

Міністерство освіти і науки України
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти бакалавра
зі спеціальності 121 – Інженерія програмного забезпечення

На тему: Комп'ютерна підтримка оперативного відновлення системи
освітлення в надзвичайних ситуаціях

Засвідчую, що в цій
кваліфікаційній роботі немає
запозичень із праць інших
авторів без відповідних
посилань.

Студент гр. ІІЗ–22-2

_____ / П. П. Журбицький/

Керівник кваліфікаційної
роботи

_____ / І. А. Котов /

Завідувач кафедри

_____ / А. М. Стрюк /

Кривий Ріг

2026

Криворізький національний університет

Факультет: Інформаційних технологій

Кафедра: Моделювання та програмного забезпечення

Ступінь вищої освіти: бакалавр

Спеціальність: 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри

_____ А. М. Стрюк

«_____» _____ 2026 р.

ЗАВДАННЯ

на кваліфікаційну роботу

студенту групи ІПЗ–22-2 Журбицькому Павлу Павловичу

1. Тема: Комп'ютерна підтримка оперативного відновлення системи освітлення в надзвичайних ситуаціях

затверджена наказом по КНУ № 285с від «28» травня 2026р.

2. Термін подання студентом закінченої роботи: «08» червня 2026р.

3. Вихідні дані по роботі: програмний комплекс для автоматизації процесу відновлення системи освітлення в надзвичайних ситуаціях.

4. Зміст пояснювальної записки (перелік питань, що їх треба розробити):
проаналізувати підходи до автоматизації процесу відновлення системи освітлення в надзвичайних ситуаціях, проаналізувати існуючі програмні системи автоматизації процесу відновлення системи освітлення в надзвичайних ситуаціях, спроектувати додаток, розробити програмне забезпечення системи автоматизації процесу відновлення системи освітлення в надзвичайних ситуаціях, здійснити тестування розробленого додатку.

5. Перелік ілюстративного матеріалу: функціональна схема, блок–схема алгоритму, зображення екранних форм додатку.

Календарний план:

№	Найменування етапів кваліфікаційної роботи	Термін виконання етапів роботи
1	Пошук науково-практичних джерел та інтернет-ресурсів з тематики роботи	02.02.26 – 09.02.26
2	Аналіз існуючих підходів і методів вирішення проблеми	10.02.26 – 23.02.26
3	Формулювання актуальності і постановка завдань роботи	24.02.26 – 02.03.26
4	Оформлення матеріалів першого розділу роботи	03.03.26 – 09.03.26
5	Розробка структурної і функціональної схем та основних алгоритмів програмного проєкту	10.03.26 – 23.03.26
6	Оформлення матеріалів другого розділу роботи	24.03.26 – 06.04.26
7	Розробка візуального інтерфейсу, баз даних і програмних модулів проєкту	07.04.26 – 20.04.26
8	Оформлення матеріалів третього розділу роботи	21.04.26 – 11.05.26
9	Оформлення додатків	12.05.26 – 18.05.26
10	Тестування розробленої програмного комплексу	19.05.26 – 01.06.26
11	Оформлення пояснювальної записки та супровідних документів	02.06.26 – 08.06.26

Дата видачі завдання: «02» лютого 2026 р.

Студент: _____ / П. П. Журбицький /

Керівник роботи: _____ / І. А. Котов /

РЕФЕРАТ

АВТОМАТИЗАЦІЯ, БАЗА ДАНИХ, БЛОК-СХЕМА, ВИПРОМІНЮВАННЯ, ІНТЕРФЕЙС, МОДЕЛЬ, ОСВІТЛЕННЯ, ПРОЕКТ, УПРАВЛІННЯ, ТЕХНОЛОГІЯ

Пояснювальна записка: 84 с., 39 рис., 1 дод., 23 джерела.

Кваліфікаційна робота: «Комп'ютерна підтримка оперативного відновлення системи освітлення в надзвичайних ситуаціях».

Метою випускової кваліфікаційної роботи є дослідження і розробка модулів програмного комплексу автоматизованої системи для оперативного світлотехнічного розрахунку з ціллю відновлення системи освітлення в надзвичайних ситуаціях.

Об'єктом дослідження і проектування є процес оперативного управління світлотехнічним розрахунком для проектування і відновлення системи освітлення в надзвичайних ситуаціях.

Теоретична частина роботи присвячена аналізу проблеми відновлення системи освітлення в надзвичайних ситуаціях, розроблена модель автоматизованої програмної системи оперативного відновлення системи освітлення в надзвичайних ситуаціях забезпечує підвищення ефективності функціонування систем освітлення в складних умовах.

Практична частина кваліфікаційної роботи полягає в розробці структурних і функціональних моделей, структур даних, алгоритмів, бази даних, моделей користувацького інтерфейсу, базових програмних модулів, програмних модулів розрахунку освітлення, програмних модулів звітів.

Аналіз розробленого програмного комплексу програмного комплексу автоматизованої системи для оперативного світлотехнічного розрахунку показує, що отримані програмні результати можуть бути використані для підвищення ефективності інформаційних систем управління системами освітлення в надзвичайних ситуаціях.

ABSTRACT

AUTOMATION, DATABASE, BLOCK DIAGRAM, RADIATION,
INTERFACE, MODEL, LIGHTING, PROJECT, CONTROL, TECHNOLOGY

Explanatory note: 84 p., 39 fig., 1 supplement, 23 sources.

Qualification work: «Computer support for operative recovery of the lighting system in emergency situations».

The purpose of the final qualifying work is to research and develop modules of the software complex of an automated system for operational lighting calculations with the aim of restoring the lighting system in emergency situations.

The object of research and design is the process of operational management of lighting calculations for the design and restoration of the lighting system in emergency situations.

The theoretical part of the work is devoted to the analysis of the problem of restoring the lighting system in emergency situations. The developed model of an automated software system for the operational restoration of the lighting system in emergency situations ensures increased efficiency of the functioning of lighting systems in difficult conditions.

The practical part of the qualification work consists of developing structural and functional models, data structures, algorithms, databases, user interface models, basic software modules, lighting calculation software modules, and report software modules.

Analysis of the developed software complex of the automated system for operational lighting calculation shows that the obtained software results can be used to improve the efficiency of information systems for managing lighting systems in emergency situations.

ЗМІСТ

ВСТУП.....	8
1 ПРОБЛЕМА ОПЕРАТИВНОГО ВІДНОВЛЕННЯ СИСТЕМИ ОСВІТЛЕННЯ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	10
1.1. Аналіз систем промислового освітлення в виробничому процесі в надзвичайних ситуаціях.....	10
1.2. Дослідження методів проєктування автоматизованих систем розробки комплексів освітлення	14
1.3. Обґрунтування завдання створення автоматизованої системи відновлення освітлення в надзвичайних ситуаціях.....	19
1.4. Розгляд сучасних систем автоматизації проєктування систем освітлення в надзвичайних ситуаціях.....	23
1.5. Формулювання комплексу задач розробки комп'ютерних засобів автоматизації оперативного відновлення системи освітлення в надзвичайних ситуаціях.....	27
2 РОЗРОБКА ФОРМАЛЬНО-ЛОГІЧНИХ І СТРУКТУРНИХ МОДЕЛЕЙ ПРОГРАМНОГО КОМПЛЕКСУ ОПЕРАТИВНОГО ВІДНОВЛЕННЯ СИСТЕМИ ОСВІТЛЕННЯ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	30
2.1. Дослідження математичних моделей розрахунку параметрів системи освітлення в надзвичайних ситуаціях.....	30
2.2. Розробка інформаційної структури автоматизованої системи оперативного відновлення системи освітлення.....	35
2.3. Розробка формально-логічної функціональної моделі програмного комплексу автоматизації.....	39
2.4. Розробка компонентів бази даних для роботи автоматизованої системи оперативного відновлення освітлення.	42

2.5. Розробка базових алгоритмів програмного застосунку системи автоматизації відновлення освітлення.....	45
2.6. Формування елементів візуального інтерфейсу програмного комплексу системи автоматизації відновлення освітлення.....	50
3. РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ ВІДНОВЛЕННЯ ОСВІТЛЕННЯ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ..	54
3.1. Розробка програмних модулів класів автоматизованої системи відновлення освітлення.....	54
3.2. Розробка програмних модулів оперативного світлотехнічного розрахунку автоматизованої системи.....	58
3.3. Реалізація бази даних для програмного комплексу автоматизованої системи.....	63
3.4. Розробка форм інтерфейсу користувача автоматизованої системи відновлення освітлення в надзвичайних ситуаціях.....	67
ВИСНОВКИ.....	71
ПЕРЕЛІК ПОСИЛАНЬ.....	73
ДОДАТОК А.....	76

ВСТУП

У сучасному світі для того, щоб повністю освітлити приміщення, необхідно провести проектне дослідження. Завдяки цьому, з одного боку, будуть дотримані норми, встановлені стандартами, а з іншого — забезпечено комфорт та задоволення користувачів приміщення. Щоб правильно провести таке проектування, необхідно дотримуватися певних процедур і прийняти певні рішення. Починаючи з вибору світильників і закінчуючи способом їх управління для досягнення найкращого результату, питання проектування стає досить складним [1].

Вибір джерела світла в останні роки є однозначним. Технологія LED завоювала ринки, оскільки забезпечує високу світловіддачу, а якість освітлення при цьому знаходиться на дуже задовільному рівні. Але що таке світлодіодна лампа? Світлодіодні лампи виготовляються з напівпровідникових матеріалів, які здатні перетворювати електричну енергію на світло з ефективністю, що часто наближається до 100%. Починаючи з початку 70-х років, коли перша світлодіодна лампа виробляла 0,1 лм/Вт, ми досягли набагато вищої ефективності, що дозволяє нам з низькими енергетичними витратами освітлювати приміщення, які раніше вимагали особливого підходу [2].

Останнім часом інтерес до природного освітлення значно зріс. І це пов'язано не стільки з якістю такого освітлення, скільки з тим благополуччям, яке воно забезпечує. Однак, оскільки рівень освітлення від природних джерел не є рівномірним, необхідна система штучного освітлення.

Системи освітлення забезпечують освітленість загальних робочих зон при одночасному зниженні рівня освітленості в сусідніх зонах. Системи локального освітлення забезпечують освітленість відносно невеликих зон, де виконуються зорові завдання. Зазвичай ці системи доповнюються освітленням загального призначення з визначеним рівнем освітленості.

Спираючись на проведений аналіз, сформульовані задачі роботи: дослідження математичних моделей розрахунку параметрів системи освітлення в надзвичайних ситуаціях; розробка інформаційної структури автоматизованої системи оперативного відновлення системи освітлення; розробка формально-логічної функціональної структури програмного комплексу автоматизації; розробка компонентів бази даних для роботи автоматизованої системи оперативного відновлення освітлення [2]; розробка базових алгоритмів програмного застосунку системи автоматизації відновлення освітлення; формування елементів візуального інтерфейсу програмного комплексу системи автоматизації відновлення освітлення; реалізація програмної системи для автоматизації відновлення освітлення в надзвичайних ситуаціях; розробка програмних модулів класів автоматизованої системи відновлення освітлення; розробка програмних модулів оперативного світлотехнічного розрахунку автоматизованої системи; реалізація бази даних для програмного комплексу автоматизованої системи; розробка форм інтерфейсу користувача автоматизованої системи відновлення освітлення в надзвичайних ситуаціях [2].

Розроблена автоматизована програмна система оперативного відновлення системи освітлення в надзвичайних ситуаціях забезпечує підвищення ефективності функціонування систем освітлення в складних умовах.

На основі отриманих результатів можна зробити висновок, що виконана робота характеризує актуальність теми, демонструє ефективність розроблених програмних модулів.

Результати роботи можуть бути використані для промисловості, установ, організацій для відновлення освітлення після аварійних відключень, а також для проектування систем освітлення з новими якостями в залежності від практичних вимог.

1 ПРОБЛЕМА ОПЕРАТИВНОГО ВІДНОВЛЕННЯ СИСТЕМИ ОСВІТЛЕННЯ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

1.1. Аналіз систем промислового освітлення в виробничому процесі в надзвичайних ситуаціях

Світло — це один із найважливіших інструментів, які були у розпорядженні людини під час розвитку її культури. Це природний подразник, що забезпечує зір. Отже, стає зрозуміло, що без джерела штучного світла людина залишилася б у буквальному та переносному сенсі в темряві.

Спочатку почали виробляти газ із вугілля. Паралельно з цим були створені системи, які спалювали газ для освітлення. Таким чином, вулиці, будинки, церкви, театри, магазини тощо швидко отримали доступ до світла, оскільки ця форма джерела світла коштувала втричі дешевше за олію. Однак незабаром й інші винахідники почали експериментувати [1].

У сучасному світі для того, щоб повністю освітлити приміщення, необхідно провести проектне дослідження. Завдяки цьому, з одного боку, будуть дотримані норми, встановлені стандартами, а з іншого — забезпечено комфорт та задоволення користувачів приміщення. Щоб правильно провести таке проектування, необхідно дотримуватися певних процедур і прийняти певні рішення. Починаючи з вибору світильників і закінчуючи способом їх управління для досягнення найкращого результату, питання проектування стає досить складним [1].

Вибір джерела світла в останні роки є однозначним. Технологія LED завоювала ринки, оскільки забезпечує високу світловіддачу, а якість освітлення при цьому знаходиться на дуже задовільному рівні. Але що таке світлодіодна лампа? Світлодіодні лампи виготовляються з напівпровідникових матеріалів, які здатні перетворювати електричну енергію на світло з ефективністю, що часто наближається до 100%. Починаючи з початку 70-х років, коли перша світлодіодна лампа виробляла 0,1 лм/Вт, ми

досягли набагато вищої ефективності, що дозволяє нам з низькими енергетичними витратами освітлювати приміщення, які раніше вимагали особливого підходу [2].

Останнім часом інтерес до природного освітлення значно зріс. І це пов'язано не стільки з якістю такого освітлення, скільки з тим благополуччям, яке воно забезпечує. Однак, оскільки рівень освітлення від природних джерел не є рівномірним, необхідна система штучного освітлення. Найпоширенішими системами освітлення є такі, рис. 1.1 [2].

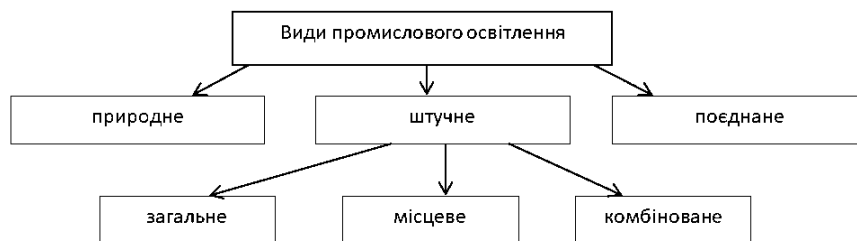


Рис. 1.1. Класифікація систем освітлення

Рівномірне загальне освітлення. У цій системі джерела світла розподіляються рівномірно, не зважаючи на розташування робочих місць. Середній рівень освітлення повинен відповідати рівню освітлення, необхідному для виконання відповідної роботи. Такі системи застосовуються переважно на робочих місцях, де немає стаціонарних робочих місць. Вони повинні мати три основні характеристики: по-перше, бути оснащені пристроями, що запобігають засліпленню (решітки, розсіювачі, відбивачі тощо) [2]; по-друге, вони повинні розподіляти частину світла на стелю та верхню частину стін; і по-третє, джерела світла повинні бути встановлені на якомога більшій висоті, щоб мінімізувати відблиски та досягти якомога більш рівномірного освітлення.

Загальне освітлення та допоміжне локальне освітлення. Це система, яка має на меті посилити схему загального освітлення шляхом розміщення світильників поруч із робочими поверхнями [2]. Світильники зазвичай створюють відблиски, тому відбивачі слід розміщувати так, щоб джерело світла не потрапляло в пряму лінію зору працівника. Рекомендується

використовувати локальне освітлення, коли вимоги до зору є критичними, як, наприклад, у випадку рівнів освітлення 1000 люксів або більше. Як правило, зір працівника погіршується з віком, що змушує підвищувати рівень загального освітлення або доповнювати його локальним освітленням. На рис. 1.2 проілюстровані ці положення [3].

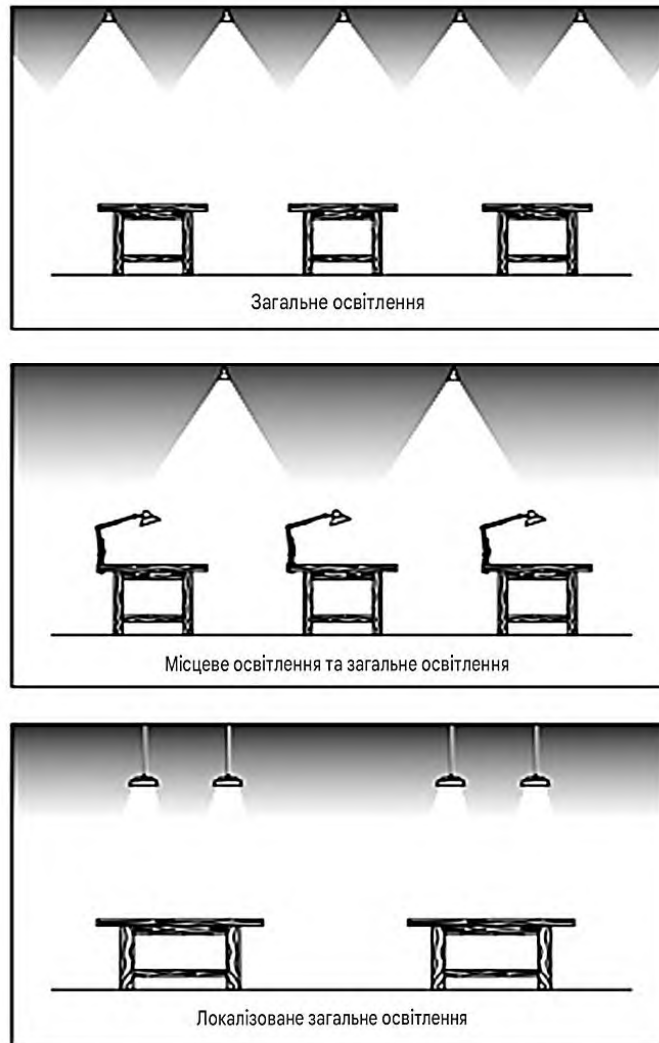


Рисунок 1.2 – Ілюстрація типів освітлення

Локальне загальне освітлення. Це тип освітлення, при якому джерела світла встановлюються на стелі та розміщуються з урахуванням двох аспектів: світлових характеристик обладнання та потреб у освітленні кожного робочого місця [2]. Воно підходить для тих приміщень або робочих зон, де необхідний високий рівень освітлення, і вимагає визначення майбутнього розташування кожного робочого місця ще до початку етапу проектування [3].

Приклад промислового освітлення приведений на рис. 1.3.



Рисунок 1.3 – Приклад системи промислового освітлення [3]

Колір: основні поняття. Правильний вибір кольору для робочого місця значною мірою сприяє підвищенню ефективності, безпеки та загального самопочуття співробітників. Так само оздоблення поверхонь та обладнання в робочому середовищі допомагає створити приємні візуальні умови та комфортну робочу атмосферу [2, 3]. Звичайне світло складається з електромагнітного випромінювання різних довжин хвиль, що відповідають кожній з смуг видимого спектра. Змішуючи червоне, жовте та синє світло, ми можемо отримати більшість видимих кольорів, включаючи білий [3]. Наше сприйняття кольору предмета залежить від кольору світла, яким він освітлюється, та від того, як сам предмет відбиває світло. Світильники можна класифікувати на три категорії залежно від кольору світла, яке вони випромінюють: теплий колір – для житлових приміщень рекомендується біле світло з червонуватим відтінком; середній колір – для робочих приміщень рекомендується біле світло; холодний колір – для завдань, що вимагають високого рівня освітлення, або для спекотного клімату рекомендується біле світло з блакитним відтінком [3].

Системи освітлення забезпечують освітленість загальних робочих зон при одночасному зниженні рівня освітленості в сусідніх зонах. Системи локального освітлення забезпечують освітленість відносно невеликих зон, де виконуються зорові завдання. Зазвичай ці системи доповнюються освітленням загального призначення з визначеним рівнем освітленості. Приклад суміщення промислового освітлення і кольору обладнання показаний на рис. 1.4.



Рисунок 1.4 – Суміщення промислового освітлення і кольору обладнання [3]

На рисунку видно типові особливості між описаними системами. При виконанні зорових завдань важливо досягти необхідного рівня освітленості та врахувати фактори, що впливають на якість освітлення [3, 4].

