

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет
Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:
Завідувач кафедри

_____ Наталія АУШЕВА

« ___ » _____ 2022 р.

Дипломна робота

на здобуття ступеня бакалавра
спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерний моніторинг та геометричне
моделювання процесів і систем»

на тему: «Моделювання гауссового шлейфу повітряної дисперсії»

Виконав (-ла):

студент (-ка) IV курсу, групи ТМ-81
Денисенко Яна Ігорівна _____

Керівник:

Професор, д.т.н.
Сліпченко Володимир Георгійович _____

Рецензент:

Ст. н. с. Інституту пролем реєстрації
Інформації НАН України, к.т.н.
Сенченко Вячеслав Родіонович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент (-ка) _____

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший

спеціальність 122 «Комп’ютерні науки»

освітня програма «Комп’ютерний моніторинг та геометричне моделювання процесів і систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Наталія АУШЕВА

(підпис)

” ___ ” _____ 2022р.

ЗАВДАННЯ

на дипломну роботу студенту

Денисенко Яни Ігорівни

(прізвище, ім’я, по батькові)

1. Тема роботи Моделювання гауссового шлейфу повітряної дисперсії

керівник роботи професор, д.т.н. Сліпченко Володимир Георгійович

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” ___ ” _____ 2022р.

№ _____

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи мова програмування C#, середовище розробки Visual Studio

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати аналогічні системи, розробити алгоритм побудови програмного забезпечення, обрати засоби реалізації, розробити інтерфес, базу даних, основні модулі для роботи з користувачем, протестувати систему.

5. Перелік ілюстративного матеріал

Структура програмного продукту, концептуальна модель бази даних, діаграма прецедентів, діаграма діяльності, приклади роботи системи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання "10" вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи		
2.	Вивчення та аналіз задачі	02-08.04	
3.	Розробка архітектури та загальної структури системи	09-15.04.22	
4.	Розробка структур окремих підсистем	16-22.04.22	
5.	Програмна реалізація системи	19-28.04.22	
6.	Оформлення пояснювальної записки	16-28.05.22	
7.	Захист програмного продукту	29.05.22	
8.	Передзахист	07.06.22	
9.	Захист		

Студент _____
(підпис)

Денисенко Я.І.
(прізвище та ініціали,)

Керівник роботи _____
(підпис)

Сліпченко В.Г.
(прізвище та ініціали,)

АНОТАЦІЯ

Обсяг дипломної роботи складає 88 сторінок, робота містить в собі 29 рисунків, 7 таблиць, 3 додатки та 20 посилань.

Метою даної роботи є розробка додатку для моделювання поширення забруднюючих речовин в атмосферному повітрі від різноманітних підприємств.

Об'єктом дослідження є моделювання поширення забруднюючих речовин у атмосферному повітрі.

У звіті наведено результати аналізу предметної галузі, проектування програмної реалізації та практичного застосування додатку для моделювання поширення забруднюючих речовин у атмосферному повітрі із відображенням результатів в вигляді звіту.

Ключові слова: C#, WINDOWS FORMS, МОДЕЛЮВАННЯ, ПОВІТРЯНА ДИСПЕРСІЯ.

ABSTRACT

The volume of the graduate work is 88 pages, the work contains 29 figures, 7 tables, 3 appendices and 20 references.

The purpose of this work is to develop an application for modeling the spread of pollutants in the air from various enterprises.

The object of research is to model the distribution of pollutants in the air.

The report presents the results of the analysis of the subject area, design of software implementation and practical application of the application for modeling the spread of pollutants in the air with the reflection of the results in the form of a report.

