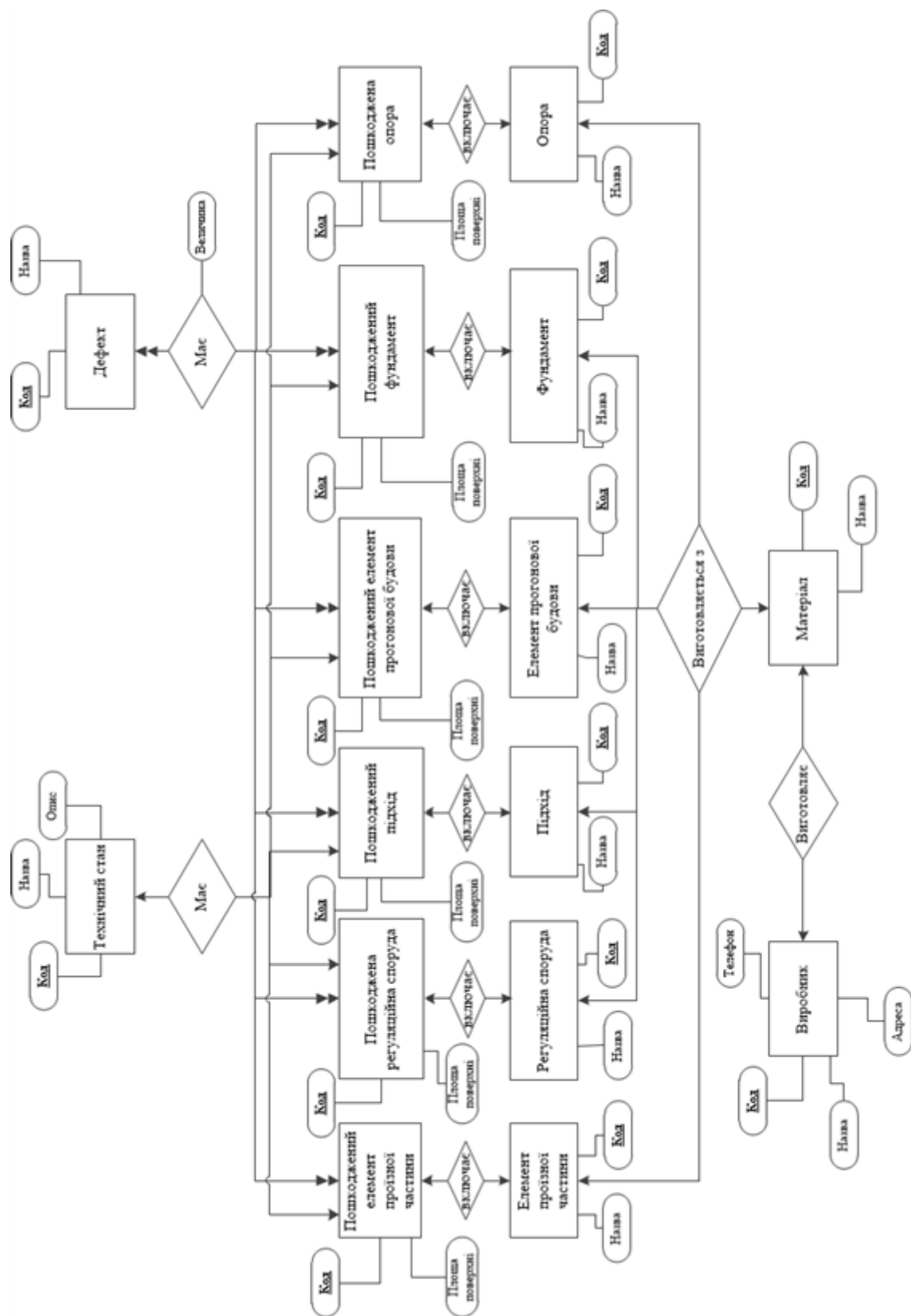


Рисунок 4.1 Концептуальна модель БД

4.3 Логічна модель бази даних

На основі концептуальної моделі (див. рис. 4.1) було розроблено логічну модель бази даних (рис. 4.2).

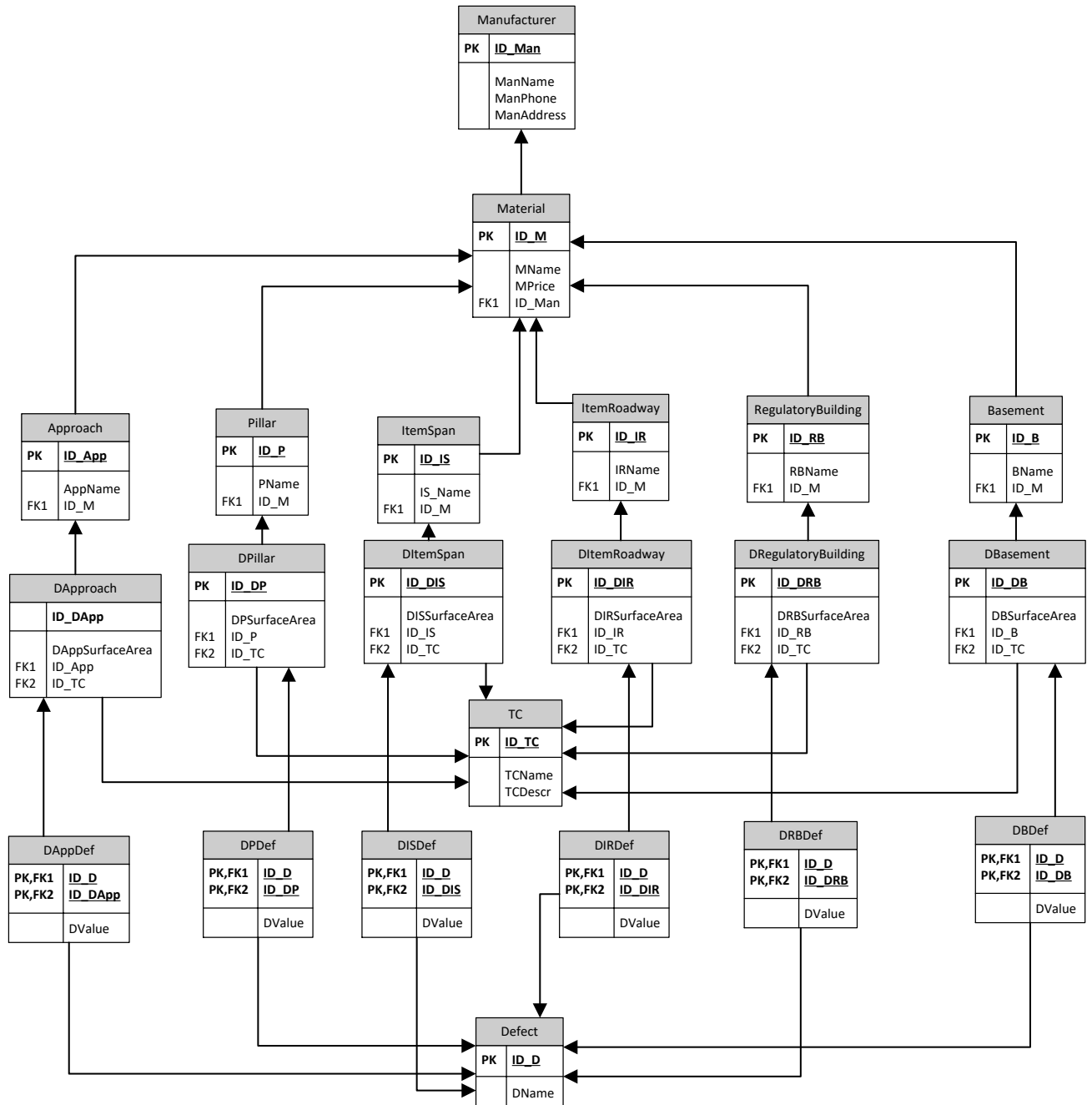


Рисунок 4.2 Логічна модель БД

4.4 Фізична модель бази даних

На основі логічної моделі розроблено фізичну модель БД (рис. 4.3)