1.2. Дослідження методів проектування автоматизованих систем розробки комплексів освітлення

У галузі освітлення зазвичай використовують кілька фізичних величин. Основні з них такі. Світловий потік: світлова енергія, випромінювана джерелом світла за одиницю часу. Одиниця виміру: люмен (лм). Світлова сила: світловий потік, випромінюваний у певному напрямку джерелом світла,

яке не має рівномірного розподілу [3]. Одиниця виміру: кандела (кд). Рівень освітлення: рівень освітлення поверхні площею один квадратний метр, яка отримує світловий потік в один люмен. Одиниця виміру: люкс = лм/м². Яскравість або фотометрична яскравість: визначається для поверхні у певному напрямку і являє собою відношення світлової інтенсивності до поверхні, яку бачить спостерігач, розташований у тому ж напрямку (видима поверхня). Одиниця виміру: кд/м² [4]. Контраст: різниця в яскравості між об'єктом і його оточенням або між різними частинами об'єкта. Відбивна здатність: частка світла, що відбивається поверхнею. Це безрозмірна величина. Її значення коливається від 0 до 1 [4].

Розташування зон огляду на робочому місці приведені на рис. 1.5.

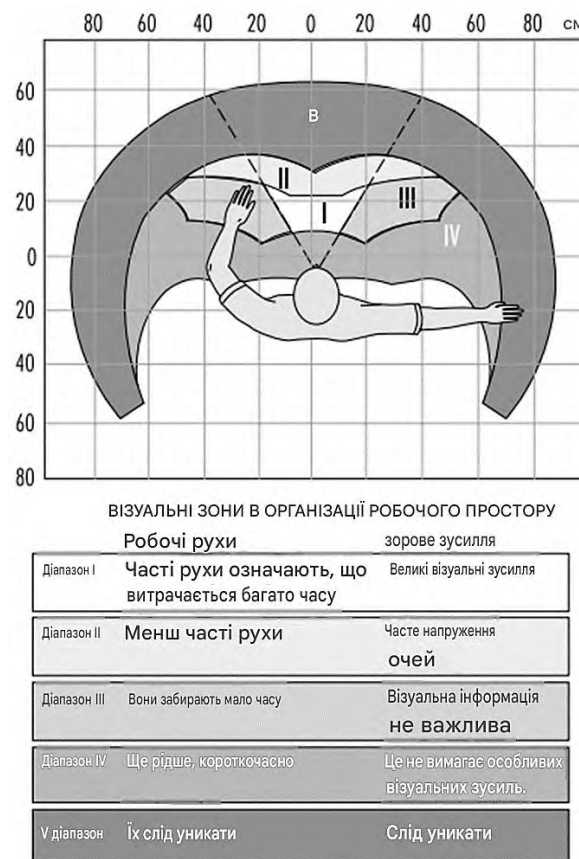


Рисунок 1.5 – Розташування зон огляду на робочому місці

Фактори, що впливають на видимість об'єктів. Рівень безпеки виконання завдання значною мірою залежить від якості освітлення та зорових можливостей [3]. Видимість об'єкта може порушуватися різними способами.

Одним з найважливіших є контраст яскравості, зумовлений факторами відбиття, тінями або кольорами самого об'єкта та факторами відбиття кольору. Око насправді сприймає різницю в яскравості між об'єктом і його оточенням або між різними частинами одного об'єкта. На рис. 1.6 наведено взаєморозподіл штучного і денного освітлення в порядку спадання [4].

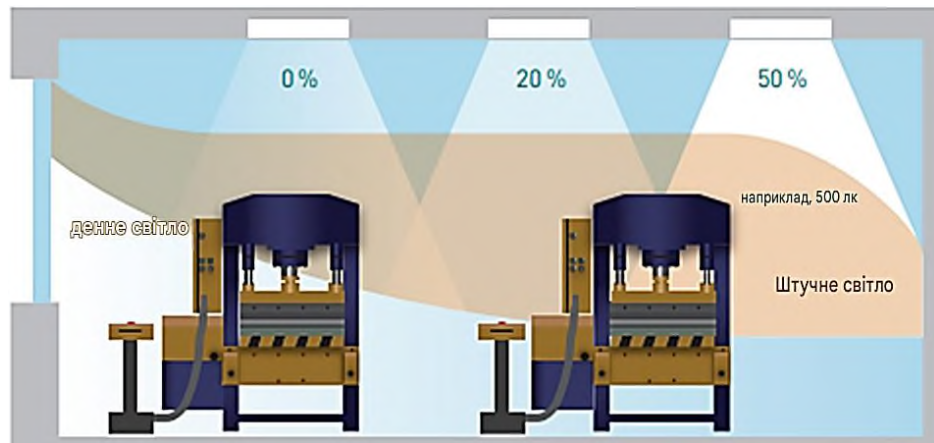


Рисунок 1.6 – Розподіл штучного і денного освітлення в порядку спадання

Коридори, приміщення для принтерів та комори, санвузли тощо часто є приміщеннями з недостатнім природним освітленням і зазвичай використовуються лише короткий час [4]. Такі приміщення повинні бути обладнані системою управління освітленням за допомогою датчиків руху, щоб не витрачати енергію на освітлення, яке не використовується. У таких системах світло зазвичай вмикається лише тоді, коли хтось перебуває в зоні дії датчика. Воно залишається увімкненим протягом заданого часу затримки.

Отже, розмір економії залежить від ймовірності перебування людей у цих приміщеннях. Встановлення датчиків руху або присутності з вбудованими комутаційними приводами чи димерами не тільки забезпечує економію коштів, а й підвищує комфорт та безпеку. Зазвичай такі пристрої управління є недорогими та простими в монтажі [5]. Однак використання простих датчиків присутності для вмикання має обмежену доцільність; іноді більш дорогі сенсорні системи з інтерфейсами керування (наприклад, для пультів дистанційного керування або кнопок) є зручнішими та добре підходять для робочих зон. Вони забезпечують більшу прийнятність серед користувачів.

Проектування будь-яких інженерних рішень сьогодні реалізується за допомогою систем САПР. Вони поділяються як показано на рис. 1.7 [5].

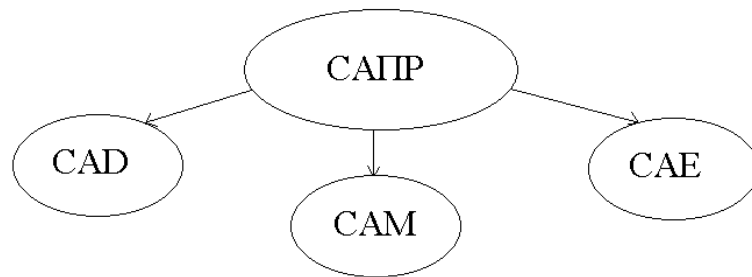


Рисунок 1.7 – Різновиди САПР

Якщо проектувальник вирішить підвищити рівень освітлення робочої поверхні за умови дотримання необхідних вимог, наприклад, з 500лк $\rightarrow$ 750лк (відповідно до ступінчастої шкали), він обов'язково повинен вибрати світильник з можливістю регулювання світлового потоку (dimmable), щоб можна було регулювати рівень освітлення в межах відповідних меж, наприклад, 750лк $\rightarrow$ 500лк [3]. Рекомендується, щоб проектування освітлення відповідало робочим поверхням для уникнення надмірного споживання. Крім того, проектувальник може спроектувати систему освітлення таким чином, щоб додаткові необхідні рівні освітлення, наприклад, додаткові 250лк, забезпечувалися за допомогою локального освітлення (наприклад, підлоговим або настільним світильником). У цьому випадку локальне освітлення включається до розрахунку потужності системи освітлення [4]. При енергетичному проектуванні локальне підвищення рівня освітлення не враховується як збільшення межі встановленої потужності.

Колірне відтворення системи освітлення перевіряється відповідно до стандарту EN 12464 за допомогою індексу кольорового відтворення (Ra або CRI) [3, 5]. Кожне приміщення має конкретні вимоги (наприклад, офіси ≥ 80), при цьому індекс коливається від 0 до 100 (точне відтворення кольорів світловим джерелом у порівнянні зі стандартним джерелом безперервного спектра). Індекс кольоропередачі Ra або CRI — це середній порівняльний показник, що виражає, наскільки точно відтворюються 8 стандартних кольорів освітленням, що випромінюється джерелом світла, порівняно зі стандартним еталонним

джерелом світла певної корельованої колірної температури. У випадку, коли показник виражає, наскільки точно відтворюються 15 типових кольорів, він позначається як Ra15 [5].

Часова модуляція освітлення. Сучасні системи освітлення значно відрізняються за ступенем коливань освітленості під час роботи (часова модуляція освітлення, Temporal Light Modulation, TLM) [4]. Це пов'язано, головним чином, зі способом використання електронної системи керування та з тим, як керують і контролюють джерело світла або світильник [5]. У той час як у світильниках старої технології спостерігалася подібна поведінка характеристик часової модуляції освітлення незалежно від виробника (наприклад, електромагнітні баласты для ламп T8), у систем світлодіодного освітлення існує значна різниця між виробниками [5].

Датчики присутності / руху. Датчики присутності автоматично вимикають освітлення, яке вони контролюють, а в деяких випадках і систему кондиціонування повітря (HVAC), коли не виявляють присутності людей у приміщеннях, де вони встановлені. Таким чином зменшується енергоспоживання будівлі. Зазвичай вони складаються з датчика руху та блоку управління. Датчик генерує та надсилає сигнал до блока управління, який, у свою чергу, вмикає або вимикає систему освітлення [6]. Більшість датчиків мають ручні та автоматичні опції для регулювання чутливості до виявлення руху та налаштування часової затримки для вимкнення системи освітлення з моменту, коли датчик - датчик не фіксує присутність людини в зоні, яку він контролює. Датчик руху збирає вхідні дані, зазвичай використовуючи ультразвук або інфрачервоне випромінювання [5]. Електронний блок управління перетворює інформацію, що надходить від датчика, і визначає, чи перебуває людина в приміщенні, використовуючи встановлений алгоритм контролю. Датчики останнього покоління мають подвійний режим роботи, оскільки використовують як інфрачервоне випромінювання, так і ультразвук. Їхня мета — зменшити помилки через неправильну роботу, вмикаючи освітлення, коли обидві технології виявляють присутність, і залишати його увімкненим, доки хоча б одна з них продовжує її виявляти [6].

1.3. Обґрунтування завдання створення автоматизованої системи відновлення освітлення в надзвичайних ситуаціях

Люди мають надзвичайну здатність пристосовуватися до свого оточення та найближчого середовища. З усіх видів енергії, які можуть використовувати люди, світло є найважливішим. Світло є невід'ємною складовою нашої здатності бачити і необхідне для сприйняття форми, кольору та перспективи предметів, що нас оточують. Більшу частину інформації, яку ми отримуємо через наші органи чуття, ми отримуємо за допомогою зору (близько 80%) [5]. І, оскільки ми настільки звикли до нього, ми сприймаємо його роботу як належне. Однак не слід забувати, що на певні аспекти людського благополуччя, такі як наш психічний стан або рівень втоми, впливають освітлення та колір речей, що нас оточують. З точки зору безпеки праці, зорові можливості та комфорт є надзвичайно важливими, оскільки багато нещасних випадків трапляються, серед іншого, через недостатнє освітлення або помилки працівника, якому важко розпізнати предмети чи ризики, пов'язані з обладнанням, транспортом, небезпечними ємностями тощо [6].

Властивістю, яка є дуже важливою для розвитку життя, яким ми його знаємо сьогодні, є властивість поглинання. Ця властивість дає предметам можливість приймати світло, поглинати його та відбивати певну частину спектра, який вони прийняли [5]. Як ми вже зазначали, світло — це випромінювання, що має певний спектр. Деякі об'єкти здатні приймати певні довжини хвиль, поглинати частину випромінювання та пропускати або відбивати іншу. Це надає предметам колір, оскільки, поглинаючи певні довжини хвиль, випромінювання, що випромінюється, стаючи біднішим, тепер має певний колір [6].

Чим більша кількість світла (але не перевищуючи певного максимального значення — межі засліплення), тим краща буде зорова продуктивність. В принципі, кількість світла з точки зору адаптації ока до виконання завдання слід визначати через величину яскравості. Яскравість

матової поверхні пропорційна добутку освітленості або рівня освітлення на цій поверхні [5]. Освітленість є прямим наслідком освітлення, а відбивна здатність є властивістю самого завдання. У певному офісі може бути багато різних завдань з різною відбивною здатністю, що значно ускладнює як попереднє вивчення перед встановленням, так і подальші вимірювання. Але освітленість залежить лише від системи освітлення і впливає на видимість [6]. Отже, для освітлення офісів кількість світла визначається в термінах освітленості і, як правило, середньої освітленості (E_{med}) на висоті робочої поверхні. Для вимірювання освітленості використовується прилад, який називається люксометром [7].

Яскравість. Це властивість, що характеризує світловий вигляд джерела світла або освітленої поверхні в заданому напрямку. Саме вона викликає у зоровому органі відчуття яскравості; більша чи менша яскравість, з якою ми бачимо однаково освітлені об'єкти, залежить від їхньої яскравості. Наприклад, книга та стіл мають однаковий рівень освітлення, проте книга бачиться чіткіше, оскільки вона має більшу яскравість, ніж стіл. Отже, можна сказати, що око сприймає саме відмінності в яскравості, а не в рівнях освітлення [7].

Фактори, які слід враховувати під час вимірювання. Під час проведення обстеження рівнів освітлення на основі вимірювання освітленості доцільно враховувати такі моменти. Люксометр повинен бути правильно відкалібрований [7]. Практично всі виробники приладів рекомендують щорічну калібрування, яка має включати перевірку спектральної чутливості та корекцію за законом косинусів. Прилад повинен бути розташований таким чином, щоб реєструвати освітленість, яку потрібно виміряти. Він може бути горизонтальним (наприклад, для визначення середнього рівня освітленості в приміщенні) або знаходитися на похилій поверхні (наприклад, на креслярській дошці). Вимірювання слід проводити в найгірших умовах або в типових умовах роботи. Необхідно вимірювати загальне освітлення та освітлення для кожного робочого місця або типового робочого місця. Сплануйте вимірювання відповідно до робочих змін, що існують на підприємстві. Завжди

слід пам'ятати про площину відліку приладу, яка зазвичай позначена безпосередньо на фотоелементі або вказана в інструкції до нього. Слід бути особливо уважними, щоб виключити з вимірювання ті джерела світла, які не належать до даної установки [6]. Також слід уникати тіней на датчик люксметра. У випадку установок з розрядними лампами важливо, щоб вони були увімкнені принаймні за двадцять хвилин до проведення вимірювання, щоб забезпечити належну стабілізацію. Зазвичай важливо зафіксувати значення напруги живлення ламп. У системах з новими розрядними лампами їх необхідно стабілізувати перед вимірюванням, що досягається після 100–200 годин роботи [7].

Оскільки світло тепер збагачується інформацією, можливо не тільки збільшити економію енергії за допомогою управління, але й включити додаткові функції, такі як можливість адаптації роботи системи освітлення відповідно до очікувань та потреб антропоцентричного освітлення (Human Centric Lighting, HCL) [7]. Технологія тепер дозволяє, крім регулювання світлового потоку світильника, змінювати видимий спектр і, відповідно, пов'язану з ним колірну температуру джерела світла. Це і є основна ідея антропоцентричного освітлення (А.О.), яке впливає на здоров'я, самопочуття та продуктивність користувачів, забезпечуючи відповідне освітлення тоді і там, де це потрібно [7]. Людиноцентричним освітленням можна визначити освітлення, яке завжди має багатогранні наслідки для людини: а) візуальні наслідки та б) невізуальні. До невізуальних наслідків належать емоційний та біологічний вплив освітлення [6].

Напрямок освітлення, тіні та колір джерел визначають вплив освітлення на приміщення. Цей комплекс параметрів, який часто описують як оптичну атмосферу, фактично визначає те, як людина сприймає атмосферу приміщення [5]. Нарешті, біологічний вплив освітлення стосується довжин хвиль видимого випромінювання, які мають біологічний вплив на людський організм. Найкращим прикладом є природне освітлення (ПО) з мінливістю інтенсивності, ритму та спектра випромінювання. Але й штучне освітлення

впливає на біологічний годинник людини, зокрема на ритм сну та пробудження, а також на багато інших функцій. При правильному застосуванні освітлення базується на динамічне змінному світловому потоці та різних спектрах протягом дня [7].

Підтвердження правильності монтажу системи освітлення має здійснюватися фахівцем, який володіє знаннями в галузі освітлення. Підтвердження, коли це необхідно, здійснюється шляхом перевірки, вимірювань та розрахунків. Перевірка повинна підтвердити правильний тип, належне розміщення та націлювання. Результатом є звіт про перевірку обладнання. Вимірювання повинні проводитися за допомогою відкаліброваного обладнання, яке матиме достатню точність для необхідного процесу та використовуватиметься відповідно до інструкцій його виробника. Вимірювання перевіряють відповідність установки та проектних значень відповідним критеріям освітлення [5]. Для кількісних критеріїв, які не можна достатньо виміряти, наприклад, засліплення, слід задокументувати значення їх розрахунку. Результатом цього є звіт, що підтверджує ефективність системи [7]. Проектувальник освітлення зобов'язаний надати програму технічного обслуговування (заміна ламп, чищення світильників тощо), якої повинен дотримуватися відповідальний за експлуатацію системи освітлення та/або електричної установки. Програма була розроблена на основі вибору коефіцієнта технічного обслуговування, а також інструкцій з експлуатації системи освітлення на етапі проектування [7]. Таким чином забезпечується збереження їх функціонування в умовах проектування та, отже, відповідно до чинних стандартів протягом усього життєвого циклу. На проміжних етапах можуть проводитися вимірювання для визначення оптимального часу технічного обслуговування системи [7].

1.4. Розгляд сучасних систем автоматизації проектування систем освітлення в надзвичайних ситуаціях

Потенціал енергоефективності застарілих світильників та ламп на підприємствах вже давно визнано. Тому багато підприємств вже розпочали перехід на більш ефективне світлодіодне освітлення (Light-emitting Diode). Однак потенціал підвищення ефективності в сфері управління освітленням часто залишається невизначеним. При цьому впровадження відповідної системи управління освітленням із датчиками денного світла або комбінованими системами управління денним світлом та присутністю може ще раз значно знизити споживання енергії [8]. В системах проектування, окрім розрахунку енергозбереження, особлива увага приділяється якості світла та критеріям закупівлі світильників і джерел світла. За результатами проектування можна уявити найважливіші аспекти планування та оптимізації освітлення. Ці знання також стануть у нагоді під час спілкування з проектувальником освітлення [7].

Процедура проведення енергоаудиту на підприємствах будується відповідно до структури стандарту енергоаудиту. Крім того, важливе значення мають енергоефективна реалізація відповідних заходів. Енергоаудит повинен проводитися поетапно, при цьому деякі етапи можуть здійснюватися паралельно. Щоб створити належні умови освітлення для різних сфер застосування і, таким чином, досягти бажаного ефекту, важливо розуміти вплив світла та освітлення на продуктивність праці, здоров'я, самопочуття, взаємодію з іншими людьми, а також (споживчу) поведінку [6]. Тому насамперед постає питання, чого саме потрібно досягти за допомогою освітлення. Адже для різних сфер застосування в промисловості та комерційній діяльності існують різні вимоги до освітлення [8].

Приклади деяких існуючих систем проектування систем освітлення приведені нижче. На рис. 1.8 приведена ілюстрація системи автоматизованого проектування систем зовнішнього освітлення «Light-in-Night Road» [8].

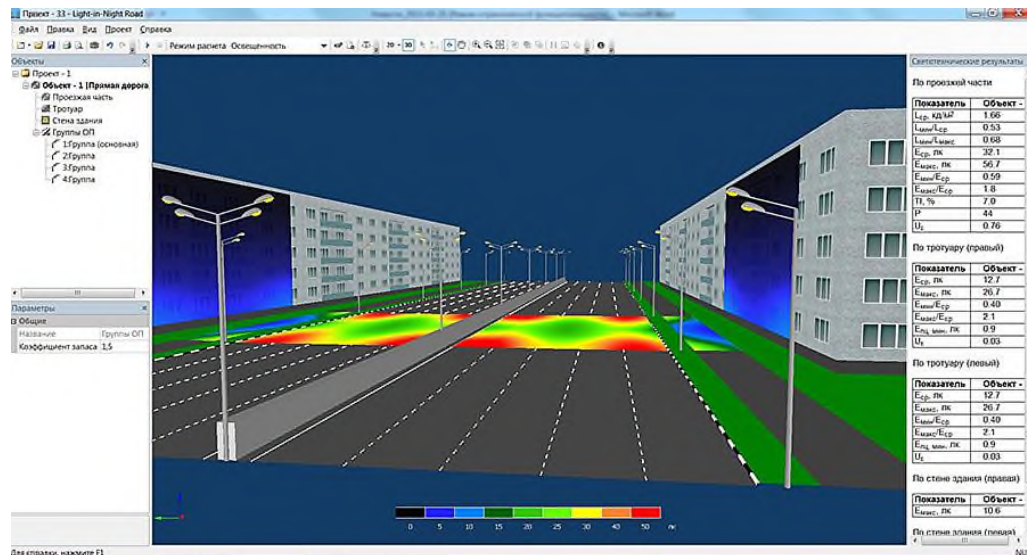


Рисунок 1.8 – Комплекс проектування систем зовнішнього освітлення «Light-in-Night Road».