Keywords: C #, WINDOWS FORMS, SIMULATION, AIR DISPERSION.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕМІНІВ.....	7
ВСТУП.....	8
1. ЗАДАЧА МОДЕЛЮВАННЯ ГАУССОВОГО ШЛЕЙФУ ПОВІТРЯНОЇ ДИСПЕРСІЇ.....	9
1.1 Постановка задачі.....	9
1.2 Висновки до розділу.....	10
2. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ ЗАДАЧІ МОДЕЛЮВАННЯ ШЛЕЙФУ ПОВІТРЯНОЇ ДИСПЕРСІЇ.....	11
2.1 Опис програмних рішень.....	11
2.2 Висновки до розділу.....	16
3. МАТЕМАТИЧНА МОДЕЛЬ.....	17
3.1 Загальні характеристики Гауссового або нормального розподілу.....	17
3.2 Загальні характеристики Гауссового або нормального розподілу.....	18
3.3 Основне рівняння розсіювання домішок у Гауссовій теорії.....	19
3.4 Окремі випадки розрахунку приземних концентрацій.....	20
3.5 Практичне використання Гауссової теорії.....	21
3.6 Висновки до розділу.....	22
4. ЗАСОБИ РЕАЛІЗАЦІЇ.....	23
4.1 Мова програмування C#.....	23
4.2 Середовище розробки Visual Studio.....	24
4.3 Програмна технологія .NET Framework.....	25
4.4 Інтерфейс програмування Windows Forms.....	26

4.5	Технологія COM Interop.....	26
4.6	Мова структурованих запитів SQL.....	27
4.7	Система керування базами даних MySQL	27
4.8	Система контролю версій Git	28
4.9	Висновки до розділу	29
5.	ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	30
5.1	Структура програмного забезпечення.....	30
5.2	Опис бази даних.....	34
5.3	Початкове вікно користувача	38
5.4	Вікно розрахунку концентрації забруднюючої речовини.....	40
5.5	Вікно генерації звіту.....	41
5.6	Допоміжні вікна	43
5.7	Висновки до розділу	44
6.	МЕТОДИКА РОБОТИ КОРИСТУВАЧА.....	45
6.1	Головна сторінка.....	45
6.2	Розрахунок концентрації забруднюючої речовини.....	50
6.3	Побудова звіту	53
6.4	Висновки до розділу	57
	ВИСНОВКИ.....	58
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
	ДОДАТОК А.....	62
	ДОДАТОК Б	65
	ДОДАТОК В	80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ПК – програмний комплекс

ГДК – гранично допустима концентрація

ГІС – геоінформаційна система

БД – база даних

IDE – інтегральне середовище розробки

ОС – операційна система

API – програмний інтерфейс додатку

COM – об’єктна модель компонентів

UML – уніфікована мова моделювання

URL – уніфікований локатор ресурсів

ВСТУП

Однією із сучасних проблем галузі охорони навколишнього середовища залишається інтенсивне забруднення атмосфери різноманітними антропогенними джерелами викидів. Ця проблема стає все більш актуальною зі стрімким розвитком різноманітних промислових підприємств.

Для розв'язання цієї проблеми був розроблений програмний продукт, який моделює шлейф повітряної дисперсії. Головною функцією є швидкий розрахунок концентрації забруднюючих речовин в атмосферному повітрі.

Прогнозування якості довкілля та оцінка можливого впливу на нього викидами від промислових підприємств, автомобільного транспорту та інших видів людської діяльності ґрунтується, як правило, на математичному моделюванні процесів перенесення забруднень у повітряному середовищі. При проведенні розрахунків без використання інформаційних технологій, ймовірність похибки є досить високою, через складність математичних моделей, саме тому автоматизація цього процесу допомагає уникнути цієї похибки в розрахунках, що є дуже важливим при прийнятті рішень та формуванні рекомендацій щодо покращення стану атмосфери при небезпечних викидах.

Окрім звичайної підтримки якості повітря, розроблений програмний продукт є невід'ємним при надзвичайних ситуаціях, які потребують негайного розв'язання проблеми.

Програмний продукт може бути застосований на організаціях, установах та підприємствах, які займаються діяльністю, що пов'язана зі шкідливим впливом на навколишнє середовище.