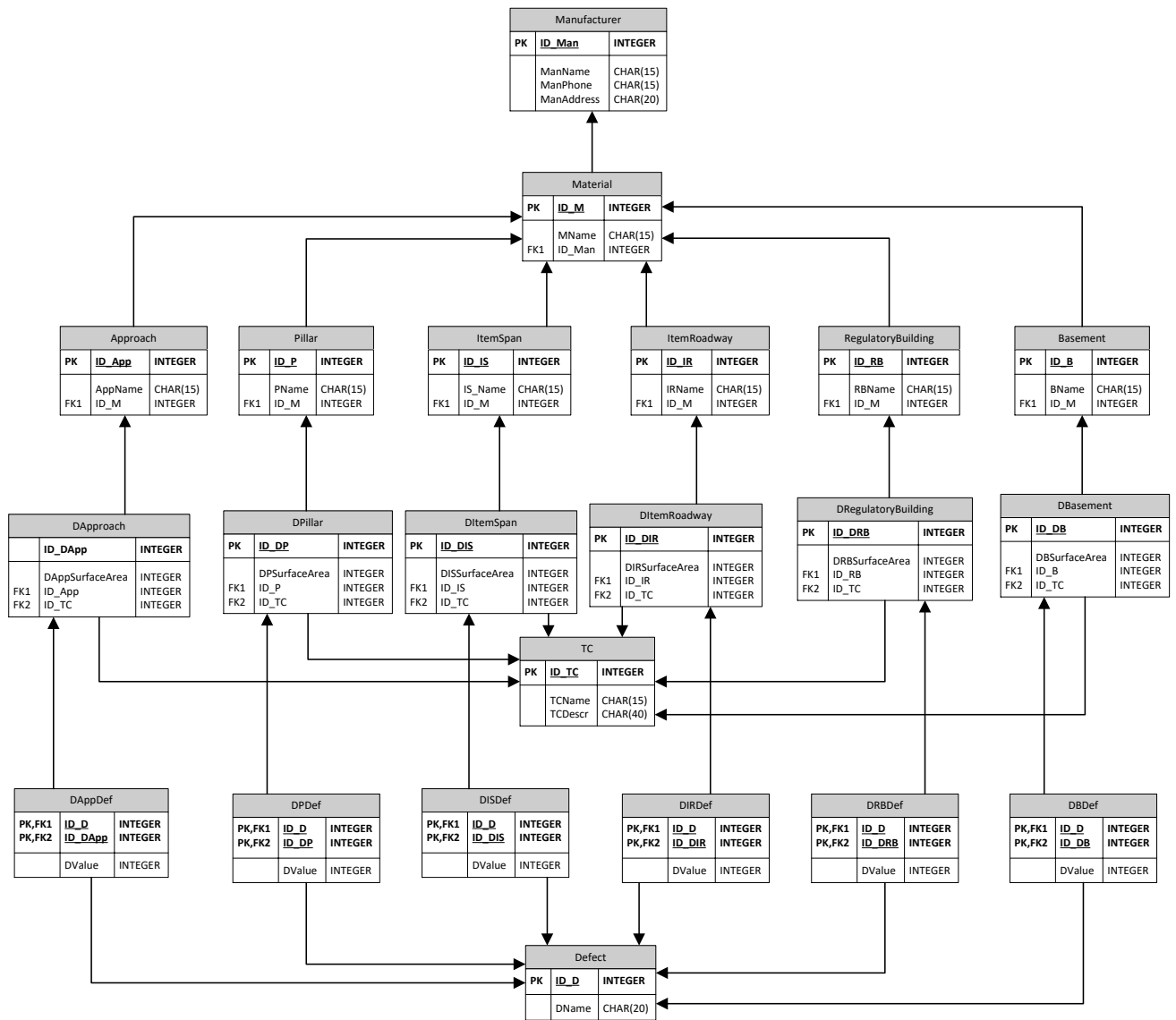


Рисунок 4.3 Фізична модель БД

Далі розглянемо структуру таблиць фізичної моделі БД.

Таблиця Manufacturer

Таблиця 4.2 Фізична структура таблиці Manufacturer

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_Man	ID_Man	Integer	PK
ManName	ManName	Char(15)	
ManPhone	ManPhone	Char(15)	
ManAddress	ManAddress	Char(20)	

ID_Man – унікальний ідентифікатор виробника;

ManName – назва підприємства;

ManPhone – телефон підприємства;

ManAddress – адреса підприємства.

Таблиця Material

Таблиця 4.3 Фізична структура таблиці Material

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_M	ID_M	Integer	PK
MName	MName	Char(15)	
ID_Man	ID_Man	Integer	FK

ID_M – унікальний ідентифікатор матеріалу;

MName – назва матеріалу;

ID_Man – ідентифікатор виробника матеріалу.

Таблиця Defect

Таблиця 4.4 Фізична структура таблиці Defect

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_D	ID_D	Integer	PK
DName	DName	Char(20)	

ID_D – унікальний ідентифікатор дефекту;

DName – назва дефекту.

Таблиця TC

Таблиця 4.5 Фізична структура таблиці TC

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_TC	ID_TC	Integer	PK
TCName	TCName	Char(15)	
TCDescr	TCDescr	Char(40)	

ID_TC – унікальний ідентифікатор категорії технічного стану;

TCName – назва категорії технічного стану;

TCDescr – опис категорії технічного стану.

Таблиця Approach

Таблиця 4.6 Фізична структура таблиці Approach

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_App	ID_App	Integer	PK
AppName	AppName	Char(15)	
ID_M	ID_M	Integer	FK

ID_App – унікальний ідентифікатор підходу;

AppName – назва підходу;

ID_M – ідентифікатор матеріалу.

Таблиця Pillar

Таблиця 4.7 Фізична структура таблиці Pillar

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_P	ID_P	Integer	PK
PName	PName	Char(15)	
ID_M	ID_M	Integer	FK

ID_P – унікальний ідентифікатор опори;

PName – назва опори;

ID_M – ідентифікатор матеріалу.

Таблиця ItemSpan

Таблиця 4.8 Фізична структура таблиці ItemSpan

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_IS	ID_IS	Integer	PK
IS_Name	IS_Name	Char(15)	
ID_M	ID_M	Integer	FK

ID_IS – унікальний ідентифікатор елементу прогонової будови;

IS_Name – назва елементу прогонової будови;

ID_M – ідентифікатор матеріалу.

Таблиця ItemRoadway

Таблиця 4.9 Фізична структура таблиці ItemRoadway

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_IR	ID_IR	Integer	PK
IRName	IRName	Char(15)	
ID_M	ID_M	Integer	FK

ID_IR – унікальний ідентифікатор елементу проїзної частини;

IRName – назва елементу проїзної частини;

ID_M – ідентифікатор матеріалу.

Таблиця RegulatoryBuilding

Таблиця 4.10 Фізична структура таблиці RegulatoryBuilding

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_RB	ID_RB	Integer	PK
RBName	RBName	Char(15)	
ID_M	ID_M	Integer	FK

ID_RB – унікальний ідентифікатор регуляційної споруди;

IRName – назва регуляційної споруди;

ID_M – ідентифікатор матеріалу.

Таблиця Basement

Таблиця 4.11 Фізична структура таблиці Basement

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_B	ID_B	Integer	PK
BName	BName	Char(15)	
ID_M	ID_M	Integer	FK

ID_B – унікальний ідентифікатор фундаменту;

BName – назва фундаменту;

ID_M – ідентифікатор матеріалу.