Наприклад, у виробничих цехах промислових підприємств вкрай важливо забезпечити освітлення робочих місць та евакуаційних шляхів відповідно до законодавчих вимог та стандартів. Особливо під час позмінної роботи в нічні години необхідне належне освітлення! У промисловій сфері також важливу роль відіграють низькі витрати на технічне обслуговування та стійкість світильників до температур, вологи, вібрацій, хімічних речовин або вибухонебезпеки [8]. Спеціальні світлові сценарії для небезпечних зон можуть зробити вирішальний внесок у безпеку праці. У готельному бізнесі, в свою чергу, йдеться про те, щоб за допомогою світла створювати різні настрої і таким чином викликати у людей певні емоції. Для кожної зони, будь то зона прийому, номер, ванна кімната або оздоровчий центр, а також ресторан, підходить власна світлова ситуація, яку можна створити за допомогою певних світильників, ламп і, можливо, навіть власної світлової програми. Наприклад, в офісах освітлення робочого місця та відсутність засліплення є важливим критерієм якості освітлення [8, 9].

Якщо відомі індекс приміщення, коефіцієнти відбиття поверхонь та тип світильника, можна визначити коефіцієнт корисної дії світильника, коефіцієнт корисної дії приміщення і, відповідно, коефіцієнт корисної дії освітлення, які

використовуються для розрахунку кількості необхідних світильників. Коефіцієнт відбиття вказує, який відсоток світлового потоку, що падає на поверхню, відбивається [8]. Темні поверхні мають низький коефіцієнт відбиття, світлі — вищий. Білі стелі або стіни відбивають до 85 % світлового потоку. Світла дерев'яна обшивка має коефіцієнт відбиття приблизно 35 %. Чорні поверхні повністю поглинають світло. Це означає, що чим темніше приміщення, тим більше світла потрібно для досягнення необхідного рівня освітлення. На рис. 1.9 показана програмна система проектування систем освітлення «Dialux» [9].

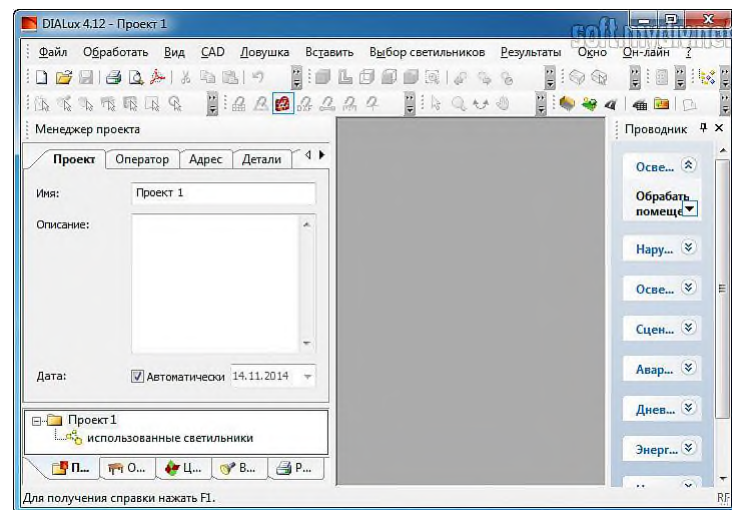


Рисунок 1.9 – Головне вікно програми «Dialux»

Якщо проектування та/або експлуатація систем освітлення на робочих місцях здійснюються виключно відповідно до стандарту, це може призвести до недотримання державних мінімальних вимог або вимог органів страхування від нещасних випадків щодо освітлення [9]. Конкретні, додаткові або відмінні від цього стандарту вимоги стосуються, зокрема: об'єднання зон зорових завдань в одну робочу зону; розширення зони безпосереднього оточення на решту приміщення; величину горизонтальної освітленості для деяких робочих місць; мінімальні значення вертикальної та циліндричної освітленості; рівномірність освітленості [8].

Перелічимо деякі стандарти, якими користуються вказані системи автоматизованого проектування [9].

Світильники та стандарти. Декларація відповідності виробника згідно з СЕ. Вимірювання електромагнітних перешкод (EMC) (EN 55015, EN 61000-3-2, EN 61000-3-3). Випробування на стійкість (EMC) (EN 61000-4-2, до 6, 8, 11, EN 61547:2009). Випробування на безпеку (LVD) (EN 60598-1, EN 60598-2-1 до 3, EN 62493). EN 62031 (Безпека світлодіодів у загальному освітленні). EN 62471 (Стандарт класифікації світлодіодних модулів за фотобіологічною придатністю / безпекою). Сертифікати ISO 9001:2008 та ISO 14001:2004 для заводу-виробника від визнаного органу сертифікації. Відповідність вимогам ROHS [10].

Всі ці стандарти враховані в наступній системі проєктування освітлення, показаної на рис. 1.10.

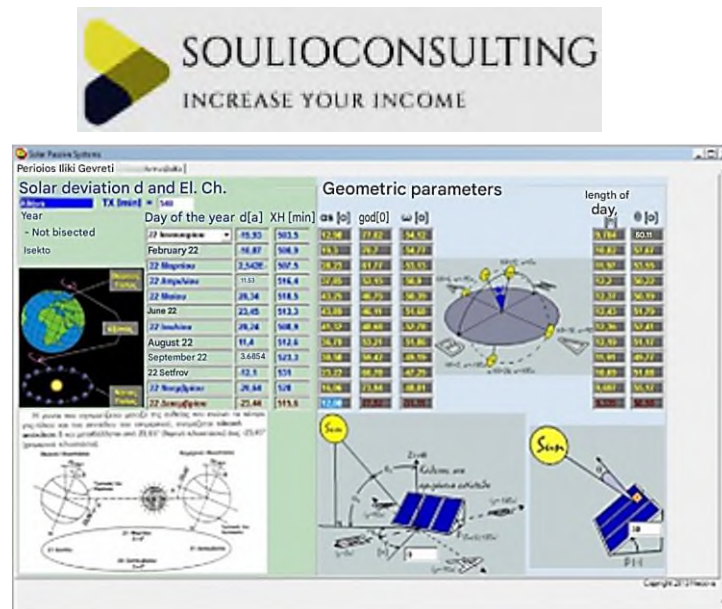


Рисунок 1.10 – Система проєктування комплексів освітлення

«SouliaConsulting»

Протягом дня для широкого спектру людських видів діяльності необхідні достатні рівні природного та штучного освітлення. Освітлення впливає як на візуальні (visual effects), так і на невізуальні ефекти (non visual effects) у людини. Рівні освітлення сприяють сприйняттю простору, а також загальному вигляду освітлених приміщень (visual effects) [10]. Освітлення робочих поверхонь може забезпечити необхідну кількість світла в

приміщеннях загалом, не переосвітлюючи при цьому периферійні/допоміжні зони [9]. Поєднання регульованого прямого освітлення робочих поверхонь та непрямого або розсіяного освітлення меншої інтенсивності в зонах, що оточують робочу поверхню, дозволяє користувачеві легше адаптуватися зорово, підвищує його гостроту зору та одночасно забезпечує належне фонове освітлення. Вимоги до освітлення робочих поверхонь та навколишнього освітлення, тобто загальних візуальних ефектів приміщення, визначені стандартами (EN 12464-1, EN 12464-2) та технічними інструкціями (TOTEE, CIE) [10].

1.5. Формулювання комплексу задач розробки комп'ютерних засобів автоматизації оперативного відновлення системи освітлення в надзвичайних ситуаціях

У сучасному світі для того, щоб повністю освітлити приміщення, необхідно провести проектне дослідження. Завдяки цьому, з одного боку, будуть дотримані норми, встановлені стандартами, а з іншого — забезпечено комфорт та задоволення користувачів приміщення [10]. Щоб правильно провести таке проектування, необхідно дотримуватися певних процедур і прийняти певні рішення. Починаючи з вибору світильників і закінчуючи способом їх управління для досягнення найкращого результату, питання проектування стає досить складним.

Вибір джерела світла в останні роки є однозначним. Технологія LED завоювала ринки, оскільки забезпечує високу світловіддачу, а якість освітлення при цьому знаходиться на дуже задовільному рівні. Світлодіодні лампи виготовляються з напівпровідникових матеріалів, які здатні перетворювати електричну енергію на світло з ефективністю, що часто наближається до 100% [11]. Починаючи з початку 70-х років, коли перша світлодіодна лампа виробляла 0,1 лм/Вт, ми досягли набагато вищої

ефективності, що дозволяє нам з низькими енергетичними витратами освітлювати приміщення, які раніше вимагали особливого підходу [8, 12].

У галузі освітлення зазвичай використовують кілька фізичних величин. Основні з них такі. Світловий потік: світлова енергія, випромінювана джерелом світла за одиницю часу. Одиниця виміру: люмен (лм). Світлова сила: світловий потік, випромінюваний у певному напрямку джерелом світла, яке не має рівномірного розподілу. Одиниця виміру: кандела (кд). Рівень освітлення: рівень освітлення поверхні площею один квадратний метр, яка отримує світловий потік в один люмен. Одиниця виміру: люкс = лм/м². Яскравість або фотометрична яскравість: визначається для поверхні у певному напрямку і являє собою відношення світлової інтенсивності до поверхні, яку бачить спостерігач, розташований у тому ж напрямку (видима поверхня) [12].

Сучасні системи освітлення значно відрізняються за ступенем коливань освітленості під час роботи (часова модуляція освітлення, Temporal Light Modulation, TLM). Це пов'язано, головним чином, зі способом використання електронної системи керування та з тим, як керують і контролюють джерело світла або світильник [11]. У той час як у світильниках старої технології спостерігалася подібна поведінка характеристик часової модуляції освітлення незалежно від виробника (наприклад, електромагнітні баласты для ламп Т8), у системах світлодіодного освітлення існує значна різниця між виробниками [12].

Чим більша кількість світла (але не перевищуючи певного максимального значення — межі засліплення), тим краща буде зорова продуктивність. В принципі, кількість світла з точки зору адаптації ока до виконання завдання слід визначати через величину яскравості. Яскравість матової поверхні пропорційна добутку освітленості або рівня освітлення на цій поверхні [12]. Освітленість є прямим наслідком освітлення, а відбивна здатність є властивістю самого завдання. У певному офісі може бути багато різних завдань з різною відбивною здатністю, що значно ускладнює як попереднє вивчення перед встановленням, так і подальші вимірювання. Але освітленість залежить лише від системи освітлення і впливає на видимість.

Отже, для освітлення офісів кількість світла визначається в термінах освітленості і, як правило, середньої освітленості (E_{med}) на висоті робочої поверхні. Для вимірювання освітленості використовується прилад, який називається люксометром [11].

Спираючись на проведений аналіз, сформулюємо задачі роботи:

- дослідження математичних моделей розрахунку параметрів системи освітлення в надзвичайних ситуаціях;
- розробка інформаційної структури автоматизованої системи оперативного відновлення системи освітлення;
- розробка формально-логічної функціональної структури програмного комплексу автоматизації;
- розробка компонентів бази даних для роботи автоматизованої системи оперативного відновлення освітлення;
- розробка базових алгоритмів програмного застосунку системи автоматизації відновлення освітлення;
- формування елементів візуального інтерфейсу програмного комплексу системи автоматизації відновлення освітлення;
- реалізація програмної системи для автоматизації відновлення освітлення в надзвичайних ситуаціях;
- розробка програмних модулів класів автоматизованої системи відновлення освітлення;
- розробка програмних модулів оперативного світлотехнічного розрахунку автоматизованої системи;
- реалізація бази даних для програмного комплексу автоматизованої системи;
- розробка форм інтерфейсу користувача автоматизованої системи відновлення освітлення в надзвичайних ситуаціях.

2 РОЗРОБКА ФОРМАЛЬНО-ЛОГІЧНИХ І СТРУКТУРНИХ МОДЕЛЕЙ ПРОГРАМНОГО КОМПЛЕКСУ ОПЕРАТИВНОГО ВІДНОВЛЕННЯ СИСТЕМИ ОСВІТЛЕННЯ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

2.1. Дослідження математичних моделей розрахунку параметрів системи освітлення в надзвичайних ситуаціях

Якісне освітлення здатне перетворити місця, де ми живемо та працюємо, на щось більше, ніж просто робоче або дозвілєве середовище. Завдяки вдалому світловому дизайну можна створити надзвичайно приємну, майже чарівну атмосферу, при цьому не забуваючи про те, що освітлювальні системи мають бути енергоефективними [11]. Параметри, що визначають якість освітлення, залежать від його призначення (театри, концертні зали, аудиторії, музеї тощо), але в будь-якому випадку вони повинні відповідати певним загальним вимогам. Рівень освітлення: необхідна освітленість (рівень світлового потоку (люкс), що падає на поверхню). Розподіл яскравості в полі зору. Обмеження засліплення [12]. Моделювання: обмеження контрасту світла та тіні, що створюється системою освітлення. Колір: колір світла та колірне відтворення. Естетика: вибір типу освітлення, ламп та світильників. Дотримання всіх цих параметрів дозволить досягти гарного світлового дизайну, не забуваючи при цьому, що правильний вибір кількості та якості освітлення залежить від простору, який буде освітлюватися, та діяльності, що в ньому відбуватиметься [12]. Перевірка того, чи є рівень освітлення в певному просторі достатнім, стає основним завданням архітектора, якщо він прагне створити простори, що вражають з точки зору освітлення. Ознайомившись із важливістю досягнення правильного рівня освітлення завдяки точній кількості ламп та світильників, стандарти рекомендують основні елементи, що входять до складу системи освітлення. Джерело світла або тип використовуваної

лампи: розжарювальна, люмінесцентна, газорозрядна. Світильник. Він контролює світловий потік, що випромінюється джерелом, і, за необхідності, запобігає або мінімізує засліплення. Системи управління та регулювання світильника. Важливо також знати про основні величини, такі як світловий потік, світлова сила, освітленість та яскравість, а також про їхні основні відмінності [11]. Після визначення цих елементів можна приступити до розрахунків, щоб оцінити, чи є рівень освітлення в приміщенні достатнім.

Розглянемо математичні моделі проектування систем освітлення [13].

Враховуємо участь прямого сонячного світла — E_M і E_H . Значення коефіцієнта відбиття, що позначається в формулах як e , знаходиться з виразу

$$e_M = \left(\frac{E_M}{E_H} \right) * 100\%. \quad (2.1)$$

На рис. 2.1 показані варіанти урахування кутів падіння сонячного випромінювання [12].

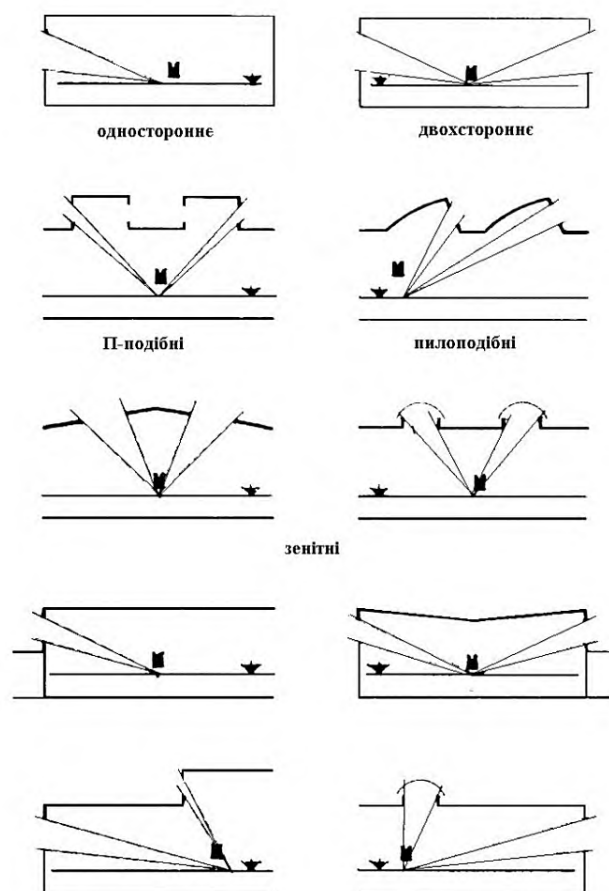


Рисунок 2.1 – Варіанти урахування кутів падіння сонячного випромінювання

Поряд з e в розрахунках природного освітлення застосовується геометричний коефіцієнт відбиття, що позначається ε . Він відрізняється від e тим, що не враховує вплив скління, і світла, відбитого фасадами будівель, землею і внутрішніми поверхнями приміщення з певною обробкою, а також не враховує нерівномірного яскравості небосхилу [13].

$$e_M = e_H + e_{\text{пр.б}} + e_{\text{під.ш}} + e_B, \quad (2.2)$$

де e_M – розрахунковий коефіцієнт природного освітлення (КПО) в даній точці приміщення; e_H – КПО від прямого світла неба (ε_H); $e_{\text{пр.б}}$ – КПО за рахунок відображення природного світла від фасаду протилежних будинків (ε_6); $e_{\text{під.ш}}$ – частка КПО за рахунок відображення від підстиляючого шару (альbedo землі); e_B – частка КПО в результаті багаторазових відображень від внутрішніх поверхонь приміщення [13].

В математичних моделях враховують також прорізи бічного освітлення і типи вікон, як те показано на рис. 2.2 [12].

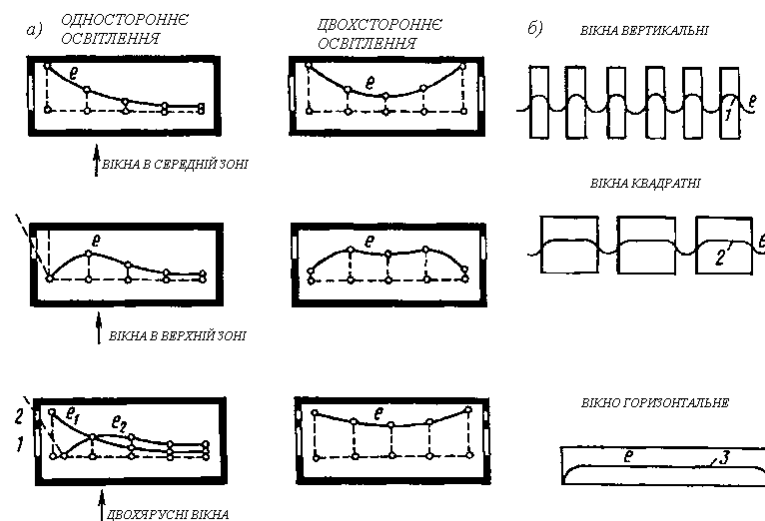


Рисунок 2.2 – Класифікація прорізів бічного освітлення і типи вікон

Розглянемо розрахунок природного освітлення за характером світлорозподілу. Індекс приміщення визначається [13]:

$$i = \frac{(B + A)}{h} (A + B), \quad (2.3)$$

де A – довжина приміщення, B – ширина приміщення, h – розрахункова висота приміщення.

На рис. 2.3 зображені варіанти розподілу зовнішнього освітлення при різних варіантах конфігурації приміщень [14].