1 ЗАДАЧА МОДЕЛЮВАННЯ ГАУССОВОГО ШЛЕЙФУ ПОВІТРЯНОЇ ДИСПЕРСІЇ

Метою даної дипломної роботи є створення додатку під операційну систему Windows, який проводить моделювання шлейфу повітряної дисперсії забруднюючих речовин в атмосферному повітрі від різноманітних джерел викидів на основі математичної моделі Гауссового розподілу.

1.1 Постановка задачі

Розроблюваний програмний продукт, має відповідати декільком вимогам, а саме:

- має бути відкритим та зрозумілим для користувача;
- повинен дозволяти користувачеві вводити параметри для подальшого розрахунку забруднення атмосферного повітря;
- має коректно розраховувати значення для подальшого моделювання;
- має коректно розраховувати коефіцієнт небезпеки для життя людини;
- має вірно сформулювати звіт на основі отриманої інформації.

Система моделювання гауссового шлейфу повітряної дисперсії має містити наступний функціонал:

1. Можливість введення необхідних параметрів для розрахунку концентрації речовин у повітрі.
2. Можливість проведення розрахунків концентрації забруднюючих речовин за формулою.
3. Можливість відображення результатів розрахунків.
4. Можливість генерації звітного документа.
5. Можливість використання підказок для коректного вводу даних.

Для реалізації програмного забезпечення та досягнення поставленої мети необхідно виконати наступні задачі:

- провести аналіз існуючих програмних рішень поставленої задачі;
- проаналізувати та обрати засоби реалізації програмного забезпечення;
- розробити алгоритм моделювання гауссового шлейфу повітряної дисперсії;
- розробити користувацький інтерфейс;
- розробити базу даних;
- розробити модуль розрахунку концентрації шкідливої речовини в точці простору з функцією аналізу впливу на здоров'я людини;
- розробити модуль генерації звіту на основі отриманої інформації;
- провести тестування програмного забезпечення.

1.2 Висновки до розділу

У даному розділі була описана мета дипломної роботи, вимоги до розроблюваного програмного забезпечення (ПЗ), описано необхідний функціонал, який має містити ПЗ та дії для досягнення поставленої мети.

2 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ ЗАДАЧІ МОДЕЛЮВАННЯ ШЛЕЙФУ ПОВІТРЯНОЇ ДИСПЕРСІЇ

Наразі існує досить багато комп'ютерних та веб – застосунків, які допомагають здійснювати екологічний моніторинг довкілля. Багато з них дають можливість проводити дослідження з різних областей моніторингу.

Оскільки атмосферне повітря відіграє в географічній оболонці надзвичайно велику роль і виконує багато важливих функцій, то формування заходів з охорони повітря є необхідною складовою екологічного моніторингу.

Для прискорення прийняття важливих рішень при забрудненні атмосферного повітря різними забруднюючими речовинами та підвищення точності розрахунків, було розроблено програмні комплекси, які допомагають у розв'язанні цих питань.

2.1 Опис програмних рішень

Основні функції систем моделювання повітряної дисперсії забруднюючих речовин є :

1. розрахунки – проведення розрахунків викидів та концентрацій забруднюючих речовин від різних виробництв;
2. звітування – представлення комплексних звітів, розроблених на основі результатів розрахунків;
3. графічне відображення – побудова графічних представлень у вигляді карт та різноманітних схем;
4. рекомендації – вироблення рекомендацій для подальших дій щодо охорони навколишнього середовища, на основі проведених розрахунків.

В таблиці 2.1 представлено співвідношення програмних комплексів, які виконують моделювання шлейфу повітряної дисперсії до можливих особливостей цих застосунків.

Таблиця 2.1 – Співвідношення аналогічних систем до їх особливостей

Система	Особливості					
	Функція звітування	Функція графічного відображення	Функція генерації рекомендацій	Можливість розрахунку для груп об'єктів	Можливість вибору метода розрахунків	Безкоштовний доступ
ПК «Призма – предприятие»	+	+	-	-	-	-
ПК «ЭПК Роса»	+	+	+	+	+	+
ПК «RECASS NT»	+	-	-	-	+	-
ПК «НОСТРАДАМУС»	+	+	+	-	-	-
ПК «ЭРА – Воздух»	+	+	+	-	-	-
ПК «Атмосфера – ПДВ»	+	+	-	+	-	-

Програмний комплекс «Призма – предприятие» [17] – комплекс, який складається з дев'яти модулів, які використовуються для формування заходів з охорони повітря для різних підприємств. Програма дозволяє провести розрахунок забруднення атмосфери та побудувати графічне представлення для одного підприємства.