Таблиця DApproach

Таблиця 4.12 Фізична структура таблиці Basement

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_DApp	ID_DApp	Integer	PK
DAppSurface-Area	DAppSurface-Area	Integer	
ID_App	ID_App	Integer	FK
ID_TC	ID_TC	Integer	FK

ID_DApp – унікальний ідентифікатор пошкодженого підходу;

DAppSurfaceArea – площа поверхні пошкодженого підходу;

ID_App – ідентифікатор підходу;

ID_TC – ідентифікатор категорії технічного стану;

Таблиця DPillar

Таблиця 4.13 Фізична структура таблиці DPillar

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_DP	ID_DP	Integer	PK
DPSurfaceArea	DPSurfaceArea	Integer	
ID_P	ID_P	Integer	FK
ID_TC	ID_TC	Integer	FK

ID_DP – унікальний ідентифікатор пошкодженої опори;

DPSurfaceArea – площа поверхні пошкодженої опори;

ID_P – ідентифікатор опори;

ID_TC – ідентифікатор категорії технічного стану;

Таблиця DItemSpan

Таблиця 4.14 Фізична структура таблиці DItemSpan

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_DIS	ID_DIS	Integer	PK
DISSurfaceArea	DISSurfaceArea	Integer	
ID_IS	ID_IS	Integer	FK
ID_TC	ID_TC	Integer	FK

ID_DIS – унікальний ідентифікатор пошкодженого елемента прогонової будови;

DISSurfaceArea – площа поверхні пошкодженого елемента прогонової будови;

ID_IS – ідентифікатор елемента прогонової будови;

ID_TC – ідентифікатор категорії технічного стану;

Таблиця DItemRoadway

Таблиця 4.15 Фізична структура таблиці DItemRoadway

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_DIR	ID_DIR	Integer	PK
DIRSurfaceArea	DIRSurfaceArea	Integer	

Продовження таблиці 4.15

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_IR	ID_IR	Integer	FK
ID_TC	ID_TC	Integer	FK

ID_DIR – унікальний ідентифікатор пошкодженого елементу проїзної частини ;

DISSurfaceArea – площа поверхні пошкодженого елементу проїзної частини;

ID_IK – ідентифікатор елементу проїзної частини;

ID_TC – ідентифікатор категорії технічного стану;

Таблиця DRegulatoryBuilding

Таблиця 4.16 Фізична структура таблиці DRegulatoryBuilding

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_DRB	ID_DRB	Integer	PK
DRBSurfaceArea	DRBSurfaceArea	Integer	
ID_RB	ID_RB	Integer	FK
ID_TC	ID_TC	Integer	FK

ID_DRB – унікальний ідентифікатор пошкодженої регуляційної споруди;

DRBSurfaceArea – площа поверхні пошкодженої регуляційної споруди;

ID_RB – ідентифікатор регуляційної споруди;

ID_TC – ідентифікатор категорії технічного стану;

Таблиця DBasement

Таблиця 4.17 Фізична структура таблиці DBasement

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_DB	ID_ DB	Integer	PK
DBSurfaceArea	DBSurfaceArea	Integer	
ID_RB	ID_ RB	Integer	FK
ID_TC	ID_ TC	Integer	FK

ID_DB – унікальний ідентифікатор пошкодженого фундаменту;

DBSurfaceArea – площа поверхні пошкодженого фундаменту;

ID_B – ідентифікатор фундаменту;

ID_TC – ідентифікатор категорії технічного стану;

Таблиця D DAppDef

Таблиця 4.18 Фізична структура таблиці DAppDef

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_D	ID_ D	Integer	PK, FK
ID_DApp	ID_DApp	Integer	PK, FK
IDValue	IDValue	Integer	

ID_D – ідентифікатор дефекту;

ID_DApp – ідентифікатор пошкодженого підходу;

IDValue – величина дефекту;

Таблиця DPDef

Таблиця 4.19 Фізична структура таблиці DPDef

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_D	ID_D	Integer	PK, FK
ID_DP	ID_DP	Integer	PK, FK
IDValue	IDValue	Integer	

ID_D – ідентифікатор дефекту;

ID_DP – ідентифікатор пошкодженої опори;

IDValue – величина дефекту;

Таблиця DISDef

Таблиця 4.20 Фізична структура таблиці DISDef

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_D	ID_D	Integer	PK, FK
ID_DIS	ID_DIS	Integer	PK, FK
IDValue	IDValue	Integer	

ID_D – ідентифікатор дефекту;

ID_DIS – ідентифікатор пошкодженого елемента прогонової будови;

IDValue – величина дефекту;

Таблиця DIRDef

Таблиця 4.21 Фізична структура таблиці DIRDef

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_D	ID_D	Integer	PK, FK
ID_DIR	ID_DIR	Integer	PK, FK
IDValue	IDValue	Integer	

ID_D – ідентифікатор дефекту;

ID_DIR – ідентифікатор пошкодженого елементу проїзної частини;

IDValue – величина дефекту;

Таблиця DRBDef

Таблиця 4.22 Фізична структура таблиці DRBDef

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_D	ID_D	Integer	PK, FK
ID_DRB	ID_DRB	Integer	PK, FK
IDValue	IDValue	Integer	

ID_D – ідентифікатор дефекту;

ID_DRB – ідентифікатор пошкодженої регуляційної споруди;

IDValue – величина дефекту;

Таблиця DBDef

Таблиця 4.23 Фізична структура таблиці DBDef

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
ID_D	ID_D	Integer	PK, FK
ID_DB	ID_DB	Integer	PK, FK
IDValue	IDValue	Integer	

ID_D – ідентифікатор дефекту;

ID_DRB – ідентифікатор пошкодженого фундаменту;

IDValue – величина дефекту;

В якості СУБД було обрано MySQL - безкоштовна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL вважається гарним рішенням для малих і середніх застосувань.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;

- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

4.5 Діаграма основних класів системи

Діаграма класів - це набір статичних, декларативних елементів моделі. Діаграми класів можуть застосовуватися і при прямому проектуванні, тобто в процесі розробки нової системи, і при зворотному проектуванні - описі існуючих і використовуваних систем. Інформація з діаграми класів безпосередньо відображається у вихідний код програми - у більшості існуючих інструментів UML-моделювання можлива кодогенерація для певної мови програмування. Таким чином, діаграма класів - кінцевий результат проектування і відправна точка процесу розробки [18, 19].

На рисунку 3.4 наведено діаграму основних класів розроблюваної системи.

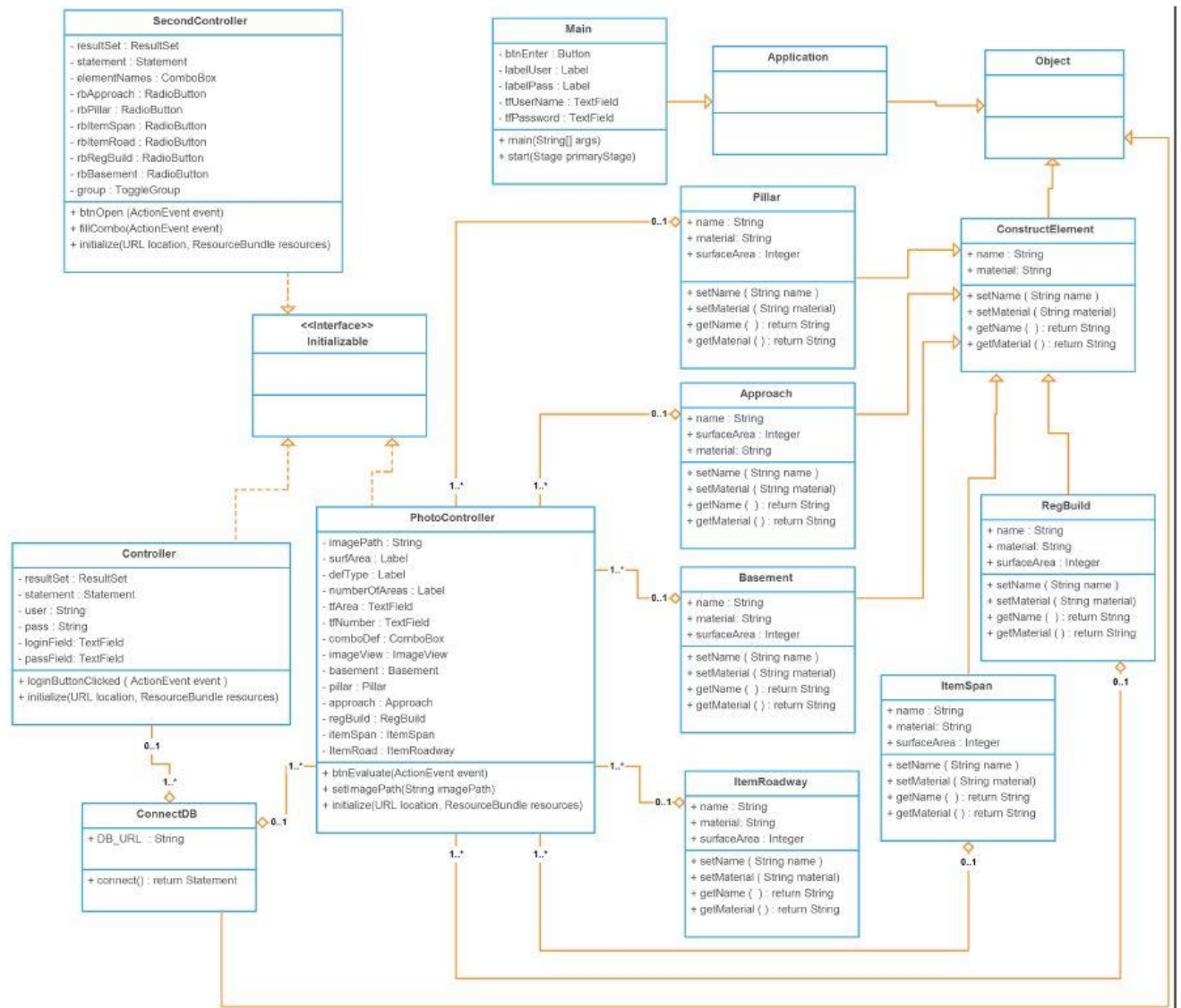


Рисунок 4.4 Діаграма класів системи

Програмний код основних класів:

Клас **ItemRoadway**

```

public class ItemRoadway extends ConstructElement {
    private Integer surfaceArea;

    public ItemRoadway(String name, String material, Integer surfaceArea) {
        super(name, material);
        this.surfaceArea = surfaceArea;
    }

    public Integer getSurfaceArea() {

```

```
        return surfaceArea;
    }
    public void setSurfaceArea(Integer surfaceArea) {
        this.surfaceArea = surfaceArea;
    }
}
```

Клас **RegBuild**

```
public class RegBuild extends ConstructElement {
    private Integer surfaceArea;

    public RegBuild(String name, String material, Integer surfaceArea) {
        super(name, material);
        this.surfaceArea = surfaceArea;
    }
    public Integer getSurfaceArea() {
        return surfaceArea;
    }
    public void setSurfaceArea(Integer surfaceArea) {
        this.surfaceArea = surfaceArea;
    }
}
```

Клас **ItemSpan**

```
public class ItemSpan extends ConstructElement {
    private Integer surfaceArea;

    public ItemSpan(String name, String material, Integer surfaceArea) {
        super(name, material);
        this.surfaceArea = surfaceArea;
    }
    public Integer getSurfaceArea() {
        return surfaceArea;
    }
}
```

```

    }
    public void setSurfaceArea(Integer surfaceArea) {
        this.surfaceArea = surfaceArea;
    }
}

```

Κлас **Basement**

```

public class Basement extends ConstructElement {
    private Integer surfaceArea;
    public Basement(String name, String material, Integer surfaceArea) {
        super(name, material);
        this.surfaceArea = surfaceArea;
    }
    public Integer getSurfaceArea() {
        return surfaceArea;
    }
    public void setSurfaceArea(Integer surfaceArea) {
        this.surfaceArea = surfaceArea;
    }
}

```

Κлас **Approach**

```

public class Approach extends ConstructElement{
    private Integer surfaceArea;
    public Approach(String name, String material, Integer surfaceArea) {
        super(name, material);
        this.surfaceArea = surfaceArea;
    }
    public Integer getSurfaceArea() {
        return surfaceArea;
    }
    public void setSurfaceArea(Integer surfaceArea) {

```

```
        this.surfaceArea = surfaceArea;
    }
}
```

Κλας **Pillar**

```
public class Pillar extends ConstructElement {
    private Integer surfaceArea;
    public Pillar(String name, String material, Integer surfaceArea) {
        super(name, material);
        this.surfaceArea = surfaceArea;
    }
    public Integer getSurfaceArea() {
        return surfaceArea;
    }
    public void setSurfaceArea(Integer surfaceArea) {
        this.surfaceArea = surfaceArea;
    }
}
```

Κλας **ConstructElement**

```
public class ConstructElement {
    private String name;
    private String material;
    public ConstructElement(String name, String material) {
        this.name = name;
        this.material = material;
    }
    public ConstructElement() {
    }
    public String getName() {
        return name;
    }
}
```

```

public void setName(String name) {
    this.name = name;
}
public String getMaterial() {
    return material;
}
public void setMaterial(String material) {
    this.material = material;
}
}

```

Клас **ConnectDB**

```

public class ConnectDB {
    static final String DB_URL = "jdbc:mysql://localhost:3306/tc";
    public static Statement connect() throws SQLException {
        Connection connection = null;
        Statement statement = null;
        ResultSet resultSet = null;
        System.out.println();
        connection = DriverManager.getConnection(DB_URL, "root", "root");
        System.out.println(connection);
        statement = connection.createStatement();
        return statement;
    }
}

```

Клас **Controller**

```

public class Controller implements Initializable {
    ResultSet resultSet = null;
    Statement statement = null;
    String user = "";
    String pass = "";
}

```

```

@FXML
private TextField loginField;
@FXML
private TextField passField;
public void loginButtonClicked(ActionEvent event) throws Exception {...}
@Override
public void initialize(URL location, ResourceBundle resources) {...}
}

```

Клас **SecondController**

```

public class SecondController implements Initializable {
    ResultSet resultSet = null;
    Statement statement = null;
    @FXML
    private ComboBox elementNames;
    @FXML
    private RadioButton rbApproach;
    @FXML
    private RadioButton rbPillar;
    @FXML
    private RadioButton rbItemSpan;
    @FXML
    private RadioButton rbItemRoad;
    @FXML
    private RadioButton rbRegBuild;
    @FXML
    private RadioButton rbBasement;
    private ToggleGroup group;
    @FXML
    private void btnOpen (ActionEvent event) throws IOException {...}
    @FXML

```

```

    public void fillCombo(ActionEvent event) throws Exception {...}
    @Override
    public void initialize(URL location, ResourceBundle resources) {

    }
}

```

Клас **PhotoController**

```

public class PhotoController implements Initializable {
    @FXML
    private AnchorPane anchorRoot;
    static String imagePath = "";
    @FXML
    private Label surfArea;
    @FXML
    private Label defType;
    @FXML
    private Label numberOfAreas;
    @FXML
    private TextField tfArea;
    @FXML
    private TextField tfNumber;
    @FXML
    private ComboBox comboDef;
    @FXML
    private ImageView imageView = new ImageView();
    MouseButton mouseButton = null;
    @Override
    public void initialize(URL location, ResourceBundle resources) {... }
    @FXML
    private void btnEvaluate(ActionEvent event) {

```

```

    }
    public static void setImagePath(String imagePath) {
        PhotoController.imagePath = imagePath;
    }
}

```

*Клас **Main***

```

public class Main extends Application {
    Button btnEnter;
    Label labelUser;
    Label labelPass;
    TextField tfUserName;
    TextField tfPassword;
    @Override
    public void start(Stage primaryStage) throws Exception {...}
    public static void main(String[] args) {...}
}

```

4.6 Опис алгоритму функціонування системи

Алгоритм – послідовність дій, що приводить до рішення поставленої задачі за скінчену кількість операцій.

Можна виділити три основних види обчислювальних алгоритмів:

- *лінійний*, в якому всі команди виконуються в порядку їх запису;
- *з розгалуженням*, в якому в залежності від деякої умови виконуються ті чи інші команди;
- *циклічний*, в якому деякі команди повторюються;
- *змішаний*, в якому зустрічаються інші типи.

Схема алгоритму – це графічне представлення логічної структури алгоритму, де кожний етап обробки інформації зображається у вигляді геометричних символів

Схема основного алгоритму функціонування системи зображена на рисунку 4.5.