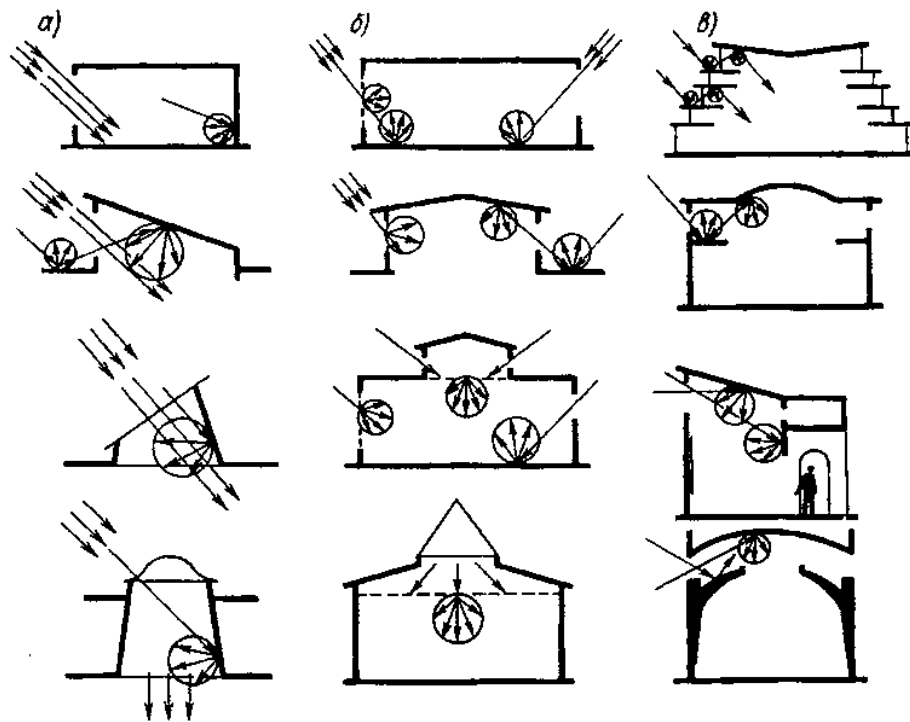


Рисунок 2.3 – Варіанти розподілу зовнішнього освітлення при різних варіантах конфігурації приміщень

Необхідний потік кожного світильника визначається наступним чином.

$$A = E * K_z * S * \frac{z}{N} * \eta, \quad (2.4)$$

де E - задана мінімальна освітленість; K_z - коефіцієнт запасу; S - освітлювана площа, м²; z - коефіцієнт нерівномірності освітлення = 1,1 ÷ 1,2; N - число світильників (намічене до розрахунку) [14].

При розрахунку люмінесцентного освітлення світловий потік обраних ламп Φ_l відомий і визначено кількість ламп в світильнику n .

$$\Phi = E * K_z * S * \frac{z}{N} * \Phi_l * \eta, \quad (2.5)$$

Потужність загальної лампи визначають наступним чином [13].

$$P = w \frac{S}{N}, \quad (2.6)$$

де w - питома потужність; S - площа приміщення; N - число світильників.

Світловий потік лампи в кожному світильнику визначається так.

$$\Phi = 1000 * E_y * \frac{K_3}{\mu} * \sum E_y, \quad (2.7)$$

де μ - коефіцієнт, що враховує дію «віддалених» світильників (1,1 ÷ 1,2); $\sum E_y$ - сумарна умовна освітленість в контрольній точці; E_y - окремого світильника.

Розрахувати або спроектувати природне освітлення значить знайти сумарну площу світлових прорізів (вікон або ліхтарів), що забезпечують нормативне значення КПО в проектуваному приміщенні. Розрахунок проводиться без урахування меблів, устаткування та інших затінюючих пристроїв [14].

Розрахунок ведуть за такими формулами.

Для бічного освітлення

$$100 * (S_o / S_n) = (KПО * K_3 * \eta_o * K_{3d}) / (\tau_o * r_1). \quad (2.8)$$

Для верхнього освітлення [15]:

$$100 * (S_{\phi} / S_n) = (KПО * K_3 * \eta_{\phi}) / (\tau_o * r_2 * K_{\phi}), \quad (2.9)$$

де $S_{o,l}$ – сумарна площа вікон і ліхтарів, м²; S_n – площа підлоги; КПО – нормативне значення КПО; K_3 – коефіцієнт запасу; $\eta_{o,l}$ – світлові характеристики вікон і ліхтарів; τ_o – загальний коефіцієнт світло пропускання; $r_{1,2}$ – коефіцієнти, які враховують відбиття світла при бічному і верхньому освітленні; K_{3d} – коефіцієнт, який враховує затінення вікон протилежними будинками; K_{ϕ} – коефіцієнт, що враховує тип ліхтаря.

Для розрахунку загального освітлення у приміщеннях можна використовувати два методи [15]. Метод люменів, який також називають «загальною системою» або «методом коефіцієнта використання». Метод люменів — це дуже практичний і простий спосіб розрахунку середнього рівня освітленості в системі загального освітлення. Він забезпечує середню освітленість з похибкою $\pm 5 \%$ і дає дуже приблизне уявлення про потреби в освітленні. Метод «точка за точкою» (або метод точкової освітленості). Цей метод використовується, якщо ви хочете дізнатися значення освітленості в конкретних точках [15].

2.2. Розробка інформаційної структури автоматизованої системи оперативного відновлення системи освітлення

Для розробки інформаційної структури автоматизованої системи оперативного відновлення системи освітлення застосовуємо дослідницький метод, що складається з трьох етапів, використовуючи конкретні приклади та виводячи на їх основі загальне рішення [16]. Ми будемо виходити з трьох наборів реальних вимог до програмної системи та інструкцій щодо визначення функціональних можливостей, що мають значний вплив на структуру системи: «Скасування операції», «Зворотний зв'язок щодо прогресу» та «Налаштування». Етапи, що були визначені, представлені на рис. 2.4.

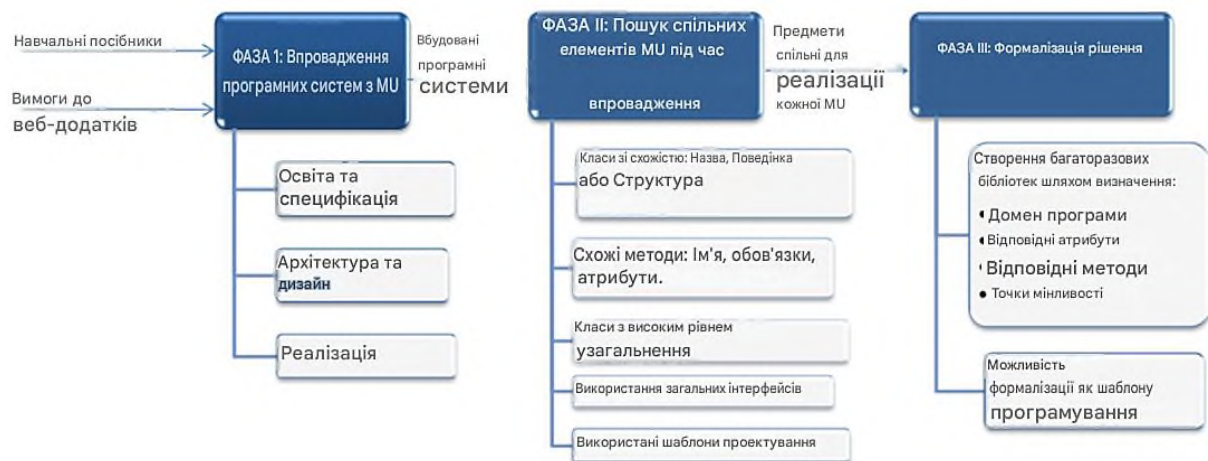


Рисунок 2.4 – Архітектура системи автоматизованого проектування системи освітлення

Етап I. Вхідними даними для цього етапу є керівництва з визначення функціональних можливостей та набір реальних вимог до застосунку. Програмні системи створюються таким чином, щоб, окрім власної функціональності, кінцева система відповідала функціональним можливостям, пов'язаним із конкретним використанням. Протягом усього процесу зберігається простежуваність функціональності, при цьому в кожному створеному артефакті виділяються елементи, пов'язані з кожним елементом [15]. При виведенні та специфікації вимог функціональні можливості кожного застосунку позначаються різними кодами. При

проектуванні компоненти, пов'язані з компонентами, позначаються іншим кольором як у діаграмі класів, так і в діаграмах послідовностей. Під час реалізації ретельно документується код, щоб можна було відрізнити той, що пов'язаний з функціональними можливостями зручності використання.

Етап II. Вихідними даними для цього етапу є три модуля, створені на етапі I, які містять функціональні можливості трьох обраних моделей [16]. На цьому етапі визначаються спільні елементи в реалізаціях кожної структури для трьох прикладів, а також встановлюється, які з них можуть бути повторно використані. Шукаються спільні елементи для трьох реалізацій на рівні визначення вимог, рішень щодо дизайну, відповідальності, класів, взаємодій, атрибутів та коду. Результатом цієї фази є елементи, які вважаються такими, що можуть бути повторно використані для реалізації функціональних можливостей зручності користування. Завдяки використанню об'єктно-орієнтованого проектування та програмування функціональні можливості, пов'язані з моделлю, інкапсульовані в окремі класи кожного застосунка [16]. Це дозволяє, серед іншого, визначити тип використовуваного проектування, відокремити елементи, пов'язані з конкретною моделлю, та визначити тип взаємодії між функціональними можливостями кожного блоку та власними функціональними можливостями застосунка.

Етап III. Вхідними даними для цього етапу є загальні та повторно використовувані елементи, виявлені на етапі II. На цьому етапі результати оформлюються у вигляді шаблонів. Для кожної структури пропонується єдиний дизайн у вигляді шаблону проектування, а попередні реалізації адаптуються до використовуваних мов програмування [16]. Поєднання запропонованого дизайну та реалізованих кодів формалізується як шаблони програмування. У рамках третьої фази також виокремлюються загальні фрагменти коду з метою створення бібліотеки класів або бібліотеки, що містить компоненти, які можна повторно використовувати та які полегшують і забезпечують включення структури під час побудови програмної системи [16].

Викладені принципи проектування проілюстровані на рис. 2.5.



Рисунок 2.5 – Принципи поетапного проектування інформаційної структури застосунку

Архітектура застосунка для показників визначається як багаторівнева тришарова архітектура: рівень збереження даних, бізнес-рівень та веб-рівень. Для управління збереженням даних використовується об'єктний шаблон [17]. Цей шаблон дозволяє абстрагувати та інкапсулювати механізм доступу до даних застосунка та вирішує проблему об'єктно-реляційної несумісності. У бізнес-шарі або шарі послуг реалізовано шаблон «Фасад», що дозволяє об'єднати функціональність високого рівня та створити єдину точку входу в додаток.

Інтерфейс відображає інформацію для користувача. У цьому випадку він здійснює асинхронні виклики за допомогою технологій AJAX, які вставляються в певну область форми [16]. Контролер отримує вхідні дані або запити, які користувач надсилає через веб-сторінку, реалізує обробку подій та перетворює вхідні дані на запити до візуалізації або структури за допомогою технологій AJAX. Утиліти, це функції на стороні клієнта, які використовуються для перевірки полів та спеціальних характеристик відображення. Вони відповідають за здійснення асинхронних викликів до сервера, спілкуються з фасадом для запити інформації з бази даних, яка повертається у вигляді XML-файлів, а також запитують необхідні для

перетворень, виконують перетворення та повертають інформацію до представлення для відображення. Також запитує статичні XML-файли, які не отримуються з бази даних, а попередньо визначені. Інтерфейс відповідає за приховування та групування деталей логіки та взаємодії з персистенцією [17]. Він спілкується з компонентом Result, щоб запитувати інформацію з бази даних відповідно до отриманого запиту; інформація повертається у вигляді XML-файлів. Result - це компонент, відповідальний за надсилання конкретного запиту на інформацію та створення відповідних об'єктів VO (Value Object). Він виконує методи кожного VO, що дозволяють сформувати XML-файл. Загальна структура взаємодії інформаційних модулів комплексу управління системою освітлення показана на рис. 2.6.

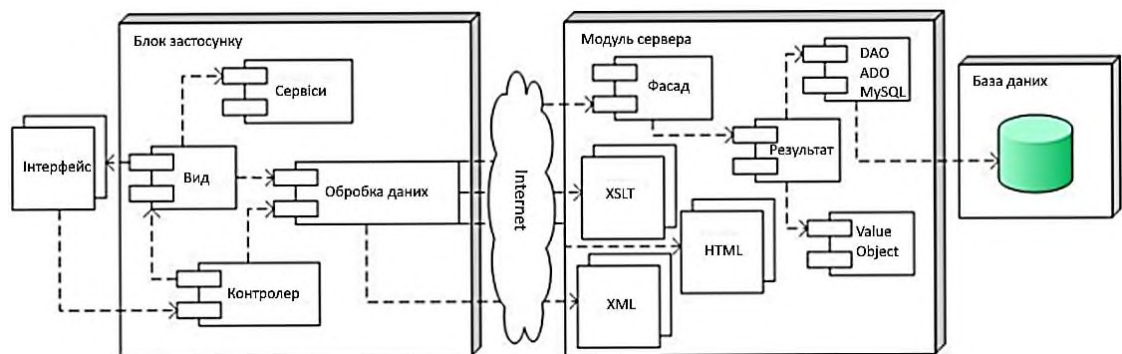


Рисунок 2.6 – Структура взаємодії інформаційних модулів комплексу управління системою освітлення

Програмний шар реалізовано з використанням архітектурного шаблону MVC. Для кожного активного представлення визначається контролер, а також контролер для меню, яке завжди залишається активним. Як представлення, так і контролер можуть взаємодіяти з моделлю через фасад за допомогою технологій AJAX [16]. На рисунку 2.6 показано схематичне зображення, яке не включає функціональні можливості всієї структури управління. Для кожного об'єкта структури існуватиме активний вигляд – ActiveView та пов'язаний з ним контролер, який є активним контролером – ActiveController [17].

2.3. Розробка формально-логічної функціональної моделі програмного комплексу автоматизації

Для розробки формально-логічної функціональної структури програмного комплексу автоматизації та розрахунку параметрів системи освітлення визначено трирівневу архітектуру: рівень збереження даних, рівень управління даними та бізнес-рівень [18]. Для управління збереженням даних використовуються компоненти .NET під назвою TableAdapter, рівень управління може бути реалізований за допомогою Forms.NET, а бізнес-рівень складається з файлів Code-Behind — класів, пов'язаних із Forms, що містять їхню функціональність. Функції кожного компонента такі. Form Events/ViewState представляє інформацію користувачеві, це XML-код, згенерований сервером, і приймає вхідні дані або запити через обробку подій, надсилає необхідні запити до структури; відповідає тому, що в архітектурі MVC є представленням та контролером [17, 18]. Сервіси, це функції на стороні клієнта, що використовуються для перевірки полів та спеціальних характеристик відображення. Forms, це класи, що успадковують System.UI.Page, які містять код на мові.NET для моделювання інтерфейсів застосунка. Code-Behind, це класи, що реалізують бізнес-логіку, пов'язану з кожною Form. NET., це компоненти зв'язку даних, що дозволяють встановити зв'язок між елементом управління, який відображає дані (gridView, TextView тощо), та логікою, що їх управляє [18]. System, це базовий компонент .NET, який надає більшість просторів імен (Namespaces), що відповідають бібліотеці базових класів мови. System. Це компонент, в якому містяться всі типи, необхідні для програмування додатків та сервісів .NET. Практичний рівень збереження даних може бути реалізований локально з використанням TableAdapters — компонентів, що надаються технологією Active Data Objects (ADO) платформи .NET. ADO забезпечують доступ до реляційних баз даних, XML-документів та додатків для роботи з даними. ADO .NET — це набір класів, що надають функції доступу до даних [18].

Загальна формально-логічна функціональна схема послідовності дій системи, яка проєктується, зображена на рис. 2.7.

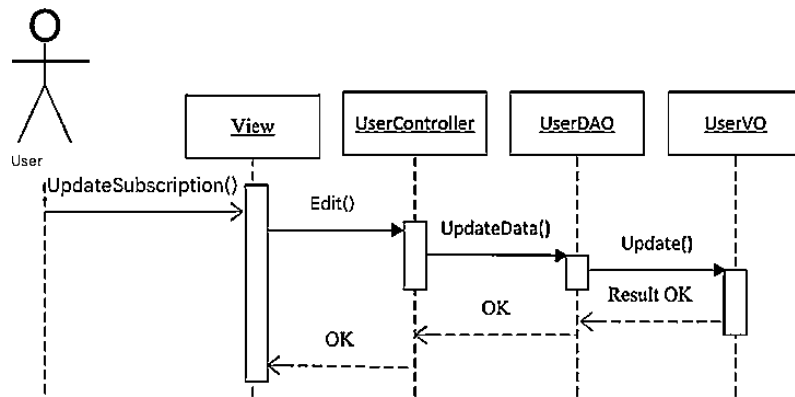


Рисунок 2.7 – Загальна формально-логічна функціональна схема послідовності дій системи

При створенні об'єкта TableAdapter використовуються один або кілька запитів, які заповнюють структуру під назвою DataTable. TableAdapter може містити кілька запитів, що заповнюють одну й ту саму структуру, з єдиним обмеженням: дані, які повертають запити, повинні відповідати схемі пов'язаної таблиці даних [18]. Крім того, можна створювати скрипти вставки, оновлення або видалення, пов'язані з тією самою структурою. Для цього застосунка буде створено лише три TableAdapter, кілька запитів для заповнення даних та один скрипт оновлення. Головна сторінка застосунка — це сторінка пошуку, яка дозволяє отримати доступ до інформації про показники за датою або за обладнанням. Для цього передбачені фільтри за запчастинами, конструкцією та проєктом [17, 18]. Щодо показників за період, можна здійснювати пошук за проєктом та періодом, а також перевіряти дані за базою, проєктувальником та періодом. На сторінці результатів пошуку щоденних змінних за датою відображається інформація про електропостачання на обрану дату [18]. Виходячи з загальної формально-логічної функціональної схеми розроблена функціональна схема оперативного відновлення системи освітлення в надзвичайних ситуаціях, яка показана на рис. 2.8.



Рисунок 2.8 – Функціональна схема оперативного відновлення системи освітлення в надзвичайних ситуаціях

Опишемо сутність функціональної схеми. Функція — це дія, що передбачає виконання одного або декількох кроків у програмі, кожен із яких вимагає взаємодії з користувачем [19]. Ця дія призводить до зміни стану програми, чи то через зміну інформації в базі даних, зміну параметрів налаштувань, чи то через зміну змінних програми або сеансу у випадку застосунків. Прикладами операцій є дії, що виконуються з діалогових вікон, форм або майстрів, які включають кілька кроків. Команда — це інструкція або наказ, який користувач дає застосунку через єдину взаємодію, натискаючи кнопку, посилання, вибираючи пункт меню або будь-яку іншу опцію, що надається застосунком. Виконання команди не обов'язково передбачає зміни у стані застосунка, оскільки вона може використовуватися лише як засіб навігації по ньому [18].

2.4. Розробка компонентів бази даних для роботи автоматизованої системи оперативного відновлення освітлення.

Рівень збереження даних було реалізовано з використанням шаблону DAO. Об'єкти DAO абстрагують та інкапсулюють механізм доступу до даних і спираються на єдиний клас DBMySQL, який централізує з'єднання з базою даних. Інформація, у свою чергу, інкапсулюється в об'єкти, які відповідають за представлення даних [19]. Для кожного набору даних існують класи, які, з одного боку, є об'єктами структури системи, а з іншого — пов'язані з таблицею в базі даних. Класи DAO розширюють загальний клас Error, що дозволяє обробляти та повідомляти про можливі помилки підключення, запиту або оновлення в базі даних. Класи DAO викликаються відповідним класом Result з бізнес-шару [19].

Нарешті, отримуємо сутно-реляційну модель бази даних. Можна сказати, що крім таблиць, які представляють дані об'єктів структури застосунка, існує допоміжна таблиця `ids_Tables`, яка використовується для збереження останнього послідовного номера, присвоєного певним таблицям, у яких не можна використовувати поля, що генеруються автоматично базою даних, оскільки це імпортована інформація і необхідно зберегти ідентифікатор джерела [18].

Для реалізації функцій у додатку використовується jQuery — набір бібліотек, що надають низку функціональних можливостей, спрощуючи взаємодію з застосунком, а також обробку подій. Сторінки застосунка мають базову структуру, що складається з чотирьох розділів. Розділ App-top є загальним для всього застосунка; він відображає опції «Головна», що дозволяє перейти на головну форму застосунка, а також «Довідка» та «Вийти», які ведуть відповідно на сторінки довідки та авторизації [19]. Розділ Global-menu показує користувача, який наразі авторизований, та надає доступ до двох основних модулів застосунка: «Індикатори освітлення» та «Адміністративне управління проєктом освітлення». У розділі «Динамічне меню»

- норми освітлення промислових приміщень;
- стандарт освітлення промислових приміщень ISO8995;
- склад програмного комплексу (де і які файли розташовані)

Інформація про проєкт відображається разом із даними про реєстрацію, здійснену проєктувальником [20]. Це дозволяє контролювати повноту та точність проєкта, а також відображає дані про призначені ресурси з розбивкою на деталі. Ці дані система розраховує на основі параметрів проєкту, що вказують час початку відновлення освітлення та закінчення, а також дату. Окрім інформації, яку система може обчислити на основі запланованих заходів по відновленню освітлення, вона дозволяє додавати понаднормові чинники, які можуть бути денними або нічними, гнучкими або додатковими, залежно від типу аварійного стану освітлення [20]. Система контролює можливі варіанти, які можуть бути зроблені для проєкта відповідно до його типу. Нарешті, система періодично генерує інформацію для розрахування ресурсів проєкта відновлення. Цей модуль генерує змінні, необхідні системі для розрахунку.