Основні задачі, реалізованих модулів [17]:

1. Модуль «Норма – предприятие» – вирішення зворотного завдання розсіювання, автоматизований розрахунок нормативів допустимих викидів забруднюючих речовин;
2. Модуль «Санзона – предприятие» – автоматизоване проектування розрахункової та нормативної санітарно – захисної зони підприємства. Визначення розмірів та графічне представлення з урахуванням рози вітрів;
3. Модуль «Том ПДВ – предприятие» – формування таблиць тому ПДВ та шаблону пояснювальної записки на підставі даних інвентаризації, результатів розрахунку полів розсіювання та нормування викидів;
4. Модуль «ГАЗ» – розрахунок забруднення атмосфери від джерел компресорних станцій, які забезпечують функціонування магістральних та інших газопроводів;
5. Модуль «Экспорт – ArcView» – пересилання різних даних з ПК «Призма» у ГІС ArcView;
6. Модуль «Импорт – AutoCAD» – прийом графічних файлів топооснови місцевості з AutoCAD у програмний комплекс «Призма»
7. Модуль «Экспорт – AutoCAD» – пересилання результатів розрахунку полів приземних концентрацій забруднюючих речовин (ізоліній) та топооснови місцевості в AutoCAD;
8. Модуль «Экспорт – MapInfo» – пересилання різних даних з ПК «Призма» у ГІС MapInfo;
9. Модуль «Импорт – ЭкоРасчет» – прийом результатів розрахунку з програмного комплексу «Модульний ЕкоРозрахунок» у БД.

Програмний комплекс «ЭПК Роса» [20] – допомагає підприємствам чи групам з декількох об'єктів проводити розрахунки розсіювання забруднюючих речовин у атмосфері з використанням різних варіантів та режимів розрахунку. Результатом є побудова ізоліній полів розсіювання забруднюючих речовин та визначені точки максимальної концентрації цих речовин в межах санітарно – захисної зони та зони житлової забудови.

Можливості ПК «ЭПК Роса» [20]:

1. Розрахунок розсіювання забруднюючих речовин в атмосфері повітря;
2. Побудова проекту нормативів гранично допустимих викидів забруднюючих речовин в атмосферу – розробляється на підставі даних інвентаризації викидів забруднюючих речовин вже існуючого підприємства або для нового підприємства, яке реконструюється;
3. Інвентаризація джерел викидів забруднюючих речовин в атмосферу – створення екологічної моделі виробництва, яка описує джерела забруднення атмосфери, технологічне обладнання і режими роботи, шкідливі речовини, газоочисні установки;
4. Вибір методики розрахунку викидів забруднюючих речовин в атмосферу – 50 методик з різних галузей промисловості та реалізований конструктор методик, який дозволяє користувачу вводити нові розрахункові модулі для подальшого використання;
5. Розрахунок категорії небезпеки підприємства;
6. Виконання додаткової звітності – обґрунтування розрахунків, протоколи розрахунків забруднення атмосфери.

Програмний комплекс «RECASS NT» [13] – комплекс для підтримки прийняття рішень при забрудненні навколишнього середовища при хімічних та радіаційних аваріях, викидах вулканічного попелу та продуктів горіння лісових та торф'яних пожеж. Основу комплексу становить система прогнозу та аналізу забруднення природнього середовища, яка забезпечує [13]:

1. Безперервне ведення бази прогностичних та поточних даних про стан атмосфери та поверхневих вод;
2. Моделювання поширення природних та аварійних забруднень в атмосфері, їх випадання на поверхню землі та перенесення поверхневими водами;

3. Розрахунок можливих очікуваних доз для населення і вироблення рекомендацій щодо заходів для захисту населення при викидах токсичних речовин та радіаційних аваріях;
4. Підготовку прогнозів аварійного забруднення, звітних форм, рекомендацій щодо контрзаходів та розсилання їх зацікавленим організаціям.