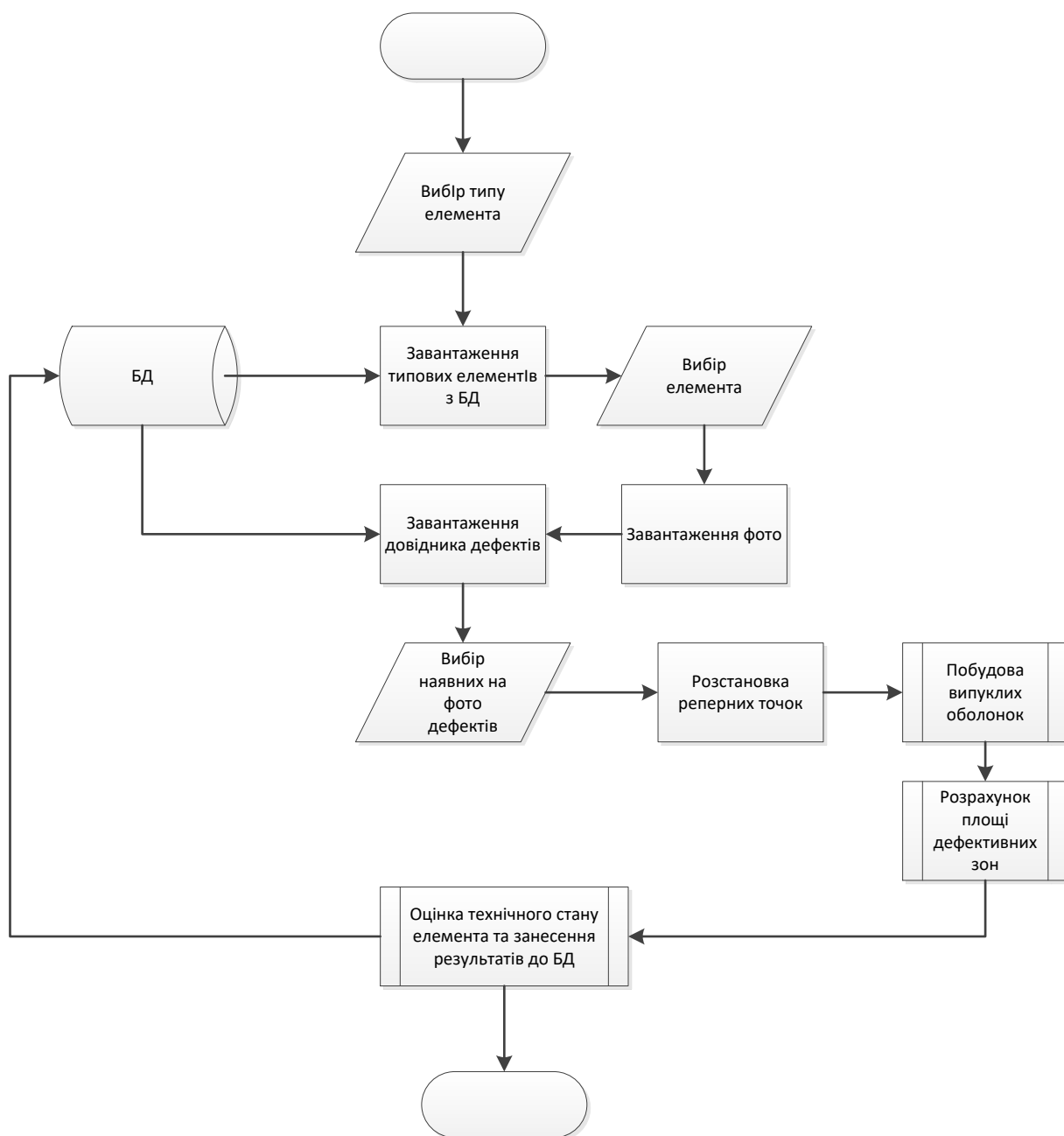


Рисунок 4.5 Схема основного алгоритму системи

5 КОНТРОЛЬНИЙ ПРИКЛАД РОБОТИ ПРОГРАМИ

5.1 Аналіз інструментальних засобів розробки системи

Для розробки програмного продукту було обрано програмне середовище IntelliJ IDEA.

IntelliJ IDEA – комерційне інтегроване середовище розробки програмного забезпечення на багатьох мовах програмування, зокрема Java, JavaScript, Python, розроблене компанією JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної версії "Community Edition" і повнофункціональної комерційної версії "Ultimate Edition", для якої активні розробники відкритих проектів мають можливість отримати безкоштовну ліцензію.

Перша версія IntelliJ IDEA з'явилася у січні 2001 року й швидко здобула популярність, як перша Java IDE із широким набором інтегрованих інструментів для рефакторингу, що дозволяла програмістам швидко реорганізовувати сирцевий код програм. Дизайн середовища орієнтовано на продуктивність праці програмістів, дозволяючи їм сконцентруватися на розробці функціональності, тоді як IntelliJ IDEA бере на себе виконання рутинних операцій.

Починаючи з шостої версії продукту IntelliJ IDEA надає інтегрований інструментарій для розробки графічного користувацького інтерфейсу.

З версії 9.0 є безкоштовний варіант Community Edition з відкритими кодами. Сирцеві коди відкритої версії IntelliJ IDEA Community Edition поширюються в рамках ліцензії Apache 2.0. Бінарні пакети підготовлені для Linux, Mac OS X і Windows.

До складу IntelliJ IDEA включені напрацювання, створені в результаті спільної роботи з компанією Google, яка використовувала IntelliJ IDEA в якості базису для свого нового відкритого середовища розробки Android Studio. Завдяки співпраці істотно розширені штатні можливості IntelliJ IDEA з розробки застосунків для платформи Android.

Огляд можливостей.

Community версія середовища IntelliJ IDEA підтримує:

інструменти для проведення тестування TestNG і JUnit;

системи контролю версій CVS, Subversion, Mercurial і Git;

засоби складання Maven і Ant;

мови програмування Java, Java ME, Scala, Clojure, Groovy і Dart.

Підтримується розробка застосунків для мобільної платформи Android.

До складу входить:

модуль візуального проектування GUI-інтерфейсу Swing UI Designer;

XML-редактор;

редактор регулярних виразів;

система перевірки коректності коду;

система контролю за виконанням завдань і доповнення для імпорту та експорту проектів з Eclipse.

Доступні засоби інтеграції з системами відстеження помилок JIRA, Trac, Redmine, Pivotal Tracker, GitHub, YouTrack, Lighthouse.

Комерційна версія "Ultimate Edition" відрізняється наявністю підтримки додаткових мов програмування (наприклад, PHP, Ruby, Python, JavaScript, CoffeeScript, HTML, CSS, SQL), підтримкою технологій Java EE, UML-діаграм, можливістю роботи з фреймворками (Rails, Grails, Google Web Toolkit, Spring, Play Java Web framework і Hibernate), засобами інтеграції з Perforce, Microsoft Team Foundation Server і Rational ClearCase[19]. JavaFX — платформа та набір інструментів для створення насичених інтернет застосунків (англ. Rich Internet Applications, RIA) з можливістю підвантаження медіа та змісту. Вперше продемонстровано Sun Microsystems на Міжнародній конференції Java розробників — JavaOne, в травні 2007. JavaFX включає в себе набір утиліт, за допомогою яких веб-розробники та дизайнери можуть швидко

створювати та надавати розвинуті інтернет застосунки для десктопів, мобільних пристроїв, телебачення та інших платформ.

5.2 Аутентифікація та авторизація

Для запобігання несанкціонованого доступу до програмного продукту було розроблено функцію аутентифікації та авторизації. Кожен користувач системи працює під своїм унікальним логіном, який видається йому адміністратором. Завдяки цьому можна визначити хто працював з системою в заданий проміжок часу.

Після запуску програмного продукту з'являється вікно авторизації (рисунок 5.1). У відповідні поля необхідно ввести виданий адміністратором логін і пароль.