Для визначення вимог до системи проведено збір інформації у підприємств, а отримані вимоги задокументовано. Щодо управління показниками, документація результатів етапу аналізу здійснюється з використанням трьох конструкцій: подання випадків використання, статичне подання структури системи з діаграмою класів та подання взаємодії з діаграмами послідовностей [19]. Головна форма застосунка — це сторінка встановлення параметрів проєкта відновлення освітлення, яка дозволяє отримати доступ до інформації про показники проєкта. Для цього передбачені списки параметрів проєкта освітлення. На формі результатів розрахунків відображається інформація про параметри всіх проєктів, у яких запланована зміна обладнання [21]. Запит параметричних змінних відображає інформацію про проєкт, який вже сформовано або закрито, і дозволяє експортувати результат. Перевірка дозволяє перевірити, які аспекти проєкта залишаються невиконаними за певний період.

2.5. Розробка базових алгоритмів програмного застосунку системи автоматизації відновлення освітлення

Розробка алгоритмів програмного застосунку системи автоматизації відновлення освітлення базується на шаблонах проектування [21]. Як і для звичайного проєкта, для застосунка, який проєктується, представлено один шаблон проектування, орієнтований на реалізацію. Потребується формалізація шаблону проектування застосунка [21]. У програмному компоненті «Структура» створюється дизайн, а оскільки це шаблон проектування, орієнтований на реалізацію, у розділі «Реалізація» покроково вказується, як кодувати запропонований дизайн. Функціональність отримається на основі визначення вимог.

Опишемо загальний підхід до проектування програмних модулів. Створити компонент у вигляді віджета для відображення індикатора прогресу та кінцевого повідомлення користувачеві у випадках, коли це передбачено. Віджет або елемент управління — це елемент графічного інтерфейсу користувача (GUI), що відображає інформацію; він може працювати як невелика самостійна програма, але в даному випадку функціонує як процес у межах програми [20]. Створити клас, який буде виконувати роль обробника основних функцій застосунка: періодично запитувати про хід виконання завдання або процесу у компонента на сервері (для багатопотокової системи) або отримувати інформацію про хід виконання від клієнтського компонента (однопотокові системи), оновлювати інформацію індикатора прогресу та закривати індикатор прогресу, коли це необхідно [22]. У випадках, коли можна отримати інформацію про хід виконання процесу під час його виконання на сервері (багатопотокові системи), створити на сервері клас, якому процес повідомляє про свій хід і який можна перевіряти з боку клієнта. Створіть клас, який підтримує та надає інформацію про те, чи процес залишається активним.

На основі цих принципів можна синтезувати загальний алгоритм розробки застосунку по відновленню освітлення, який показано на рис. 2.10.

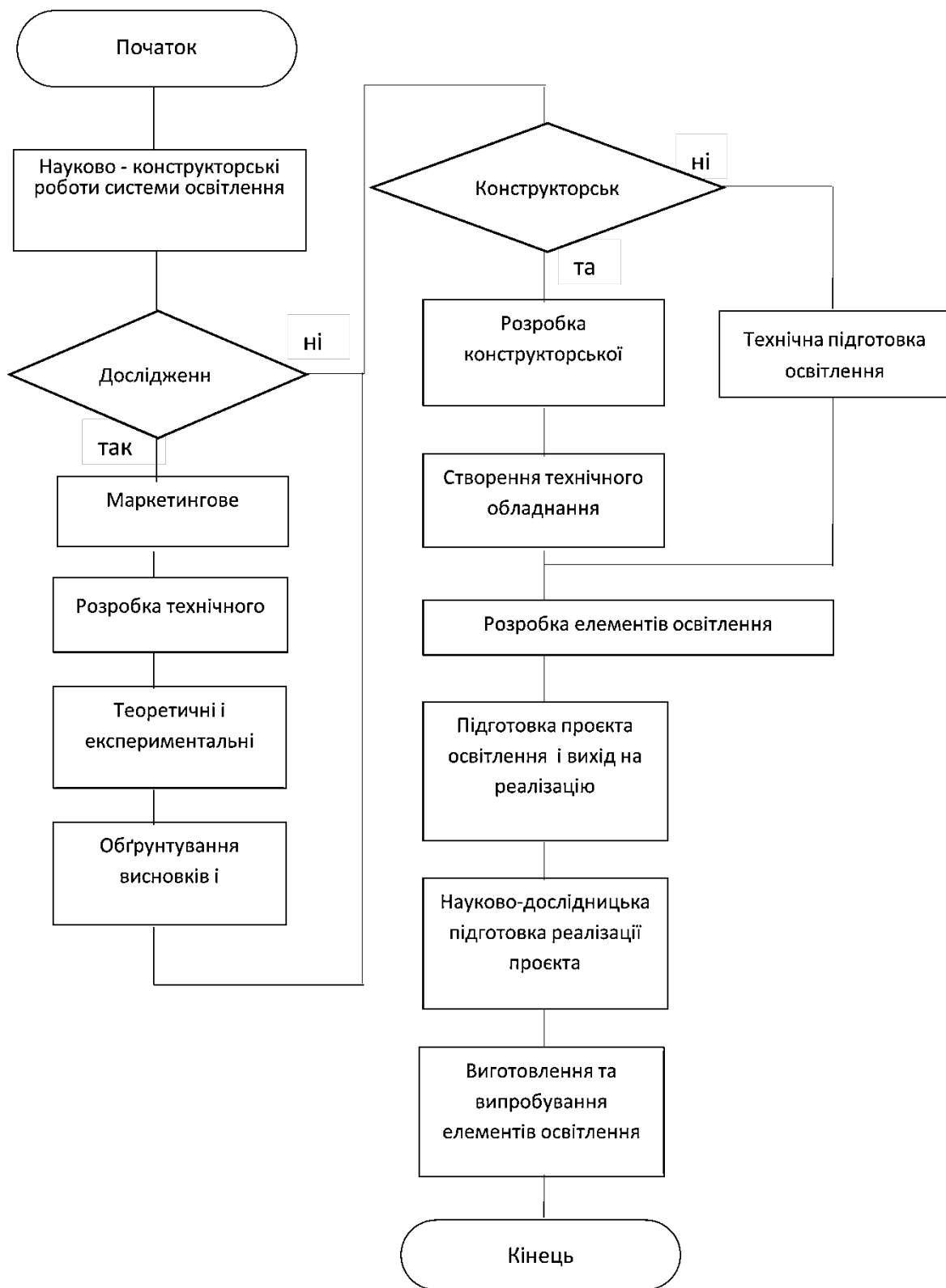


Рисунок 2.10 – Блок-схема алгоритму розробки системи відновлення освітлення

Створіти компонент, який буде працювати як фасад для всіх компонентів, що використовуються при реалізації функціональності зворотного зв'язку про хід виконання. Реалізувати відповідну функціональність у додатку, щоб у кожному процесі, який триває більше двох секунд, запускався індикатор з відповідними параметрами [21]. Реалізувати відповідну функціональність у додатку, щоб у багатопотокових системах компонент оновлювався з урахуванням прогресу, який вважається відповідним, або у випадку однопотокових систем здійснювалися виклики до сервера таким чином, щоб можна було надавати інформацію через функцію `SetParam`. Реалізуйте відповідну функціональність, щоб під час скасування процесу через надсилалося повідомлення для оновлення відповідної інформації у віджеті та деактивації процесу. Реалізуйте відповідну функціональність, щоб система закривала індикатор прогресу, надсилаючи повідомлення або відповідь у випадках, коли це необхідно [21].

Приведемо наступні кроки розробки системи. Створити компонент у вигляді віджета для відображення індикатора прогресу та кінцевого повідомлення користувачеві у випадках, коли це передбачено. Віджет або елемент управління — це елемент графічного інтерфейсу користувача, що відображає інформацію [20]. Створити клас, який буде виконувати роль обробника основних функцій: періодично запитувати про хід виконання завдання або процесу у компонента на сервері або отримувати інформацію про хід виконання від клієнтського компонента, оновлювати інформацію та закривати індикатор прогресу, коли це необхідно створити клас на сервері, якому процес повідомляє про свій хід і який можна перевіряти з боку клієнта. Створіть клас, який підтримує та надає інформацію про те, чи процес залишається активним. Створіть компонент, який буде працювати як фасад для всіх компонентів, що використовуються при реалізації функціональності зворотного зв'язку про хід виконання [22].

Отже, тепер можна розробити алгоритм розрахунку штучного освітлення, який показано на рис. 2.11.

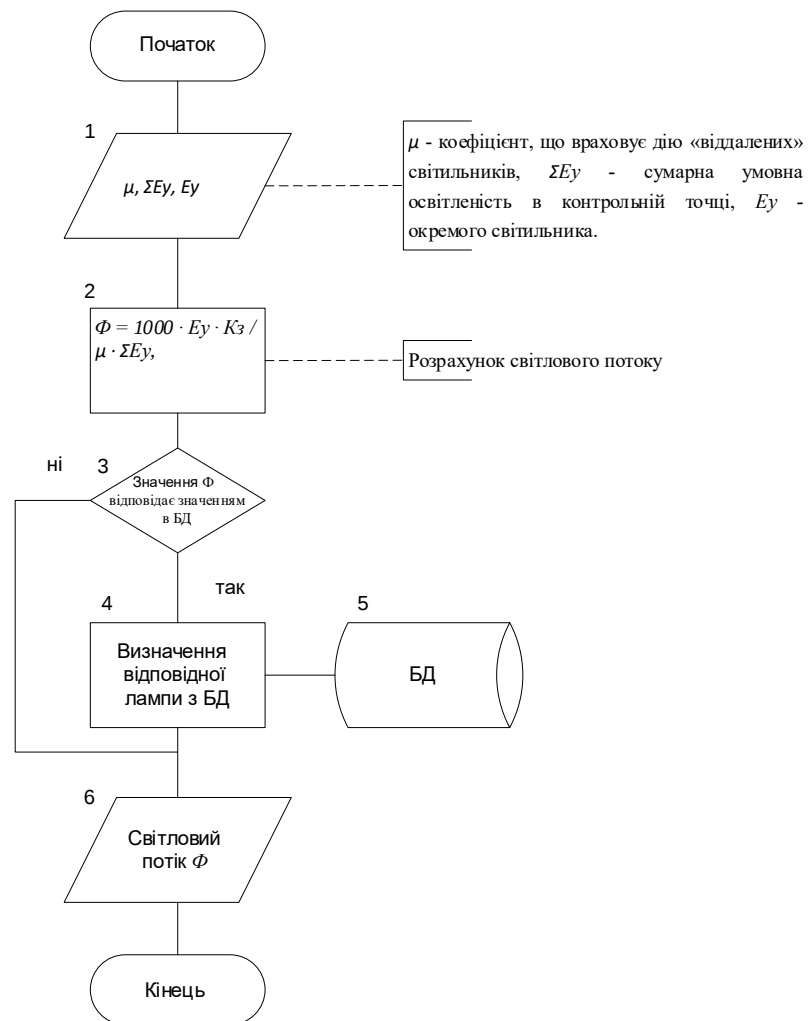


Рисунок 2.11 – Блок-схема алгоритму розрахунку штучного освітлення точковим методом

Розробимо алгоритм розрахунку природнього освітлення. Для цього необхідно реалізувати відповідну функціональність у додатку, щоб у кожному процесі, який триває більше двох секунд, запускався індикатор прогресу з відповідними параметрами [21]. Реалізувати відповідну функціональність у додатку, щоб у багатопотокових системах компонент оновлювався з урахуванням прогресу, який вважається відповідним, або у випадку однопотокових систем здійснювалися виклики до сервера таким чином, щоб можна було надавати інформацію про прогрес. Реалізувати відповідну функціональність, щоб під час скасування процесу надходило повідомлення для оновлення відповідної інформації [22].

Алгоритм розрахунку природнього освітлення приведений на рис. 2.12.

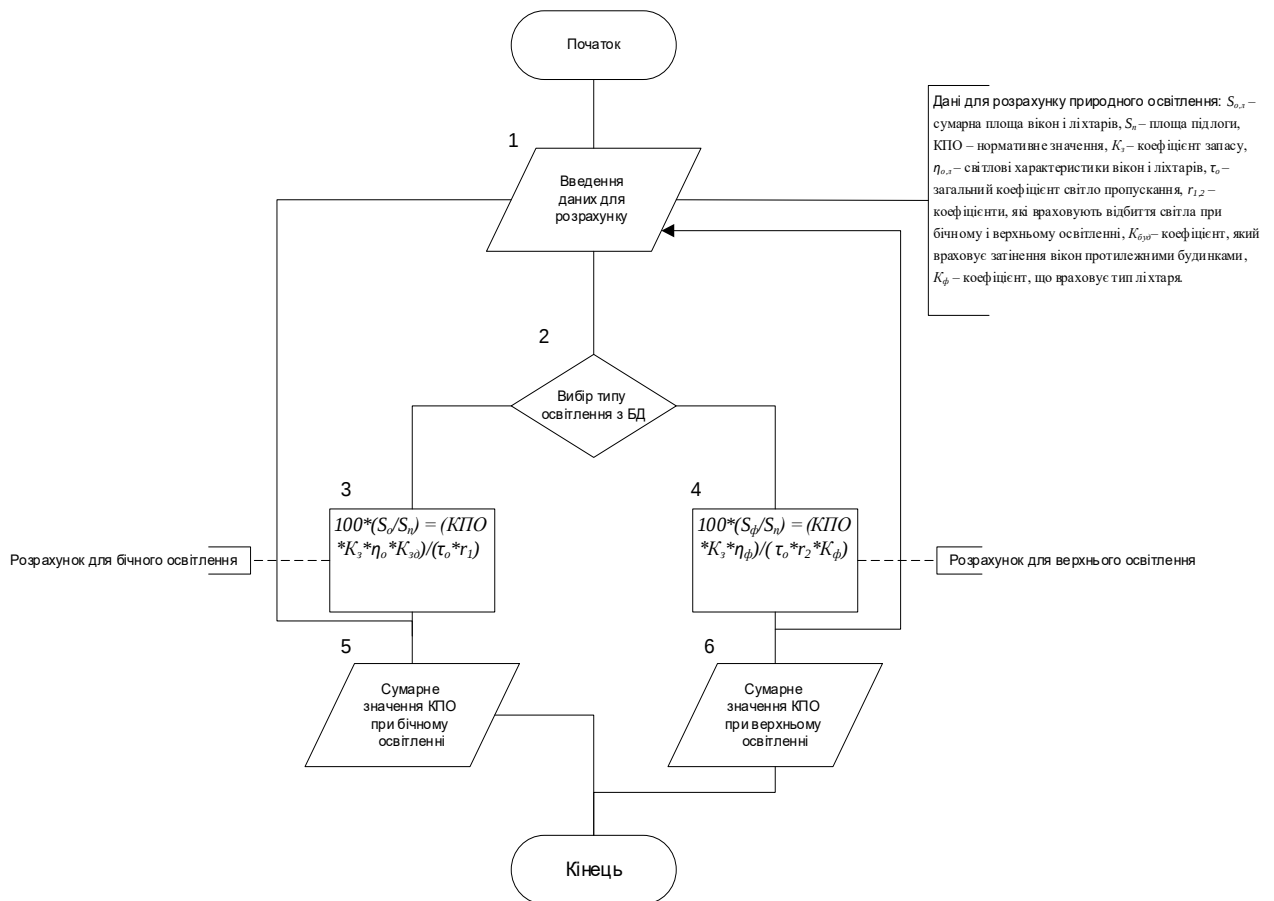


Рисунок 2.12 – Блок-схема алгоритму розрахунку природного освітлення

У цьому додатку бізнес-шар має два основні компоненти: фасад під назвою та набір об'єктів-результатів — по одному для кожного класу, визначеного в структури системи, які й взаємодіють із системою збереження даних [23]. При цьому, необхідно реалізувати відповідну функціональність, щоб система закривала індикатор прогресу, надсилаючи повідомлення або відповідь у випадках, коли це необхідно. Також необхідно реалізувати відповідну функціональність у кожній частині програми, де можна вносити зміни, щоб у разі зміни чогось у програмі оновлювався стан [21].

На рисунку 2.12 показано алгоритм і те, як процес, що виконує роль фасаду, слугує інструментом розрахунку і єдиною точкою входу до бізнес-процесу, а клас приховує та групує деталі логіки й взаємодіє з системою, ідентифікує, який варіант і яка операція запитується з боку застосунка, та запитує дані у відповідного об'єкта, який поверне інформацію [23].

2.6. Формування елементів візуального інтерфейсу програмного комплексу системи автоматизації відновлення освітлення

Комп'ютери та, загалом, більшість портативних і стаціонарних пристроїв є сьогодні потужним інструментом сучасного суспільства, покликаним надавати підтримку в більшості сфер нашого повсякденного життя. Їхня роль часто стає визначною, оскільки вони є засобом, що забезпечує належну інфраструктуру для функціонування безлічі додатків, з якими ми взаємодіємо на систематичній основі [23]. Окрім таких можливостей, ці інструменти забезпечують легку та швидку комунікацію, інформування, розваги, а також сприяють творчості їхніх користувачів. Вони також можуть, після відповідної обробки, надавати користувачеві цільовий зворотний зв'язок щодо результатів низки його дій. Загальні принципи організації функціонування візуального інтерфейсу можуть бути зображені наступною схемою, рис. 2.13 [21].

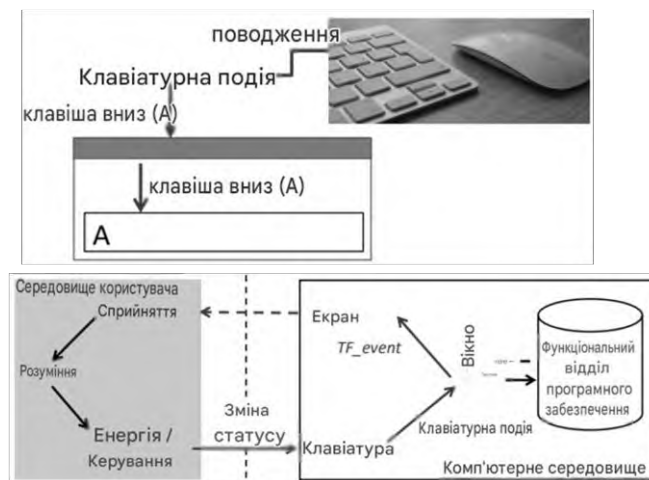


Рисунок 2.13 – Принципи подієвої організації функціонування візуального інтерфейсу

Основним кольором застосунка є сірий та його різні відтінки, які використовуються для розмежування окремих елементів інтерактивних екранів застосунка. Цей вибір був зроблений після вивчення візуального стилю промислових застосунків. Вони пов'язані, головним чином, із підтримкою

строгого стилю роботи і взаємодії. Відповідно, додаток використовує той самий колір, щоб забезпечити увагу користувача на роботі та створити позитивне й плідне відчуття. У підході також використовується кілька відтінків сірого, який також є кольором, пов'язаним із діловими емоціями, для візуального контрасту [22]. Таким чином, створюються відповідні умови для того, щоб забезпечити увагу оператора під час візуального контакту з застосунком, незалежно від того, чи використовує він його вперше, чи ні, а також підтримувати цей інтерес на високому рівні протягом усього часу використання [19].

Прототип форми вибору типу проєкта освітлення приведений на рис. 2.14.