Програмний комплекс «НОСТРАДАМУС» [18] – призначений для швидкого прогнозування радіаційної обстановки у ближній і дальній зоні при викидах радіоактивних речовин. Використовується для прийняття рішень в реальному часі радіаційної аварії, дозволяє з високою точністю прогнозувати розвиток радіаційної обстановки в окремих точках, або на всій території розрахункової області. Результати подаються на картографічній основі. Лагранжева траєкторна стохастична модель атмосферного перенесення домішок є основою для розрахунків – вона дозволяє розраховувати поширення радіонуклідів на відстані в сотні кілометрів, врахує вплив рельєфу місцевості, тимчасовості і неоднорідності, зміни вітрового поля та опадів на процес розсіювання домішок, а також може розглядати джерело довільної конфігурації та форми з параметрами, які можуть змінюватись з часом.

Програмний комплекс «ЭРА – Воздух» [19] – призначений для автоматизації розрахунків у галузі екологічного нормування та проектування, розробки природоохоронної документації для діючих та проєктованих підприємств. Виконує наступні функції [19]:

1. Проведення інвентаризації джерел викидів забруднюючих речовин з підготовкою та оформленням результатів у формі звіту;
2. Розрахунок розсіювання в атмосферному повітрі та зведення розрахунків по містам та підприємствах;
3. Підготовка та випуск карт з ізолініями концентрацій, побудова розрахункової та нормативної санітарно – захисної зони;
4. Підготовка звітності за формою 2 – ТП.

Програмний комплекс «Атмосфера – ПДВ» [16] – призначений для вирішення проблем та завдань у сфері охорони атмосфери повітря. Програма може використовувати вихідні дані, які були підготовлені програмами з серії «Еколог». Комплекс «Атмосфера – ПДВ» дозволяє [16]:

1. Проводити інвентаризацію джерел, які здійснюють викиди забруднюючих речовин;
2. Проводити розрахунки приземних концентрацій забруднюючих речовин з використанням методики ОНД – 86, а також проводити розрахунки разових та середньорічних концентрацій за формулами «Гауссового факела»;
3. Створювати та випустити комплект документації тому ПДВ та карти – схеми з ізолініями розрахункових концентрацій забруднюючих речовин, джерел забруднення, меж житлових та санітарно–захисних зон;
4. Розраховувати валові викиди від джерел викидів за реалізованими методиками або методиками, які може розробити сам користувач.

2.2 Висновки до розділу

Аналізуючи системи, які були описані в цьому розділі, можна зробити висновок, що їх основною проблемою є відсутність вільного доступу. ПК «ЕПК Роса» є єдиною системою, що має безкоштовну демоверсію, яка дає змогу краще ознайомитись з застосунком, його функціоналом та зрозуміти чи підходить цей додаток користувачу. Також ПК «ЕПК Роса» має найбільшу кількість переваг серед конкурентів, найбільш вагомими з яких – можливість вибору метода розрахунку та розробка рекомендаційних подальших дій, оснований на результатах розрахунків.

Особливістю розробленого ПЗ, яка відрізняє дану систему від вже розроблених аналогічних систем є те, що вона проводить аналіз ризику впливу викидів забруднюючих речовин на здоров'я людини.

3 МАТЕМАТИЧНА МОДЕЛЬ

Перед розробкою застосунку необхідно вибрати модель забруднення атмосферного повітря, за допомогою якої буде здійснюватися моделювання поширення забруднюючих речовин в атмосферному повітрі за заданих метеоумов.

Були розглянуті три моделі, які дозволяють проводити розрахунки концентрації: Гауссова модель, модель турбулентної дифузії та методика ОНД – 86.

Ці моделі є найпоширенішими та підходять для моделювання забруднення атмосферного повітря.