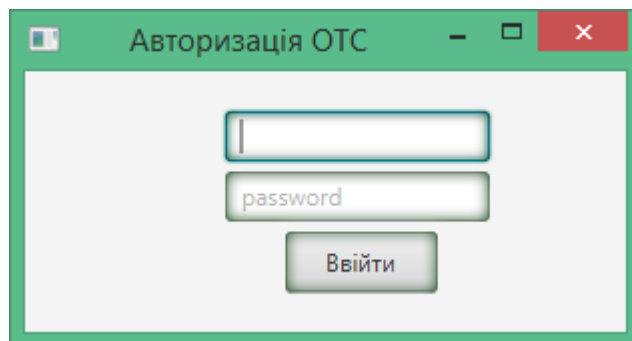


Рисунок 5.1 Вікно авторизації

У випадку, якщо аутентифікація проходить успішно, такий користувач дійсно зареєстрований в системі, проводиться перевірка паролю. Якщо останній правильний, користувач отримує доступ до основного функціоналу системи (рисунок 5.2)

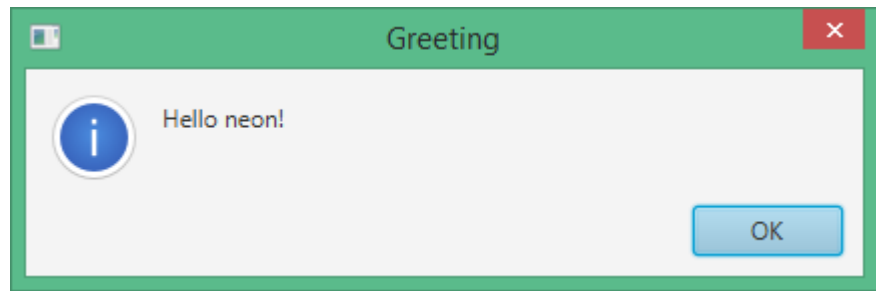


Рисунок 5.2 Вікно привітання після успішної авторизації

В разі неправильного введення логіну або пароля, або якщо користувач не зареєстрований в системі виводиться вікно, що сигналізує про заборону доступу до програмного продукту (рисунок 5.3)

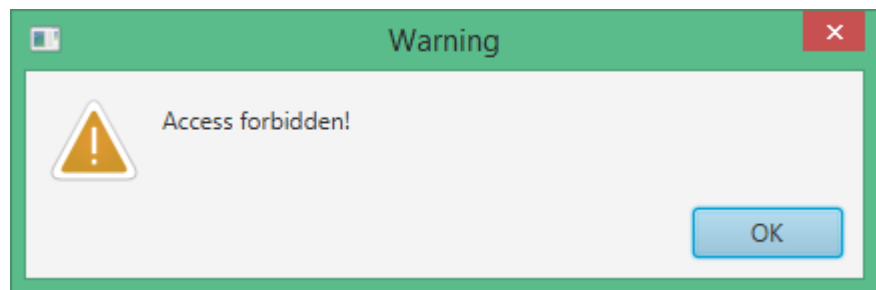


Рисунок 5.3 Заборона входу в систему

5.3 Форма вибору конструктивного елемента

Якщо все введено вірно, на екрані з'являється форма вибору конструктивного елемента, який необхідно оцінити (рисунок 5.4).

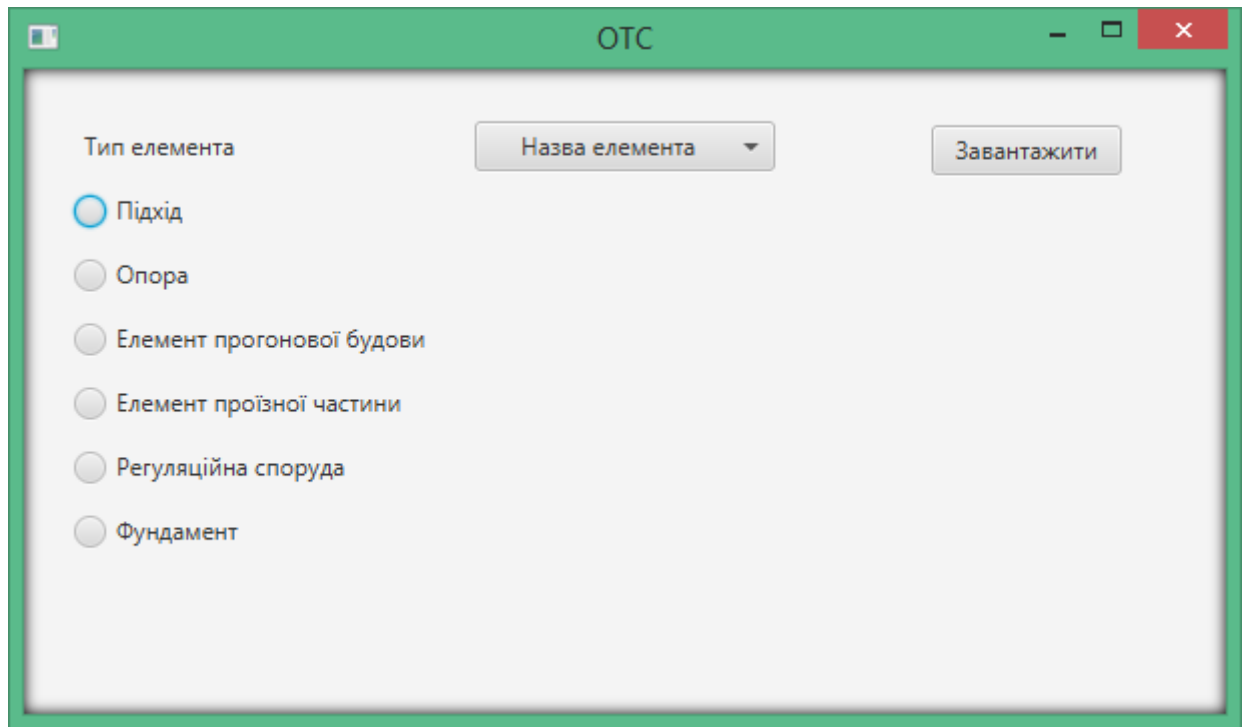


Рисунок 5.4 Форма вибору конструктивного елемента

Спочатку обираємо один з шести типів елементів мосту:

1. підхід;
2. опора;
3. елемент прогонової будови;
4. елемент проїзної частини;
5. регуляційна споруда;
6. фундамент.

Після вибору типу елемента, за допомогою випадаючого списку обираємо з довідника необхідний елемент (рисунок 5.5).

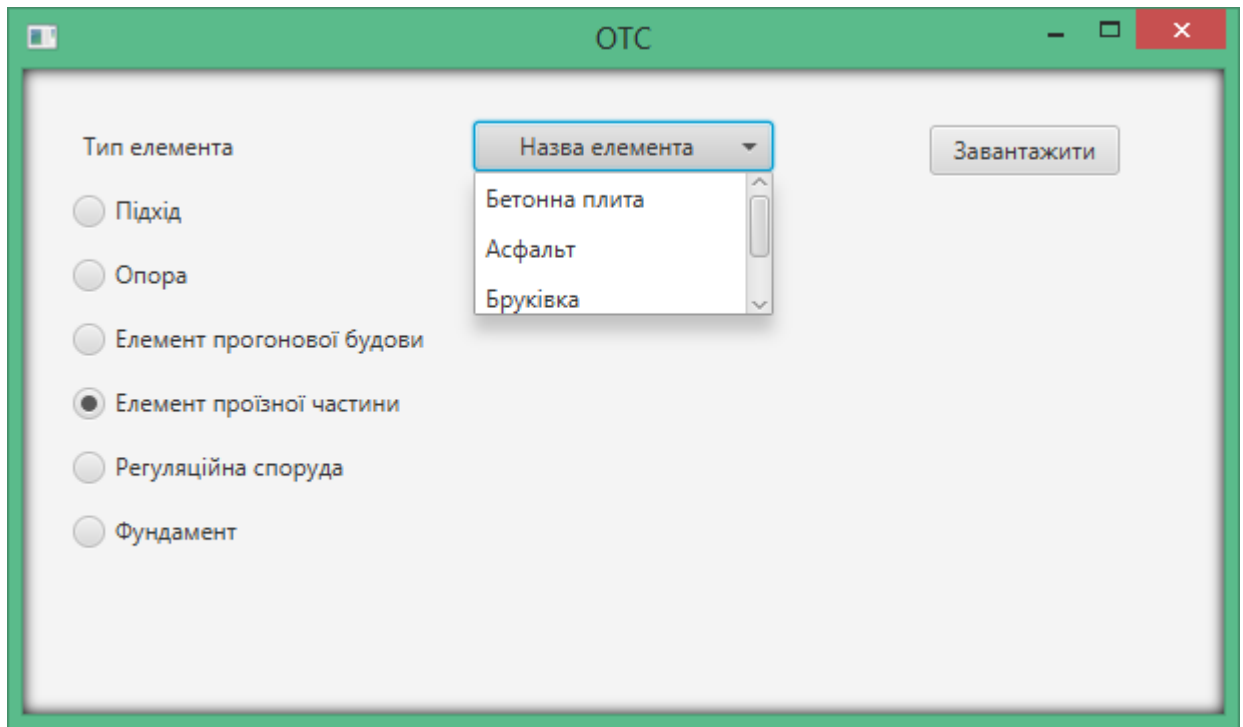


Рисунок 5.5 Вибір типового елемента

Обравши один з типових елементів, завантажуюємо його фото, клікнувши лівою клавішею миші по кнопці «Завантажити». Відкривається діалог вибору фотокартки. (рисунок 5.6).