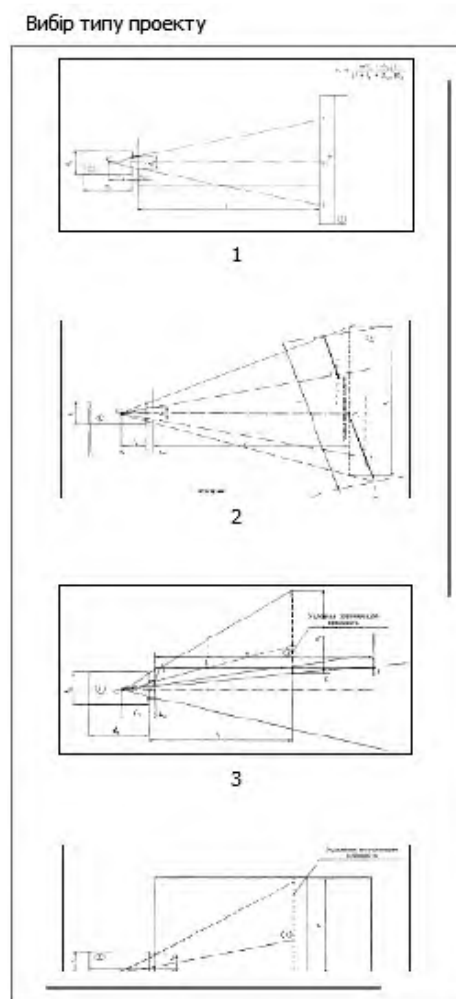


Рисунок 2.14 – Прототип форми вибору типу проєкта освітлення

Галузь програмної інженерії присвятила значну частину своїх досліджень розробці стандартних і систематизованих процесів, що забезпечують розробку та проектування програмних продуктів [20]. Ці процеси описують важливі етапи

розробки продукту і відомі як структури розробки або як життєвий цикл програмних систем. Функція цих моделей полягає у наданні рамки для розуміння процесу розробки, з конкретними припущеннями та відповідним керівництвом щодо дій та співпраці команди розробників [22].

Проектування UI належить до категорії розробки програмного забезпечення за моделями (прототипами), а саме, у багатьох випадках підходить до структури антропоцентричного проектування (Human-Centred Design). Модель антропоцентричного дизайну — це творчий підхід до процесу розробки програмного забезпечення, а також до вирішення проблем, що виникають під час цієї розробки, і на практиці означає значну залученість користувачів у процес проектування та створення мультимедійних додатків [22]. Основні причини такого підходу пов'язані, головним чином, із зручністю використання та покращеною ергономікою додатків, що адаптуються до особливостей користувачів, які будуть ними користуватися. Більш детально, зручність використання програми є необхідною для її прийняття та успіху, тоді як покращена ергономіка досягається завдяки оптимізації зручності використання пристроїв та інструментів, що знаходяться у розпорядженні користувача [23].

Прототип форми для результатів розрахунку параметрів системи освітлення приведений на рис. 2.15.

Результати розрахунку	
Коефіцієнт освітленості	
Площа світлових отворів	
Коефіцієнт відбиття	
Кількість світильників	
Сумарна потужність	
Сумарний потік	

Рисунок 2.15 – Прототип форми для результатів розрахунку параметрів системи освітлення

Загалом, виходячи з наведеного вище аналізу та результатів прототипіювання, проектування інтерфейсу та користувацького досвіду системи, можна стверджувати, що проектування інтерфейса стосується, насамперед, створення візуальної «мови» та візуалізації, яка об'єднує та втілює всі практики правильного та ефективного проектування системи [23]. Крім того, створюються відповідні умови для існування та гармонізації в рамках єдиної системи, яка об'єднує досвід користувача в усьому обсязі застосунка і не спричиняє невинуватених відхилень від його типової поведінки. Нарешті, надається та адаптується належним чином гнучка основа, яка просуває ім'я, інновації, які вона надає, а також вираження унікальності кожного застосунка [20].

Поряд зі своїми цілями, дизайн UI/UX, гармонізує та відповідає набору основних принципів. На першому плані — візуальна форма, тобто те, як реальне та матеріальне середовище передається на екрані. Отже, це відтворення, в першу чергу, відображає професійне середовище та його текстуру. У цей аспект входить те, як відбивається світло та створюються тіні на різних поверхнях [23]. Такого підходу потрібно дотримуватися під час проектування, щоб перенести професійне середовище використання у віртуальне середовище застосунка.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ ВІДНОВЛЕННЯ ОСВІТЛЕННЯ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

3.1. Розробка програмних модулів класів автоматизованої системи відновлення освітлення

.NET Framework є частиною операційної системи Microsoft Windows. Він був створений для спрощення програмування та задоволення багатьох типових потреб у цій сфері, але його застосування не обмежується лише цим. Він використовувався у веб-застосунках, підключенні до баз даних, інтернет-комунікації, створенні інтерфейсів та числових алгоритмах. Ця технологія дозволяє програмістам використовувати методи з будь-якого класу у поєднанні з власним кодом та створювати автономні додатки, такі як, наприклад, додатки Silverlight. Нижче наведено рівні .NET Framework [23].

Розглянемо рівні цієї технології. В структурі .NET Framework ми бачимо бібліотеки класів, де відбувається виконання коду, на якому рівні створюються служби. Також бачимо, що управління застосунками знаходиться у зовнішній частині циклу розробки, оскільки перед ним йдуть служби, а потім виконання .NET-середовища [22]. Цей простір він ділить з некерованими модулями. Решта керованих елементів містить бібліотеки об'єктів і класів, а також виконання, яке знаходиться глибше, на якому базуються бібліотеки. У центрі всього цього знаходиться операційна система та апаратне забезпечення, де знаходяться основні елементи для побудови решти команд керованих та некерованих модулів [23].

Для виконання завдань в кваліфікаційній роботі прийняте рішення використовувати мову програмування C#. C# — це мова програмування, що підтримує повною мірою об'єктно-орієнтоване програмування (ООП). ООП використовує об'єкти для створення програм і забезпечує ефективне

створення візуального інтерфейса [21]. Лістинг модуля класу ініціалізації головної форми застосунка приведений на рис. 3.1.

```
namespace LightApp
{
    partial class MainForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
        /// be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.menuStripMain = new System.Windows.Forms.MenuStrip();
            this.fileToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.connectDbToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.exitToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.referencesToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.EquipmentsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.agentsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.operationsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.ProjectsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.ParsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.payoutsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.reportsToolStripMenuItem = new
```

Рисунок 3.1 – Лістинг модуля ініціалізації головної форми застосунка

Будь-яка складна програмна система може містити сотні або навіть тисячі невидимих «рухомих частин». Кожна з цих рухомих частин несе власну відповідальність за розробку коду [20]. Інкапсуляція — це основний етап, на якому кожна частина має відповідальність і конкретні властивості щодо себе та знає якомога менше про інші частини коду. Це призводить до створення

більш стабільного середовища для програми, до більш зрозумілого коду, який забезпечує зміну хоча б в одній частині, щоб вплинути на решту системи [20].

Лістинг встановлення параметрів візуальних компонентів приведений на рис. 3.2.

```
//
// btnProjectDelete
//
this.btnProjectDelete.Location = new System.Drawing.Point  ➤
(988, 396);
this.btnProjectDelete.Name = "btnProjectDelete";
this.btnProjectDelete.Size = new System.Drawing.Size(138,  ➤
23);
this.btnProjectDelete.TabIndex = 19;
this.btnProjectDelete.Text = "Видалити";
this.btnProjectDelete.UseVisualStyleBackColor = true;
this.btnProjectDelete.Click += new System.EventHandler  ➤
(this.btnProjectDelete_Click);
//
// btnProjectEdit
//
this.btnProjectEdit.Location = new System.Drawing.Point  ➤
(988, 315);
this.btnProjectEdit.Name = "btnProjectEdit";
this.btnProjectEdit.Size = new System.Drawing.Size(138,  ➤
23);
this.btnProjectEdit.TabIndex = 18;
this.btnProjectEdit.Text = "Редагувати";
this.btnProjectEdit.UseVisualStyleBackColor = true;
this.btnProjectEdit.Click += new System.EventHandler  ➤
(this.btnProjectEdit_Click);
```

Рисунок 3.2 – Лістинг встановлення параметрів візуальних компонентів

В системі об'єкти отримують інструкції та деталі щодо того, як вони будуть працювати в програмі, від класів. Класи містять лише дві важливі складові: властивості та методи. Властивості можуть створюватися під час виконання програми або створюватися з самого початку для конкретної мети. Спадковість (Inheritance) — це найважливіша частина ООП. Один клас може бути нащадком іншого класу і створювати власних нащадків. В основному класи успадковують усі характеристики від інших класів [23].

Можливість обслуговування (Maintainability). Об'єктно-орієнтовані мови програмування забезпечують кращу можливість обслуговування програм. Цьому сприяє наявність об'єктів у кожній програмі та те, що вони використовують класи, які базуються на модулях [21]. Код набагато легше підтримувати, ніж код, який базується на програмуванні підпрограм. Код є підтримуваним, оскільки його можна розділити на методи — члени класів, властивості, обробники подій, оскільки сам програміст, створюючи власні

бібліотеки, може розділити код на такі структури, як, наприклад: загальні шаблони, класи, інтерфейси, члени класів та інші.

На лістингу нижче, рис. 3.3, приведений фрагмент програмного коду з ілюстрацією створення нових об'єктів [23].

```
//
// btnParEdit
//
this.btnParEdit.Location = new System.Drawing.Point(853, 244);
this.btnParEdit.Name = "btnParEdit";
this.btnParEdit.Size = new System.Drawing.Size(151, 23);
this.btnParEdit.TabIndex = 11;
this.btnParEdit.Text = "Редарувати";
this.btnParEdit.UseVisualStyleBackColor = true;
this.btnParEdit.Click += new System.EventHandler (this.btnParEdit_Click);
//
// dtpParDate
//
this.dtpParDate.Location = new System.Drawing.Point(833, 96);
this.dtpParDate.Name = "dtpParDate";
this.dtpParDate.Size = new System.Drawing.Size(188, 20);
this.dtpParDate.TabIndex = 10;
//
// numParAmount
//
this.numParAmount.Location = new System.Drawing.Point(833, 161);
this.numParAmount.Maximum = new decimal(new int[] { 1874919423, 2328306, 0, 0});
this.numParAmount.Name = "numParAmount";
this.numParAmount.Size = new System.Drawing.Size(188, 20);
this.numParAmount.TabIndex = 9;
```

Рисунок 3.3 – Програмний код з ілюстрацією створення нових об'єктів

Система автоматичного очищення пам'яті в С# базується на управлінні пам'яттю. Її робота визначається саме так, як вказано в назві (видаляє сміття з комп'ютера). Об'єкти займають пам'ять під час створення. Після створення об'єктів деякі з них більше не потрібні. Garbage collector автоматично їх виявляє, звільняє пам'ять від них і видаляє їх. У підсумку це забезпечує більш ефективне управління пам'яттю та полегшує її використання, оскільки зменшує помилки пам'яті. Крім того, С# дозволяє вставляти додаткові коментарі у вихідний код. Для цього перед кожним рядком вихідного коду потрібно поставити // [22]. Тоді С# сприймає вихідний код як XML і може експортувати цей код в окремий XML-файл. Це корисно, коли над одним додатком працює багато розробників, а текст коментарів повинен пояснити наступному розробнику, як використовувати вихідний код.

3.2. Розробка програмних модулів оперативного світлотехнічного розрахунку автоматизованої системи

Для розробки програмних модулів оперативного світлотехнічного розрахунку C# має багато потужних можливостей, оскільки базується на інструментах, які взаємодіють не тільки з нею середовищем розробки, а й з рештою системи. Компанія Microsoft, яка розробила C#, заснувала її на певній технології та побудувала на базі бібліотек .NET Framework. Кожного разу, коли оновлювався .NET Framework, оновлювалася й C#. Тому на сьогодні маємо потужний інструмент для розробки програмних модулів оперативного світлотехнічного розрахунку [17].

За основу для розрахунків візьмемо орієнтовний план приміщення з елементами освітлення, який показано на рис. 3.4.

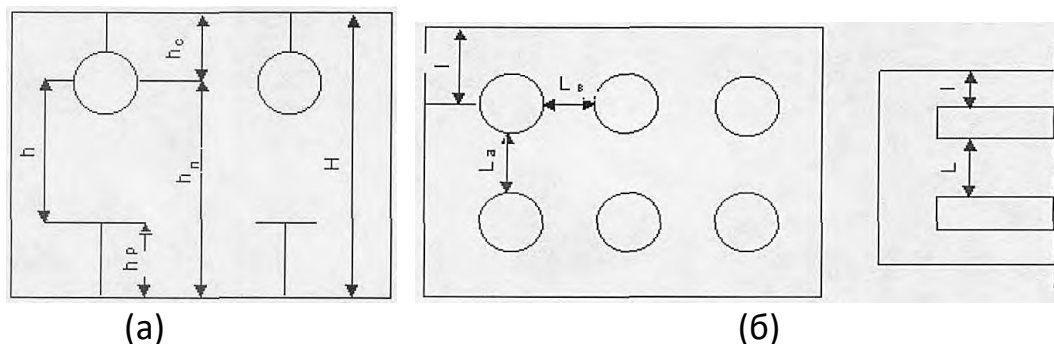


Рис. 3.4 – Орієнтовний план приміщення з елементами освітлення

На рисунку зображені позначки:

- H - висота;
- h_p - висота поверхні;
- h_c - висота звису;
- $h = H_{hc-hp}$ - розрахункова висота;
- $h_n = H_{hc}$ - висота підвісу;
- L_a і L_b - відстань між світильниками;
- l - відстань від крайніх світильників до стін.

Властивості світильників:

- ГСП 52-003;
- Г - лампа типу ДРІ;
- С- підвіс;
- П- для пром підприємств;
- 52- серія;
- 003 - модифікація;
- захист - IP50;
- кліматичне виконання - УХЛ1;
- К.К.Д. світильника-87%;
- крива сили світла - Г.

Відстань світильників над підлогою [16]:

$$h_n = H - h_c = 1,3м,$$

де $H = 2,3м$ – висота;

$h_c = 1м$ – відстань від перекриття.

Розрахункова висота [16]:

$$h = h_c - h_p = 1,6 м$$

Розрахуємо відстань між лампами:

$$\lambda = L/h, \quad L = \lambda * h,$$

де $\lambda = \lambda_c = 0,8$;

h – висота;

L – відстань між лампами.

$$L = 0.8 * 7.5 = 6,00 м$$

Кількість ламп в ряду [15]:

$$N_1 = \frac{A - L}{L} = \frac{60 - 6.73}{6} = 8,88$$

Приймемо кількість ламп в ряду 6 шт.

Кількість рядів:

$$N_2 = \frac{B - L}{L} = \frac{60 - 6.93}{6.93} = 7.65$$

Кількість рядів ламп – 8.

Відстань між лампами:

$$L = \frac{A - l * 2}{7} = \frac{60 - 6.93}{7} = 7.58 \text{ м.}$$

Відстань між лампами – 7.4 м. Оскільки А = В відстань між рядами 7,5м.

Відстані ламп від стіни:

$$l = \frac{A - L * 8}{2} = \frac{60 - 7.6 * 6}{2} = 7.2 \text{ м.}$$

Лампи розташовані в вершинах прямокутника 7,2м x 7,2 м.

При використанні ламп ДРЛ за формулою визначається потік однієї лампи, і підбирається стандартна лампа [16]. Потік лампи повинен відрізнятися від необхідного не більше ніж на + 20% або -10%.

Визначимо:

- площа - $S = 3600 \text{ м}^2$;
- висота - $h = 1,6 \text{ м}$;
- мінімальна освітленість $E = 200 \text{ лк}$.

Формула для розрахунку:

$$F_{\text{л}} = \frac{E \cdot k \cdot z \cdot S}{N \cdot \eta} ,$$

де $k = 1,5$ – коефіцієнт запасу;

$z = 1,15$ – коефіцієнт нерівномірності освітлення;

η - коефіцієнт використання світлового потоку.

Індекс приміщення [16]:

$$i = \frac{A * B}{h * (A + B)} = \frac{60 * 60}{7.8 * (60 + 60)} = 3.7$$

Значення коефіцієнтів відбиття стелі, підлоги, стін і робочої поверхні:

$$\rho_n = 60\% \quad \rho_{cm} = 55\%; \quad \rho_p = 12\% \quad \eta = 0,69.$$

Визначення світлового потоку лампи [17]:

$$\Phi = \frac{E \cdot S \cdot Z \cdot K_3}{n \cdot \eta} = 36837 \text{ Лм,}$$

де $S = 3600 \text{ м}^2$; $E = 300 \text{ лк}$ - освітленість; $Z = 1,15$ - коефіцієнт мінімальної освітленості; $K_3 = 1,5$ - коефіцієнт запасу; $n = 60 \text{ шт.}$ - кількість світильників; $\eta = 0,79$ - коефіцієнт використання світлового потоку; Найближча стандартна лампа ДРИ 400 має світловий потік 36000 лм :

$$\Delta\Phi = \frac{\Phi_{\text{п}} - \Phi_{\text{р}}}{\Phi_{\text{п}}} * 100 = -2,33\%$$

що на 2,25 % менше заданого значення, що допустимо.

Потік лампи [17]:

$$\Phi = \frac{1000 \cdot E \cdot K_3}{\mu \cdot \sum \varepsilon},$$

де $\sum \varepsilon$ – умовна освітленість в контрольній точці; μ - коефіцієнт, що враховує дію віддалених від розрахункової точки світильників і відбитого світлового потоку від стін, стелі, підлоги, обладнання, що падає на робочу поверхню в розрахунковій точці (приймають в межах $\mu = 1,1 \dots 1,2$); k_3 - коефіцієнт запасу; Нормована освітленість $E = 250 \text{ лк}$.

Знаходимо відстань від світильників 1,2,3,4 до точки А (рис. 3.5).

$$d = \sqrt{3.75^2 + 3.75^2} = 5.3 \text{ м.}$$

Знаходимо відстань від світильників 5,6 до точки А

$$d = \sqrt{((7,5 + 3.75)^2 + 3.75^2} = 11,86 \text{ м.}$$

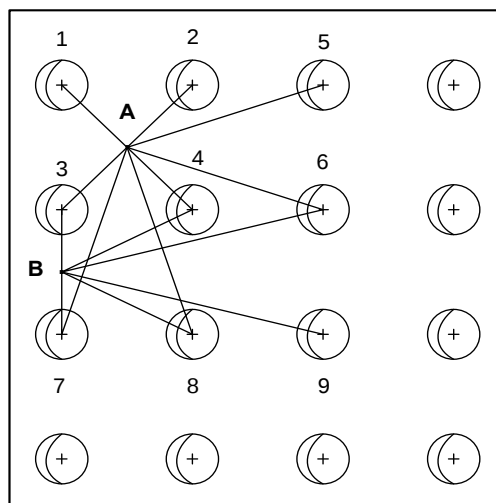


Рисунок 3.5 – Розрахункові відстані між лампами

Грунтуючись на наведених розрахунках, реалізовані програмні модулі. Форму інтерфейсу з результатами розрахунків показано на рис. 3.6.

Вхідні дані для розрахунку		Результати розрахунку
Мета проекту Виробничий цех	Коефіцієнт запасу 1,6	Коефіцієнт освітленості 9,6
Необхідна освітленість, лк 400	Тип джерела люмінесцентний	Площа світлових отворів 6,4
Площина нормування Стеля	Назва типу ПВЛМ	Коефіцієнт відбиття 244
Кут входу світлового потоку 60	Лампи розжарювання ППР	Кількість світильників 1,04
Середня нормована освітленість 250	Висота підвісу 2.4	Сумарна потужність 1600
Тип підвісної системи штучного освітлення сегменти		Сумарний потік 62000

Рисунок 3.6 – Форма інтерфейсу з результатами розрахунків

При програмної реалізації потрібно мати на увазі наступне.

У C# усі змінні передаються за значенням, якщо їх не вказано інакше. Проте слід бути обережними при розумінні та створенні цих типів. Оскільки параметри, що передаються за посиланням, зберігають посилання на об'єкт, саме це посилання буде скопійовано, а не сам об'єкт, що його містить. Тип змінної, навпаки, зберігає власне дані, тому в метод може бути передана точна копія. Цілочисельна змінна типу `int`, наприклад, передається в метод, і скільки б змін не відбулося під час виконання методу в змінній, при виході з методу ініціалізація змінної не змінюється [17]. Так само масив або будь-який тип посилання, такий як клас, передається в клас, і метод використовує оператор для зміни значення масиву, а нове значення передається в об'єкт масиву [17].

Проте C# вимагає, щоб змінні були ініційовані початковим значенням, перш ніж їх визначити як `ref` у методі. Якщо під час компіляції C# у коді зустрічається точка короткого замикання (тобто якщо компілятор бачить неініційовану змінну, яка мала б мати якесь значення), він її ініціює. Це досягається шляхом визначення змінних з неважливими даними [19].

3.3. Реалізація бази даних для програмного комплексу автоматизованої системи

Реалізація бази даних для програмного комплексу автоматизованої системи відновлення освітлення базується на загальних принципах побудови реляційних баз даних. База даних — це інструмент для збору та систематизації інформації. У базах даних можна зберігати інформацію про людей, товари, замовлення та інші об'єкти [18]. Таблиця, у якій кожен запис (рядок) містить інформацію про окремий об'єкт, а кожне поле (стовпець) — певний тип інформації, наприклад, ім'я, прізвище, адреса тощо. Поля повинні бути визначені як певний тип даних — текст, дата чи час, число або інший тип [19].