3.1 Порівняння існуючих методів

Дисперсійні моделі або моделі розсіювання повітря зазвичай використовують при оцінюванні концентрації токсинів або забруднюючих речовин, які викидаються з різноманітних джерел.

Існує три найрозповсюджені моделі розсіювання повітря:

1. Гауссова модель;
2. Модель турбулентної дифузії;
3. Методика ОНД – 86.

Ступінь забруднення атмосфери небезпечними речовинами з деяких джерел, відповідно до моделі турбулентної дифузії, визначається по найбільшому значенню приземної разової концентрації цих речовин. Концентрація речовин встановлюється з урахуванням несприятливих метеорологічних умовах в момент, коли швидкість вітру досягає небезпечного значення і відбувається інтенсивний турбулентний обмін в приземному шарі. Метод використовується для вирішення завдань довгострокового та оперативного прогнозування [14].

Методика ОНД – 86 дозволяє проводити розрахунок максимально можливого розподілу концентрації викидів усереднених в інтервалі 20 – 30 хвилин за умови, що стан атмосфери є помірно нестійким. Недоліком методики є те, що вона не враховує шорсткість підстильної поверхні та клас стійкості атмосфери. Проведення розрахунків концентрації можливе при відстані від джерела не більше ніж 100 км [15].

Гауссова модель широко використовується для визначення локальної дисперсії. Ця модель, аналітичне рішення якої отримано для стаціонарного та нестаціонарного розподілів, створена на основі уявлень Гаусса про статистичні розподіли. При цьому можливий розгляд деякої кількості частинок речовини, що викидаються з точкового джерела [11].

Гауссова модель найчастіше використовується в більшості країн в нормативних документах на державному рівні для контролювання та прогнозування якості повітря, тому саме ця модель була обрана для розробки програмного продукту.

3.2 Загальна характеристика Гауссового або нормального розподілу

Змінна x нормально розподілена, якщо виконується співвідношення для функції густини $f(x)$:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x - \mu)^2}{2\sigma^2}\right] \quad (3.1)$$

де σ – стандартне відхилення (будь – яке дійсне число більше нуля), яке розширює або стискає дзвонову криву (площа під кривою завжди дорівнює одиниці) ;

μ – будь – яке дійсне число, яке визначає положення максимального значення $f(x)$ на осі x .

Функція Гауссового розподілу у нормованому вигляді – площа, обмежена кривою, дорівнює одиниці.

Загалом близько 68% площі під кривою перебуває в інтервалі між $+\sigma$ і $-\sigma$, а близько 95% – між $+2\sigma$ і -2σ .

Це зростання ширини функції розподілу зі збільшенням σ має важливий фізичний сенс при вивченні розсіювання забруднювачів у атмосфері.

Таким чином, μ і σ є характеристиками положення та форми кривої Гауссового розподілу.

Рівняння для розсіювання забруднювачів матимуть вигляд подвійного Гауссового розподілу. Подвійний Гауссовий розподіл відносно двох координат, (y і z), визначається простим перемноженням двох Гауссових розподілів по кожній з координатних осей. Отже:

$$f(y, z) = \frac{1}{2\pi\sigma_y\sigma_z} \exp\left[\frac{-(y - \mu_y)^2}{2\sigma_y^2} + \frac{-(z - \mu_z)^2}{2\sigma_z^2}\right] \quad (3.2)$$

Загальні характеристики Гауссового або нормального розподілу описані на основі джерела [11].

3.3 Основне рівняння розсіювання домішок у Гауссовій теорії

Усі моделі оцінки розсіювання забруднюючих речовин у атмосфері зводяться до однієї й тієї функції розподілу концентрації забруднювача – Гауссового розподілу.

Агентство охорони навколишнього середовища Сполучених Штатів Америки використовує як основу моделі розсіювання статистичну Гауссову теорію.

Дрібні частинки та газові домішки розсіюються в атмосфері за законами нормального розподілу. Струмінь диму поширюється вздовж (паралельно) осі x .