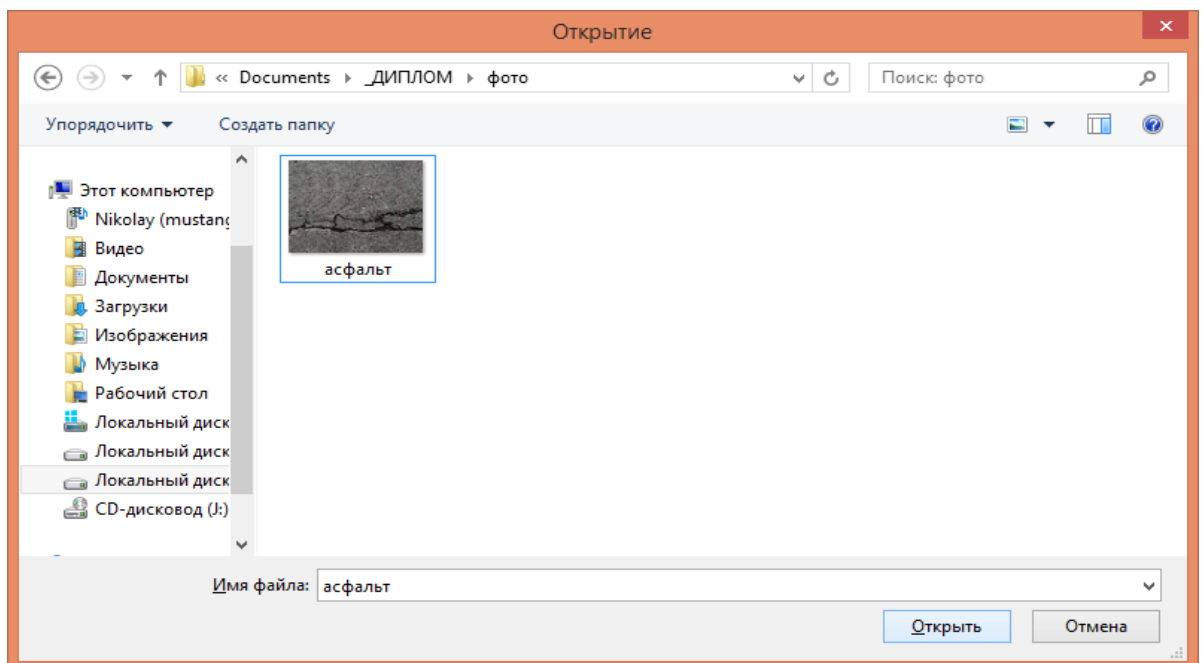


Рисунок 5.6 Діалог вибору фотокартки

5.4 Форма оцінки технічного стану

Знаходимо фотокартку, яка нам необхідна, в каталозі і натискаємо кнопку «Открыть». Після цього завантажується остання форма системи, яка дозволяє оцінити технічний стан обраного типового елемента конструкції (рисунок 5.7).



Рисунок 5.7 Форма оцінки технічного стану елемента

Отже, маємо завантажене фото поверхні оцінюваного об'єкту. Тепер вводимо площу його поверхні в метрах квадратних у відповідне поле. Також вказуємо, який дефект наявний на зображенні. Після цього, клікаючи лівою клавiшею миші по фотокартці, розставляємо реперні точки навколо дефективної зони (рисунок 5.8). Це необхідно для побудови оболонки навколо пошкодженої зони і визначення її площі.

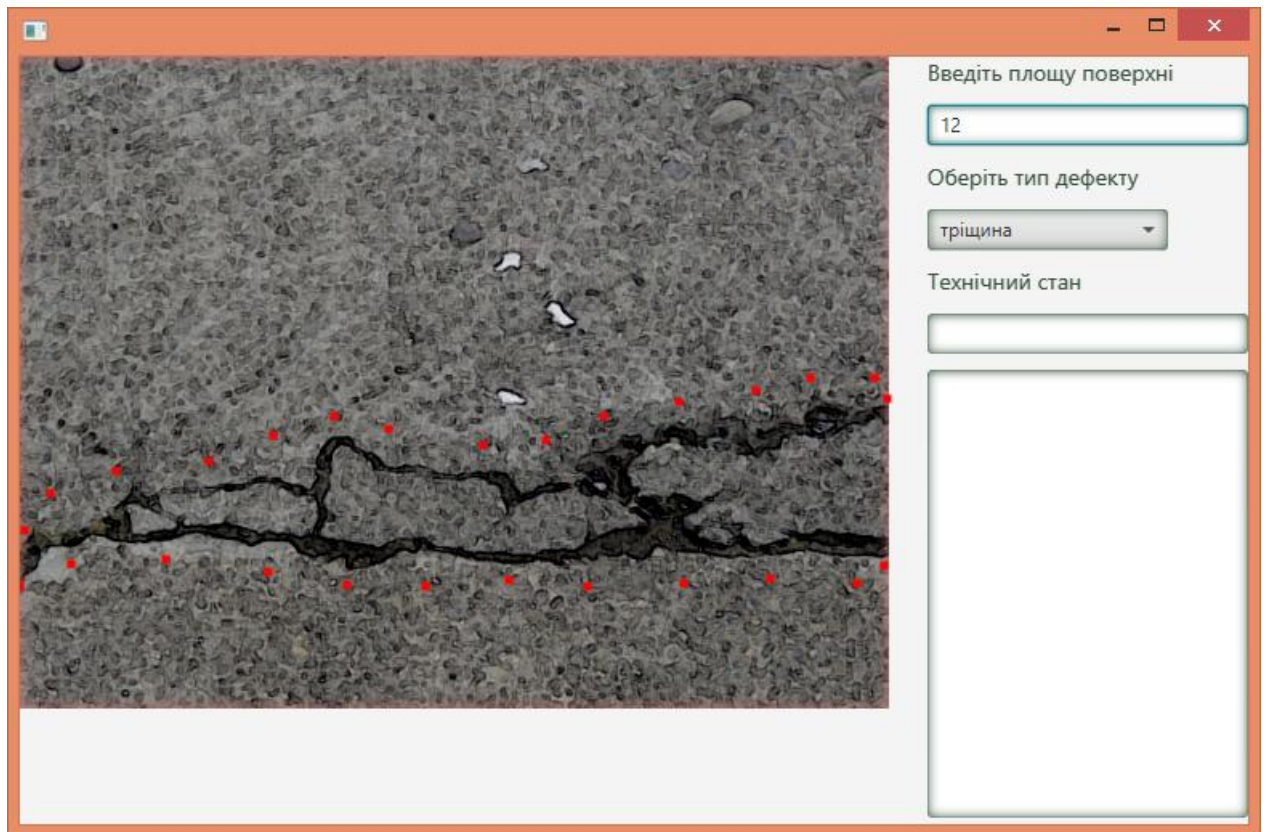
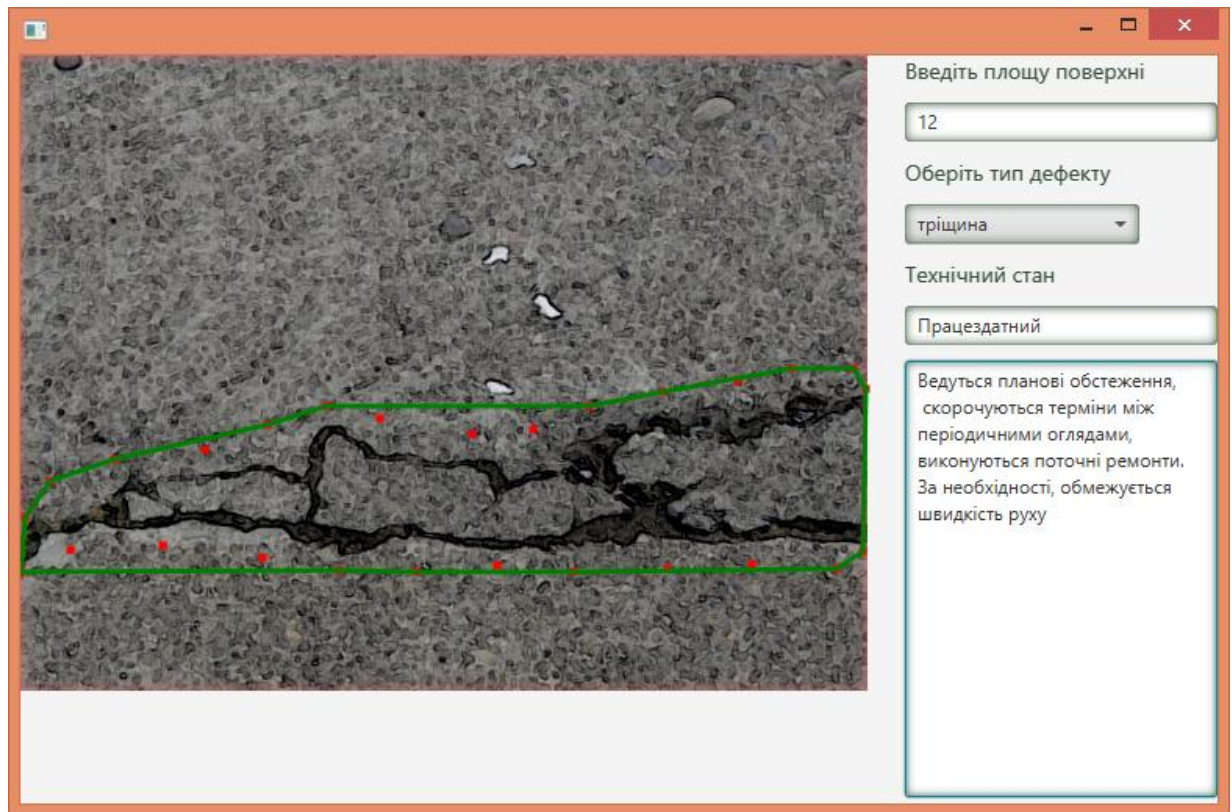