В кваліфікаційній роботі використовується база даних MySQL. MySQL надає набір інструментів для розробки та управління базами даних. Інсталятор MySQL забезпечує зручний процес інсталяції за допомогою майстра для всього програмного забезпечення MySQL у Windows. MySQL Workbench надає адміністраторам баз даних та розробникам інтегроване середовище інструментів для проектування баз даних, адміністрування, розробки SQL та міграції баз даних [19].

На рис. 3.7 приведена ілюстрація середовища для створення нової бази даних.

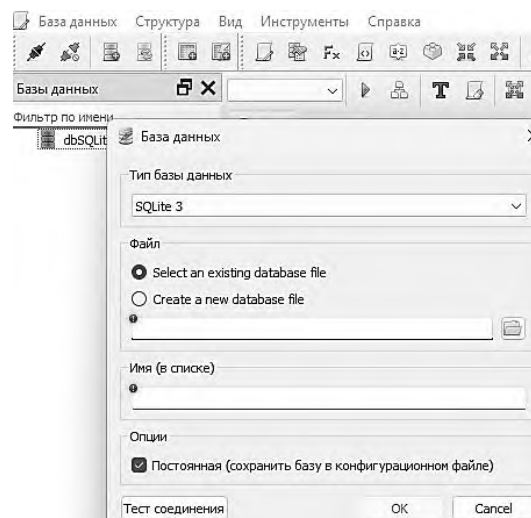


Рисунок 3.7 – Ілюстрація створення нової бази даних

Процес визначення назви база даних приведений на рис. 3.8.

The image shows a standard database creation dialog box. It has a title bar and a 'General' tab. Inside the dialog, there are three labeled fields: 'Database Name' with the text 'diploma', 'Character set' with a dropdown menu showing 'Default character set', and 'Collation' with an empty dropdown menu. At the bottom right, there are two buttons: 'OK' and 'Cancel'.

Рисунок 3.8 – Вибір імені бази даних та кодування

При заповненні бази даних потрібно вносити інформацію про тривалість кожного процесу проектування з метою ведення історії робіт. Також планується зберігати інформацію про продовження існуючих проєктів, тобто знати, чи є певний проєкт продовженням іншого, щоб можна було легко відстежувати послідовність проєктів того ж самого проєктувальника [18]. Потрібно включити інформацію про довідкові джерела таким чином, щоб кожен проєкт або вид роботи управлявся одним проєктувальником. Крім того, кожен проєкт бажано зберігати в окремому інформаційному блоці.

Ми розуміємо реляційну модель як підхід, що дозволяє розглядати дані як об'єкти реального світу, які відрізняються один від одного своїми основними характеристиками [19]. Кожен об'єкт можна описати за допомогою набору його характеристик (так званих атрибутів) і на основі цього відрізнити його від інших об'єктів. Ця модель підтримує відносини «один до багатьох», «один до одного» та «багато до багатьох», посилаючись на інформацію, яку ми зберігаємо, наприклад: система освітлення містить дані про один до багатьох систем освітлення; це еквівалентно відношенню «один» (лампа) до «багатьох» (серія ламп). Виходячи з цих принципів покажемо побудову таблиць в базі даних, як проілюстровано на рис. 3.9.

Приведемо типи даних, які були використані при побудові бази даних.

Чисельні [18]. INT (INTEGER) – 4 байти. SMALLINT – 2 байти. TINYINT – 1 байт. MEDIUMINT – 3 байти. BIGINT – 8 байтів. DECIMAL (NUMERIC) – Числа з плаваючою комою у вигляді рядків. FLOAT (m,d) – числа

з плаваючою комою, де «m» — кількість цифр у цілій частині, а «d» — кількість знаків після коми. DOUBLE (REAL) – число з плаваючою комою з подвійною точністю. BIT (BOOL, BOOLEAN) – ціле число зі значенням 0 або 1.

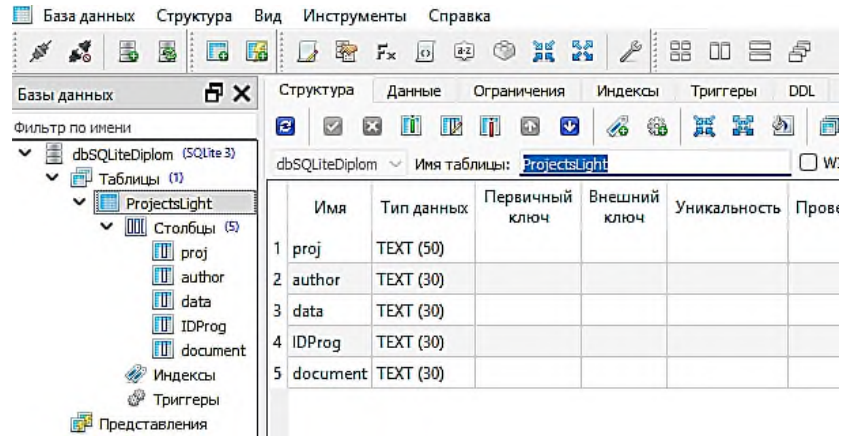


Рисунок 3.9 – Створення таблиці у базі даних

DATA [18]. DATE – дата з роком, місяцем і днем. DATETIME – дата (рік-місяць-день) і час (години-хвилини-секунди). TIME – час (години-хвилини-секунди). Формат збереження — «HH:MM:SS». TIMESTAMP – дата та час за UTC. YEAR – рік, що подається у вигляді 2 або 4 цифр, за замовчуванням – 4.

STRING [18]. CHAR – фіксована довжина. Від 1 до 255 символів. VARCHAR – змінна довжина. Від 1 до 255 символів. BLOB – максимальна довжина 65 535 символів. Підходить для бінарних об'єктів, таких як текстові файли, зображення, аудіо- та відеофайли. TINYBLOB – максимальна довжина 255 символів. MEDIUMBLOB – максимальна довжина 16 777 215 символів. LONGBLOB – максимальна довжина 4 294 967 298 символів.

STRING [18]. SET – зберігає 0, одне або кілька значень у списку, що містить максимум 64 можливі значення. ENUM – так само, як SET, але може зберігати лише одне значення. TEXT – максимальна довжина — 65 535 символів. Використовується для зберігання простого тексту без форматування. TINYTEXT – максимальна довжина 255 символів. MEDIUMTEXT – максимальна довжина 16 777 215 символів. LONGTEXT – максимальна довжина 4 294 967 298 символів.

На рис. 3.10 приведена форма з заповненням параметрів таблиць.

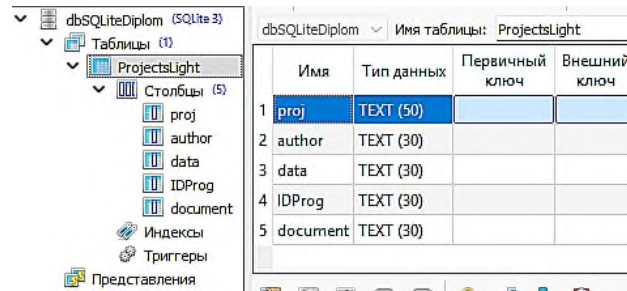


Рисунок 3.10 – Заповнення параметрів таблиць

Створення таблиць реалізується запитом `CREATE TABLE`, який створює нову таблицю в базі даних, що використовується. Таблиця містить стовпці, які є текстовими або числовими. Якщо спробувати зберегти в них значення, що перевищують їх межі, вони будуть обрізані для можливості їх зберігання [17]. З цієї причини важливо зарезервувати достатньо місця для кожного стовпця. Якщо передбачається, що багато записів займатимуть лише частину зарезервованого простору, можна використовувати тип `varchar`, подібний до `char`, з тією різницею, що значення займатиме менше місця, ніж зазначено, якщо рядок коротший за вказаний максимум, що дозволить заощадити простір для зберігання [18].

Приклад створеної таблиці приведений на рис. 3.11.

proj	author	data	IDProg	document
workshop	Zhurbitsky	25.02.26	00023	contract
corridor	Klimenko	14.03.26	00087	agreement
storage room	Tishchenko	18.04.26	00678	contract
living room	Zhurbitsky	27.04.26	00042	contract
factory workshop	Ivashchenko	02.05.26	01974	agreement

Рисунок 3.11 – Таблиця з даними бази даних ProjectsLight

Основна ідея інформаційних систем полягає в тому, щоб за допомогою різних інструментів відображати ситуації з реального світу. Тобто переносити реальність конкретної ситуації в базу даних. Одним із таких інструментів є модель даних, яка дозволяє абстрактне описати, у якому вигляді інформація, що міститься в базі, буде зберігатися [18].

3.4. Розробка форм інтерфейсу користувача автоматизованої системи відновлення освітлення в надзвичайних ситуаціях

Поряд зі своїми цілями, дизайн UI гармонує та відповідає набору основних принципів. На першому плані — візуальна форма, тобто те, як реальне та матеріальне середовище передається на екрані. Отже, ми розуміємо, що це відтворення надихається природним світом та його текстурою [20]. У цей аспект входить навіть те, як відбивається світло та створюються тіні на різних поверхнях. Такого підходу дотримуються і під час проектування, щоб перенести природне середовище використання у віртуальне середовище застосунка. Крім того, потрібно дотримуватися колірних рішень, що стали звичними в цих середовищах і стосуються переважно яскравих кольорів, таких як червоний, а також нейтральних, як-от білий. Крім того, ми використовуємо такі елементи, як шрифт, колір, пробіли, щоб створити відповідну ієрархію, яка обрамляє змістовний контент, з кінцевою метою зосередити увагу користувача на досвіді використання застосунка [20]. Додаток, який проєктується, це система визначення параметрів освітлення. У конкретному випадку роботи обране проєкт освітлення, хоча реалізація системи здійснена таким чином, щоб її можна було легко розширювати та модифікувати [19]. Отже, всі відповідні вимірювання та аналізи здійснюються відповідно до таких алгоритмів, що відповідають специфікаціям та правилам даного проєкту. Його метою є надання інструментів для отримання якості освітлення, що включають вдосконалення техніки проєктування, нові методи розрахунків, а також вдосконалення обладнання [19]. У нашому випадку користувач спочатку зможе створити обліковий запис або увійти безпосередньо в додаток, якщо він вже створював обліковий запис раніше. Там він зможе переглянути різні статистичні дані з попередніх проєктів, в яких брав участь. Постійна інтеграція — це філософія програмування та набір практик, які спонукають команди розробників вносити невеликі зміни та часто перевіряти свій код, розміщуючи його у репозиторіях.

Оскільки більшість сучасних додатків вимагають розробки коду на різних платформах та з використанням різних інструментів, команді потрібен механізм для інтеграції та валідації своїх змін [20]. Технічна мета інтеграції — запровадити послідовний та автоматизований спосіб побудови та тестування додатків. Завдяки послідовності в процесі інтеграції команди частіше вносять зміни до коду, що призводить до кращої співпраці та вищої якості програмного забезпечення [22]. На рис. 3.12 зображена головна форма застосунку.

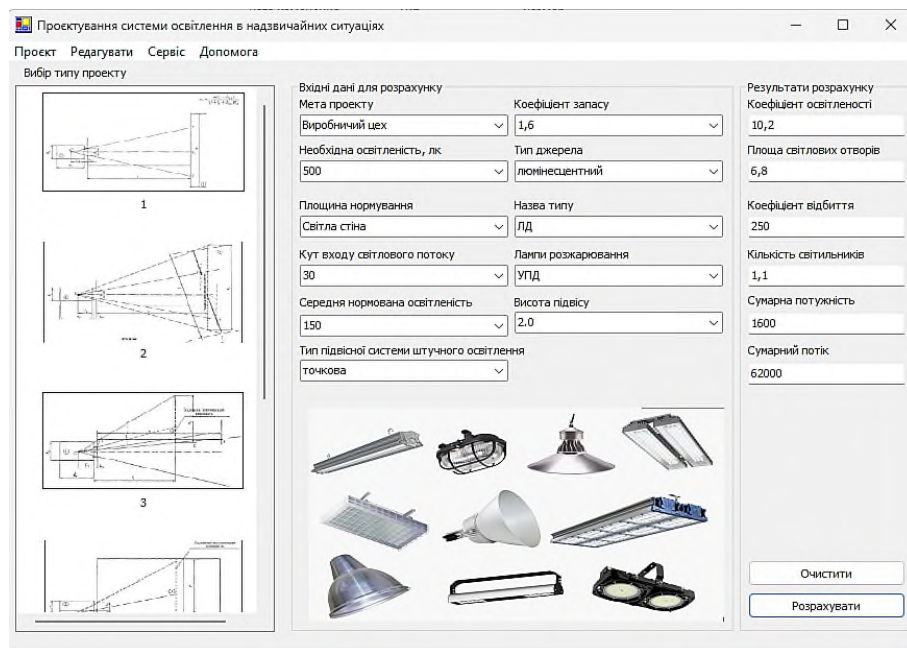


Рисунок 3.12 – Головна форма застосунку

Архітектура застосунка, як і будь-якого іншого застосунка та системи в будь-якій операційній системі, відповідає певним стандартам, кінцевою метою яких є створення потужних та якісних додатків [21]. Вважається, що додатки мають більш або менш складну структуру порівняно з настільними застосунками різних операційних систем. Типовий застосунок включає в себе багато компонентів, деякі з яких — це форми, фрагменти, служби тощо. Поняття зручності використання стосується простоти користування інтерфейсом, того, наскільки швидко користувач розуміє принцип його роботи та виконує необхідні дії, а також надання корисної послуги. Це дуже важливий критерій оцінки, оскільки інтерфейс з високим рівнем зручності використання краще задовольняє потреби користувача [23].

На рис. 3.13 приведена форма з результатами розрахунків.

The screenshot shows a vertical form titled "Результати розрахунку" (Calculation Results). It contains several input fields with numerical values: "Коефіцієнт освітленості" (1,8), "Площа світлових отворів" (14,6), "Коефіцієнт відбиття" (1,7), "Кількість світильників" (16), "Сумарна потужність" (1600), and "Сумарний потік" (62000). At the bottom, there are two buttons: "Очистити" (Clear) and "Розрахувати" (Calculate).

Результати розрахунку	
Коефіцієнт освітленості	1,8
Площа світлових отворів	14,6
Коефіцієнт відбиття	1,7
Кількість світильників	16
Сумарна потужність	1600
Сумарний потік	62000

Очистити

Розрахувати

Рисунок 3.13 – Форма з результатами розрахунків

Зовнішній вигляд пов'язаний з уявленням про красу та способами, якими ми можемо її сприймати. Естетика інтерфейсу має значний вплив на думку користувача про нього і, більше того, впливає на його сприйняття зручності використання, надійності та продуктивності системи. Деякі дослідження, зокрема, вказують, що перше враження, яке формує користувач, складається протягом перших 50 мілісекунд. і зумовлене виключно естетичними характеристиками інтерфейсу [20].

Мова C# забезпечує можливість проектувати інтерфейс застосунка як на етапі конструювання форм, так і в процесі виконання програми. На лістинге нижче, рис. 3.14, приведений програмний код, який характеризує таку можливість створення і управління інтерфейсом [21].

```

// dgvAgents
//
this.dgvAgents.ColumnHeadersHeightSizeMode =
    System.Windows.Forms.DataGridViewColumnHeadersHeightMode.AutoSize;
this.dgvAgents.Location = new System.Drawing.Point(22,
    22);
this.dgvAgents.Name = "dgvAgents";
this.dgvAgents.Size = new System.Drawing.Size(703, 429);
this.dgvAgents.TabIndex = 0;
this.dgvAgents.SelectionChanged += new System.EventHandler
    (this.dgvAgents_SelectionChanged);
//
// tabProjects
//
this.tabProjects.Controls.Add(this.btnProjectClear);
this.tabProjects.Controls.Add(this.btnProjectDelete);
this.tabProjects.Controls.Add(this.btnProjectEdit);
this.tabProjects.Controls.Add(this.numProjectTerm);
this.tabProjects.Controls.Add(this.dtpProjectDate);
this.tabProjects.Controls.Add(this.cmbProjectAgent);
this.tabProjects.Controls.Add(this.cmbProjectEquipment);
this.tabProjects.Controls.Add(this.btnProjectAdd);
this.tabProjects.Controls.Add(this.txtProjectSpec);
this.tabProjects.Controls.Add(this.txtProjectNumber);
this.tabProjects.Controls.Add(this.lblProjectAgent);
this.tabProjects.Controls.Add(this.lblProjectEquipment);
this.tabProjects.Controls.Add(this.lblProjectTerm);
this.tabProjects.Controls.Add(this.lblProjectSpec);
this.tabProjects.Controls.Add(this.lblProjectDate);
this.tabProjects.Controls.Add(this.lblProjectNumber);
this.tabProjects.Controls.Add(this.dgvProjects);
this.tabProjects.Location = new System.Drawing.Point(4,
    22);
this.tabProjects.Name = "tabProjects";
this.tabProjects.Padding = new
    System.Windows.Forms.Padding(3);
this.tabProjects.Size = new System.Drawing.Size(1288,
    669);
this.tabProjects.TabIndex = 2;
this.tabProjects.Text = "Договори";
this.tabProjects.UseVisualStyleBackColor = true;

```

Рисунок 3.14 – Програмний код, який характеризує можливість створення і управління інтерфейсом

ВИСНОВКИ

Була виконана випускна кваліфікаційна робота: "Комп'ютерна підтримка оперативного відновлення системи освітлення в надзвичайних ситуаціях». Робота присвячена актуальній темі – розробці системи автоматизації проєктування нових і відновлення існуючих систем освітлення. Актуальність теми кваліфікаційної роботи полягає у необхідності вирішення проблеми відновлення промислового і побутового освітлення в надзвичайних ситуаціях.

Під час виконання кваліфікаційної роботи було розроблено програму для автоматизації процесу відновлення промислового і побутового освітлення в надзвичайних ситуаціях.

Були отримані наступні результати:

- проведено дослідження математичних моделей розрахунку параметрів системи освітлення в надзвичайних ситуаціях;
- розроблені інформаційні структури автоматизованої системи оперативного відновлення системи освітлення;
- розроблені формально-логічні функціональні структури програмного комплексу автоматизації;
- розроблені компоненти бази даних для роботи автоматизованої системи оперативного відновлення освітлення;
- розроблені базові алгоритми програмного застосунку системи автоматизації відновлення освітлення;
- сконструйовані елементи візуального інтерфейсу програмного комплексу системи автоматизації відновлення освітлення;
- реалізована програмна система для автоматизації відновлення освітлення в надзвичайних ситуаціях; розробка програмних модулів класів автоматизованої системи відновлення освітлення;
- розроблені програмні модулі оперативного світлотехнічного розрахунку автоматизованої системи;

– реалізовані бази даних для програмного комплексу автоматизованої системи; розробка форм інтерфейсу користувача автоматизованої системи відновлення освітлення в надзвичайних ситуаціях.

Розроблена програмна система дає змогу підвищити ефективність процесу проектування нового та відновлення існуючого комплексу промислового і побутового освітлення. Система забезпечує можливості створювати и керувати проектами освітлення різних типів.

На основі отриманих результатів можна зробити висновок, що виконана робота характеризує актуальність теми, демонструє ефективність розроблених програмних модулів.

Результати роботи можуть бути використані для промисловості, установ, організацій для відновлення освітлення після аварійних відключень, а також для проектування систем освітлення з новими якостями в залежності від практичних вимог.