Рисунок 5.8 Розстановка реперних точок

Розставивши реперні точки, клікаємо правою клавішею миші на зображенні. Після цього починає працювати алгоритм Грехема-Ендрю. Навколо пошкодженої ділянки будується опукла оболонка (рисунок 5.9). По завершенню роботи алгоритму, дефективна зона буде охоплена оболонкою. Отже, непридатну ділянку визначено. Тепер розраховується її площа за допомогою триангуляції Делоне. Далі, враховуючи відношення площі пошкодженої ділянки до загальної площі, визначається технічний стан елемента та надається його короткий опис (рисунок 5.9).



Риснок 5.9 Результат роботи програми

Отже, в результаті роботи програмного продукту, отримуємо оцінку технічного стану об'єкта на основі його фотокартки та технічного паспорту.

Така технологія значно полегшує роботу з проведення оцінки технічного стану мостових конструкцій, а також дозволяє моделювати ситуації, до того як вони виникнуть в реальних умовах. Завдяки цьому можна уникнути багатьох аварій. Адже мостові конструкції – це місце постійного скупчення великої кількості людей. Безпека таких об'єктів повинна бути на першому місці.

ВИСНОВКИ

1. На основі експериментальних і аналітичних досліджень проведено порівняльний аналіз існуючих інформаційних технологій оцінки технічного стану. Взято до уваги плюси і мінуси кожної з них; розроблено дерева цілей та функцій, побудовано концептуальну модель системи; запроектовано діаграми потоків даних і відображено декомпозицію кожної з них.

2. Проведено аналіз основних алгоритмів побудови опуклої оболонки. В результаті порівняння було обрано алгоритм Грехема-Ендрю. Визначальними факторами у виборі стали надійність та швидкість роботи алгоритму $O(n \log n)$. У якості методу визначення площі випуклої оболонки було обрано триангуляцію Делоне.

3. Розроблено концептуальну, логічну та фізичну моделі бази даних системи. В якості СУБД використано сервер MySQL.

4. Запроектовано інтуїтивний та ергономічний дизайн користувацького інтерфейсу. Програмний продукт реалізовано в середовищі IntelliJ IDEA.

5. У результаті проведення тестування розробленої системи підтверджено її працездатність у напрямку визначення технічного стану існуючого об'єкту з можливістю його моделювання через декілька років.

6. Розроблено інформаційно-довідкову систему, що дозволить оперативно вирішувати задачі діагностики технічного стану та забезпечити зниження трудомісткості та вартості обстеження конструкцій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Інтелектуальна інформаційна технологія діагностики технічного стану будівель [Текст] : монографія /В.М. Михайленко, О.О. Терентьєв, М.І. Цюцюра // – К: ЦП «Компринт», 2015. – С. 162.
2. Моделі і методи системи діагностики технічного стану будівель [Текст] : монографія /А.О. Білощицький, П.Є. Григоровський, О.О. Терентьєв // – К: ЦП «Компринт», 2015. – С. 232.
3. Терентьєв О.О. Моделі визначення фізичного зношення конструктивних елементів будівлі для задач діагностики технічного стану / Баліна О.І., Шабала Є.Є.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 26/2016, КНУБА, 2016. – С. 153-157.
4. Терентьєв О.О. Побудова діагностичних моделей основних конструкцій будівель /Шабала Є.Є., Баліна О.І., Доля О.В.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 28/2016, КНУБА, 2016. – С. 155-159.
5. Михайленко В.М. Аналіз сучасних інформаційних методів системи діагностики технічного стану будівель /Терентьєв О.О., Шабала Є.Є.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 29/2017, КНУБА, 2017. – С. 136-143.
6. Інтегровані моделі і методи автоматизованої системи діагностики технічного стану об'єктів будівництва [Текст] : монографія /В.М. Михайленко, П.Є. Григоровський, І.В. Русан, О.О. Терентьєв // – К: ЦП «Компринт», 2017. – С. 229.
7. Моделі, методи та інформаційна технологія діагностики технічного стану будівельних конструкцій і споруд [Текст] : монографія /В.М. Михайленко, О.О. Терентьєв, Є.Є. Шабала, К.І. Київська, Є.В. Горбатюк // – К: ЦП «Компринт», 2017. – С. 161.

9. Інтегровані моделі, які забезпечують прогнозування надійності прийняття рішень для задачі системи діагностики технічного стану будівель /Терентьєв О.О., Шабала Є.Є., Саченко І.А.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 32/2017, КНУБА, 2017. С. 76-80.

10. Моделі та методи інформаційної системи діагностики технічного стану об'єктів будівництва. Підручник /В.М. Міхайленко, І.В. Русан, П.Є. Григоровський, О.О. Терентьєв, А.Т. Свідерський, Є.В. Горбатюк. – К.: Компрінт, 2018. – 325 с.:іл.

11. Olexander Terentyev The Method of Direct Grading and the Generalized Method of Assessment of Buildings Technical Condition /Mykola Tsiutsiura// – International Journal of Science and Research (IJSR), Volume 4 Issue 7, July 2015. – P. 827-829.

12. Olexander Terentyev The Method of Prediction of Deformations of Buildings and Failure Analysis the Examination of Technical Condition of Buildings /Malyna Bohdan// – International Journal of Science and Research (IJSR), Volume 4 Issue 8, August 2015. – P. 280-282.

13. Olexander Terentyev Methodology a comprehensive survey and assessment of technical condition of staircases – Scientific Journal «ScienceRise», Volume 8/2(13), August 2015. – P. 41-46.

14. Svitlana Tsiutsiura The Method of Assessing Risk Management at Various Stages of the Life Cycle for the Problem of Diagnostics of Technical Condition of Buildings /Olexander Terentyev// – International Journal of Science and Research (IJSR), Volume 4 Issue 9, September 2015. – P. 588-590.

15. Olexander Terentyev Mathematical model of the system of decision support for problem diagnostics of technical condition of building constructions – Scientific Journal «ScienceRise» №9/2(14), September 2015. – P. 35-40.

16. Olexander Terentyev Development of models and methods for determining the physical deterioration of items for the task of diagnostics of

technical condition of buildings and structures /Olexander Poltorak// – Scientific Journal «ScienceRise» №8/2(25), August 2016. – P. 14-19.

17. Olexander Terentyev Risk assessment of delayed damage diagnostics of technical condition of building structures /Olexander Poltorak// – Scientific Journal «ScienceRise» №2(31), February 2017. – P. 42-45.