ПЕРЕЛІК ПОСИЛАНЬ

1. ДБН В.2.5-28:2018. *Природне і штучне освітлення* (із змінами № 1, внесеними у 2025 р., діє з 01.01.2026р.). - Київ: Мінрегіон України, 2025. - 114с.
2. ДСТУ EN 1838:2019. *Світлотехніка. Освітлення аварійне* (EN 1838:2013, IDT). — Київ: ДП «УкрНДНЦ», 2020. — 32 с.
3. ДСТУ EN 12464-1:2022. *Світло і освітлення. Освітлення робочих місць. Частина 1. Внутрішні робочі місця* (EN 12464-1:2021, IDT). — Київ: ДП «УкрНДНЦ», 2023.
4. ДСТУ EN 12464-2:2022. *Світло і освітлення. Освітлення робочих місць. Частина 2. Робочі місця просто неба* (EN 12464-2:2014, IDT). — Київ: ДП «УкрНДНЦ», 2023.
5. ДСТУ EN 13201-2:2019. *Освітлення доріг. Частина 2. Вимоги до експлуатаційних характеристик* (EN 13201-2:2015, IDT). — Київ: ДП «УкрНДНЦ», 2020.
6. Lopez-Navarro J., Industrial-District Lighting Group. Optimization of street lighting grid resilience and failure physics modeling under high thermal stress. *IEEE Transactions on Power Delivery*. 2025. Vol. 40, No. 2. P. 884–893.
7. Назаренко Л. О. *Джерела світла та світлотехнічні установки: підручник*. — Харків: ХНУМГ ім. О. М. Бекетова, 2021. — 342 с.
8. Король О. О., Самойленко М. В. *Енергоефективне електроосвітлення та світловий дизайн міського середовища: навчальний посібник*. — Київ: Каравела, 2022. — 216 с.
9. Черкашина О. Л., Литвиненко А. С. *Проектування та моделювання світлотехнічних систем в DIALux EVO: навч. посібник*. — Харків: ХНУМГ, 2023. — 158 с.
10. Tregenza P., Loe D. *The Design of Lighting. 3rd Edition*. — London: Routledge, 2022. — 314 p.
11. Winchip S. M. *Fundamentals of Lighting. 4th Edition*. — New York: Fairchild Books, 2023. — 400 p.

12. Пінчук Н. В., Жиделєв О. В. *Електрична частина та освітлення промислових підприємств: навч. посібник*. — Кривий Ріг: КНУ, 2022. — 194 с.
13. Мартяш О. В., Федорчук В. М. *Автоматизація керування системами штучного освітлення будівель: монографія*. — Львів: Видавництво Львівської політехніки, 2024. — 204 с.
14. Marques G., Pitarma R. Designing IoT-driven indoor air quality and smart lighting systems for healthcare environments. *Journal of Ambient Intelligence and Humanized Computing*. 2022. Vol. 13. P. 4511–4525.
15. Сорокін В. М., Кожухар О. В. Енергетична ефективність та надійність світлодіодних систем загального освітлення. *Технічна електродинаміка*. 2021. № 4. С. 67–73.
16. Баранов П. Л., Ткаченко Ю. А. Моделювання інтелектуальних систем внутрішнього освітлення на базі концепції Human Centric Lighting. *Світлотехніка та електроенергетика*. 2023. № 2 (34). С. 12–19.
17. EN 12464-1:2021. *Light and lighting — Lighting of work places — Part 1: Indoor work places*. — Brussels: European Committee for Standardization (CEN), 2021.
18. Поліщук С. Ю., Андрійчук В. А. Особливості проектування мереж зовнішнього штучного освітлення з використанням автономних сонячних електростанцій. *Енергетика і автоматика*. 2022. Вип. 5. С. 45–56.
19. Жуков К. А. Розрахунок світлового забруднення при проектуванні вечірнього архітектурного освітлення міських об'єктів. *Сучасні технології в будівництві*. 2024. № 1. С. 89–95.
20. Муха М. О., Савченко О. В. Порівняльний аналіз інженерних методів розрахунку освітленості при використанні точкових LED-джерел. *Вісник Вінницького політехнічного інституту*. 2025. № 3. С. 112–118.
21. Mishra S., Khare R. *Smart Lighting Systems: Design, Control and Energy Efficiency*. — Cham: Springer, 2024. — 245 p.

22. CIE 250:2022. *Action Plan for the Spectrally Non-Linear Characteristics of Photodetectors Used in LED Lighting Measurements*. — Vienna: International Commission on Illumination, 2022. — 48 p.

23. Zhu Y., Lightner C., & Mahdavi A. Implementation algorithms for Human Centric Lighting (HCL) in automated building management systems. *Energy and Buildings*. 2023. Vol. 282. Art. 112790.

ДОДАТОК А

Основні лістинги програмних модулів

```
namespace LightApp
{
    partial class MainForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
        /// be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.menuStripMain = new System.Windows.Forms.MenuStrip();
            this.fileToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.connectDbToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.exitToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.referencesToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.EquipmentsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.agentsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.operationsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.ProjectsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.ParsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.payoutsToolStripMenuItem = new
                System.Windows.Forms.ToolStripItem();
            this.reportsToolStripMenuItem = new
```

```

        System.Windows.Forms.ToolStripItem();
this.generateReportToolStripMenuItem = new
    System.Windows.Forms.ToolStripItem();
this.tabControlMain = new System.Windows.Forms.TabControl
    ();
this.tabEquipments = new System.Windows.Forms.TabPage();
this.btnEquipmentClear = new System.Windows.Forms.Button
    ();
this.btnEquipmentDelete = new System.Windows.Forms.Button
    ();
this.btnEquipmentEdit = new System.Windows.Forms.Button();
this.btnEquipmentAdd = new System.Windows.Forms.Button();
this.txtEquipmentAddress = new
    System.Windows.Forms.TextBox();
this.txtEquipmentPassport = new
    System.Windows.Forms.TextBox();
this.txtEquipmentName = new System.Windows.Forms.TextBox
    ();
this.lblEquipmentAddress = new System.Windows.Forms.Label
    ();
this.lblEquipmentPassport = new System.Windows.Forms.Label
    ();
this.lblEquipmentName = new System.Windows.Forms.Label();
this.dgvEquipments = new System.Windows.Forms.DataGridView
    ();
this.tabAgents = new System.Windows.Forms.TabPage();
this.btnAgentClear = new System.Windows.Forms.Button();
this.btnAgentDelete = new System.Windows.Forms.Button();
this.btnAgentEdit = new System.Windows.Forms.Button();
this.btnAgentAdd = new System.Windows.Forms.Button();
this.txtAgentSpec = new System.Windows.Forms.TextBox();
this.txtAgentAddress = new System.Windows.Forms.TextBox();
this.txtAgentPassport = new System.Windows.Forms.TextBox
    ();
this.txtAgentName = new System.Windows.Forms.TextBox();
this.lblAgentSpec = new System.Windows.Forms.Label();
this.lblAgentAddress = new System.Windows.Forms.Label();
this.lblAgentPassport = new System.Windows.Forms.Label();
this.lblAgentName = new System.Windows.Forms.Label();
this.dgvAgents = new System.Windows.Forms.DataGridView();
this.tabProjects = new System.Windows.Forms.TabPage();
this.btnProjectClear = new System.Windows.Forms.Button();
this.btnProjectDelete = new System.Windows.Forms.Button();
this.btnProjectEdit = new System.Windows.Forms.Button();
this.numProjectTerm = new
    System.Windows.Forms.NumericUpDown();
this.dtpProjectDate = new
    System.Windows.Forms.DateTimePicker();
this.cmbProjectAgent = new System.Windows.Forms.ComboBox
    ();
this.cmbProjectEquipment = new
    System.Windows.Forms.ComboBox();
this.btnProjectAdd = new System.Windows.Forms.Button();

```

```

this.txtProjectSpec = new System.Windows.Forms.TextBox();
this.txtProjectNumber = new System.Windows.Forms.TextBox
    ();
this.lblProjectAgent = new System.Windows.Forms.Label();
this.lblProjectEquipment = new System.Windows.Forms.Label
    ();
this.lblProjectTerm = new System.Windows.Forms.Label();
this.lblProjectSpec = new System.Windows.Forms.Label();
this.lblProjectDate = new System.Windows.Forms.Label();
this.lblProjectNumber = new System.Windows.Forms.Label();
this.dgvProjects = new System.Windows.Forms.DataGridView
    ();
this.tabPars = new System.Windows.Forms.TabPage();
this.btnParClear = new System.Windows.Forms.Button();
this.btnParDelete = new System.Windows.Forms.Button();
this.btnParEdit = new System.Windows.Forms.Button();
this.dtpParDate = new System.Windows.Forms.DateTimePicker
    ();
this.numParAmount = new System.Windows.Forms.NumericUpDown
    ();
this.cmbParProject = new System.Windows.Forms.ComboBox();
this.btnParAdd = new System.Windows.Forms.Button();
this.lblParAmount = new System.Windows.Forms.Label();
this.lblParDate = new System.Windows.Forms.Label();
this.lblParProject = new System.Windows.Forms.Label();
this.dgvPars = new System.Windows.Forms.DataGridView();
this.tabPayouts = new System.Windows.Forms.TabPage();
this.btnPayoutDelete = new System.Windows.Forms.Button();
this.btnPayoutClear = new System.Windows.Forms.Button();
this.btnPayoutEdit = new System.Windows.Forms.Button();
this.dtpPayoutDate = new
    System.Windows.Forms.DateTimePicker();
this.numPayoutAmount = new
    System.Windows.Forms.NumericUpDown();
this.cmbPayoutProject = new System.Windows.Forms.ComboBox
    ();
this.btnPayoutAdd = new System.Windows.Forms.Button();
this.lblPayoutAmount = new System.Windows.Forms.Label();
this.lblPayoutDate = new System.Windows.Forms.Label();
this.lblPayoutProject = new System.Windows.Forms.Label();
this.dgvPayouts = new System.Windows.Forms.DataGridView();
this.menuStripMain.SuspendLayout();
this.tabControlMain.SuspendLayout();
this.tabEquipments.SuspendLayout();
((System.ComponentModel.ISupportInitialize)
    (this.dgvEquipments)).BeginInit();
this.tabAgents.SuspendLayout();
((System.ComponentModel.ISupportInitialize)
    (this.dgvAgents)).BeginInit();
this.tabProjects.SuspendLayout();
((System.ComponentModel.ISupportInitialize)
    (this.numProjectTerm)).BeginInit();
((System.ComponentModel.ISupportInitialize)

```

```

        (this.dgvProjects)).BeginInit();
this.tabPars.SuspendLayout();
((System.ComponentModel.ISupportInitialize).BeginInit()
    (this.numParAmount)).BeginInit();
((System.ComponentModel.ISupportInitialize).BeginInit()
    (this.dgvPars)).BeginInit();
this.tabPayouts.SuspendLayout();
((System.ComponentModel.ISupportInitialize).BeginInit()
    (this.numPayoutAmount)).BeginInit();
((System.ComponentModel.ISupportInitialize).BeginInit()
    (this.dgvPayouts)).BeginInit();
this.SuspendLayout();
//
// menuStripMain
//
this.menuStripMain.Items.AddRange(new
    System.Windows.Forms.ToolStripItem[] {
this.fileToolStripMenuItem,
this.referencesToolStripMenuItem,
this.operationsToolStripMenuItem,
this.reportsToolStripMenuItem});
this.menuStripMain.Location = new System.Drawing.Point(0,
    0);
this.menuStripMain.Name = "menuStripMain";
this.menuStripMain.Size = new System.Drawing.Size(1320,
    24);
this.menuStripMain.TabIndex = 0;
this.menuStripMain.Text = "menuStripMain";
//
// fileToolStripMenuItem
//
this.fileToolStripMenuItem.DropDownItems.AddRange(new
    System.Windows.Forms.ToolStripItem[] {
this.connectDbToolStripMenuItem,
this.exitToolStripMenuItem});
this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
this.fileToolStripMenuItem.Size = new System.Drawing.Size
    (48, 20);
this.fileToolStripMenuItem.Text = "Файл";
//
// connectDbToolStripMenuItem
//
this.connectDbToolStripMenuItem.Name =
    "connectDbToolStripMenuItem";
this.connectDbToolStripMenuItem.Size = new
    System.Drawing.Size(205, 22);
this.connectDbToolStripMenuItem.Text = "Підключити Базу
    Даних";
this.connectDbToolStripMenuItem.Click += new
    System.EventHandler
    (this.connectDbToolStripMenuItem_Click);
//
// exitToolStripMenuItem

```

```

this.cmbProjectAgent.Size = new System.Drawing.Size(167, 21);
this.cmbProjectAgent.TabIndex = 15;
//
// cmbProjectEquipment
//
this.cmbProjectEquipment.FormattingEnabled = true;
this.cmbProjectEquipment.Location = new System.Drawing.Point(972, 185);
this.cmbProjectEquipment.Name = "cmbProjectEquipment";
this.cmbProjectEquipment.Size = new System.Drawing.Size(167, 21);
this.cmbProjectEquipment.TabIndex = 14;
//
// btnProjectAdd
//
this.btnProjectAdd.Location = new System.Drawing.Point(988, 275);
this.btnProjectAdd.Name = "btnProjectAdd";
this.btnProjectAdd.Size = new System.Drawing.Size(138, 23);
this.btnProjectAdd.TabIndex = 13;
this.btnProjectAdd.Text = "Додати";
this.btnProjectAdd.UseVisualStyleBackColor = true;
this.btnProjectAdd.Click += new System.EventHandler(this.btnProjectAdd_Click);
//
// txtProjectSpec
//
this.txtProjectSpec.Location = new System.Drawing.Point(972, 109);
this.txtProjectSpec.Name = "txtProjectSpec";
this.txtProjectSpec.Size = new System.Drawing.Size(167, 20);
this.txtProjectSpec.TabIndex = 9;
//
// txtProjectNumber
//
this.txtProjectNumber.Location = new System.Drawing.Point(972, 25);
this.txtProjectNumber.Name = "txtProjectNumber";
this.txtProjectNumber.Size = new System.Drawing.Size(167, 20);
this.txtProjectNumber.TabIndex = 7;
//
// lblProjectAgent
//
this.lblProjectAgent.AutoSize = true;
this.lblProjectAgent.Location = new System.Drawing.Point(831, 226);
this.lblProjectAgent.Name = "lblProjectAgent";
this.lblProjectAgent.Size = new System.Drawing.Size(36, 13);

```

0

```

this.lblProjectAgent.TabIndex = 6;
this.lblProjectAgent.Text = "Агент";
//
// lblProjectEquipment
//
this.lblProjectEquipment.AutoSize = true;
this.lblProjectEquipment.Location = new System.Drawing.Point(831, 185);
this.lblProjectEquipment.Name = "lblProjectEquipment";
this.lblProjectEquipment.Size = new System.Drawing.Size(39, 13);
this.lblProjectEquipment.TabIndex = 5;
this.lblProjectEquipment.Text = "Клієнт";
//
// lblProjectTerm
//
this.lblProjectTerm.AutoSize = true;
this.lblProjectTerm.Location = new System.Drawing.Point(831, 147);
this.lblProjectTerm.Name = "lblProjectTerm";
this.lblProjectTerm.Size = new System.Drawing.Size(90, 13);
this.lblProjectTerm.TabIndex = 4;
this.lblProjectTerm.Text = "Строк, у місяцях";
//
// lblProjectSpec
//
this.lblProjectSpec.AutoSize = true;
this.lblProjectSpec.Location = new System.Drawing.Point(831, 109);
this.lblProjectSpec.Name = "lblProjectSpec";
this.lblProjectSpec.Size = new System.Drawing.Size(74, 13);
this.lblProjectSpec.TabIndex = 3;
this.lblProjectSpec.Text = "Спеціалізація";
//
// lblProjectDate
//
this.lblProjectDate.AutoSize = true;
this.lblProjectDate.Location = new System.Drawing.Point(831, 67);
this.lblProjectDate.Name = "lblProjectDate";
this.lblProjectDate.Size = new System.Drawing.Size(33, 13);
this.lblProjectDate.TabIndex = 2;
this.lblProjectDate.Text = "Дата";
//
// lblProjectNumber
//
this.lblProjectNumber.AutoSize = true;
this.lblProjectNumber.Location = new System.Drawing.Point(831, 28);
this.lblProjectNumber.Name = "lblProjectNumber";

```

```

this.lblProjectNumber.Size = new System.Drawing.Size(41, 13);
this.lblProjectNumber.TabIndex = 1;
this.lblProjectNumber.Text = "Home";
//
// dgvProjects
//
this.dgvProjects.ColumnHeadersHeightSizeMode =
    System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dgvProjects.Location = new System.Drawing.Point(18, 17);
this.dgvProjects.Name = "dgvProjects";
this.dgvProjects.Size = new System.Drawing.Size(773, 491);
this.dgvProjects.TabIndex = 0;
this.dgvProjects.SelectionChanged += new
    System.EventHandler(this.dgvProjects_SelectionChanged);
//
// tabPars
//
this.tabPars.Controls.Add(this.btnParClear);
this.tabPars.Controls.Add(this.btnParDelete);
this.tabPars.Controls.Add(this.btnParEdit);
this.tabPars.Controls.Add(this.dtpParDate);
this.tabPars.Controls.Add(this.numParAmount);
this.tabPars.Controls.Add(this.cmbParProject);
this.tabPars.Controls.Add(this.btnParAdd);
this.tabPars.Controls.Add(this.lblParAmount);
this.tabPars.Controls.Add(this.lblParDate);
this.tabPars.Controls.Add(this.lblParProject);
this.tabPars.Controls.Add(this.dgvPars);
this.tabPars.Location = new System.Drawing.Point(4, 22);
this.tabPars.Name = "tabPars";
this.tabPars.Padding = new System.Windows.Forms.Padding
    (3);
this.tabPars.Size = new System.Drawing.Size(1288, 669);
this.tabPars.TabIndex = 3;
this.tabPars.Text = "Внески";
this.tabPars.UseVisualStyleBackColor = true;
//
// btnParClear
//
this.btnParClear.Location = new System.Drawing.Point(853, 287);
this.btnParClear.Name = "btnParClear";
this.btnParClear.Size = new System.Drawing.Size(151, 23);
this.btnParClear.TabIndex = 13;
this.btnParClear.Text = "ОЧИСТИТИ";
this.btnParClear.UseVisualStyleBackColor = true;
this.btnParClear.Click += new System.EventHandler
    (this.btnParClear_Click);
//
// btnParDelete

```

```

// dgvAgents
//
this.dgvAgents.ColumnHeadersHeightSizeMode =
    System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dgvAgents.Location = new System.Drawing.Point(22,
    22);
this.dgvAgents.Name = "dgvAgents";
this.dgvAgents.Size = new System.Drawing.Size(703, 429);
this.dgvAgents.TabIndex = 0;
this.dgvAgents.SelectionChanged += new System.EventHandler
    (this.dgvAgents_SelectionChanged);
//
// tabProjects
//
this.tabProjects.Controls.Add(this.btnProjectClear);
this.tabProjects.Controls.Add(this.btnProjectDelete);
this.tabProjects.Controls.Add(this.btnProjectEdit);
this.tabProjects.Controls.Add(this.numProjectTerm);
this.tabProjects.Controls.Add(this.dtpProjectDate);
this.tabProjects.Controls.Add(this.cmbProjectAgent);
this.tabProjects.Controls.Add(this.cmbProjectEquipment);
this.tabProjects.Controls.Add(this.btnProjectAdd);
this.tabProjects.Controls.Add(this.txtProjectSpec);
this.tabProjects.Controls.Add(this.txtProjectNumber);
this.tabProjects.Controls.Add(this.lblProjectAgent);
this.tabProjects.Controls.Add(this.lblProjectEquipment);
this.tabProjects.Controls.Add(this.lblProjectTerm);
this.tabProjects.Controls.Add(this.lblProjectSpec);
this.tabProjects.Controls.Add(this.lblProjectDate);
this.tabProjects.Controls.Add(this.lblProjectNumber);
this.tabProjects.Controls.Add(this.dgvProjects);
this.tabProjects.Location = new System.Drawing.Point(4,
    22);
this.tabProjects.Name = "tabProjects";
this.tabProjects.Padding = new
    System.Windows.Forms.Padding(3);
this.tabProjects.Size = new System.Drawing.Size(1288,
    669);
this.tabProjects.TabIndex = 2;
this.tabProjects.Text = "Договори";
this.tabProjects.UseVisualStyleBackColor = true;

```

```

        System.Windows.Forms.AutoScaleMode.Font;
        this.EquipmentSize = new System.Drawing.Size(1320, 753);
        this.Controls.Add(this.tabControlMain);
        this.Controls.Add(this.menuStripMain);
        this.MainMenuStrip = this.menuStripMain;
        this.Name = "MainForm";
        this.Text = "MainForm";
        this.menuStripMain.ResumeLayout(false);
        this.menuStripMain.PerformLayout();
        this.tabControlMain.ResumeLayout(false);
        this.tabEquipments.ResumeLayout(false);
        this.tabEquipments.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.dgvEquipments)).EndInit();
        this.tabAgents.ResumeLayout(false);
        this.tabAgents.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.dgvAgents)).EndInit();
        this.tabProjects.ResumeLayout(false);
        this.tabProjects.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.numProjectTerm)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.dgvProjects)).EndInit();
        this.tabPars.ResumeLayout(false);
        this.tabPars.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.numParAmount)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.dgvPars)).EndInit();
        this.tabPayouts.ResumeLayout(false);
        this.tabPayouts.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.numPayoutAmount)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.dgvPayouts)).EndInit();
        this.ResumeLayout(false);
        this.PerformLayout();
    }

```