Овальні фігури є перетином спрощеного струменя. Профілі концентрацій підпорядковуються законам нормального розподілу.

Основне рівняння розсіювання в Гауссовій теорії має вигляд:

$$C(x, y, z, H) = \frac{M}{2\pi u \sigma_y \sigma_z} \exp\left[-\left(\frac{y^2}{2\sigma_y^2}\right)\right] \left\{ \exp\left[\frac{-(z-H)^2}{2\sigma_z^2}\right] + \exp\left[\frac{-(z+H)^2}{2\sigma_z^2}\right] \right\} \quad (3.3)$$

де C – концентрація у певній точці з координатами x, y, z , г/м³;

M – потужність викиду, г/с;

u – середня швидкість вітру, м/с;

H – висота віртуального джерела (сума геометричної висоти труби та початкового підйому струменя), м.

При розгляді Гауссового струменя прийняті такі припущення:

1. Розсіювання струменя в горизонтальній та вертикальній площинах описується Гауссовим розподілом зі стандартними відхиленнями розподілу концентрацій σ_y та σ_z по осях y та z відповідно.

2. Середня швидкість вітру, що діє на струмінь є постійною у всьому шарі поширення струменя, напрям вітру не змінюється.

3. Потужність викиду стала.

4. Немає осадження або взаємодії струменя з поверхнею, яка знаходиться під ним.

Основне рівняння розсіювання домішок у Гауссовій теорії описане на основі джерела [11].

3.4 Окремі випадки розрахунку приземних концентрацій

Для розрахунку концентрації біля поверхні землі (координата $z = 0$) рівняння набуває вигляду:

$$C(x, y, 0, H) = \frac{M}{2\pi u \sigma_y \sigma_z} \exp\left[-\left(\frac{y^2}{2\sigma_y^2}\right)\right] \exp\left[\frac{-H^2}{2\sigma_z^2}\right] \quad (3.4)$$

Для розрахунку концентрації біля поверхні землі, вздовж осьової лінії струменя (координати $z = 0$, $y = 0$) рівняння набуває вигляду:

$$C(x, 0, 0, H) = \frac{M}{\pi u \sigma_y \sigma_z} \exp\left[\frac{-H^2}{2\sigma_z^2}\right] \quad (3.5)$$

Для розрахунку концентрації біля поверхні землі, вздовж осьової лінії струменя для наземного джерела (координати $z = 0$, $y = 0$, $H = 0$) рівняння набуває вигляду:

$$C(x, 0, 0, H) = \frac{M}{\pi u \sigma_y \sigma_z} \quad (3.6)$$

Окремі випадки розрахунку приземних концентрацій описані на основі джерела [12].

3.5 Практичне використання рівнянь Гауссової теорії

Для практичного використання наведених рівнянь крім фізичних даних (координати, потужність викиду, ефективна висота джерела) необхідно знати величини u , σ_y , σ_z .

Зазвичай використовують значення швидкості u , усереднене по всьому струменю. Через труднощі її визначення приймають середню швидкість вітру на рівні горловини труби.

Коефіцієнт бічної (поперечної) дифузії σ_y характеризує собою горизонтальне розширення струменя у напрямі, перпендикулярному напрямку руху.

Коефіцієнт вертикальної дифузії σ_z характеризує розширення струменя по вертикалі.

Значення σ_y , σ_z (горизонтальне та вертикальне стандартне відхилення)

залежать від положення точки x у напрямку вітру від джерела та від умов стійкості атмосфери.

Значення σ_y , σ_z визначають за діаграмами, отриманими експериментальним шляхом.

A, B, C, D, E, F – різні класи стійкості атмосфери:

A – відповідає найбільшій нестійкості (ясне небо, висота сонця над горизонтом понад 60° , типовий літній сонячний день пообіді. Дуже конвективна атмосфера);

F – відповідає максимальній стійкості.

При оцінці розсіювання газів зазвичай вибирають такий клас стійкості, який є типовим для даного району і дає найгірші характеристики з погляду забруднення атмосфери.

Практичне використання рівняння Гауссової теорії описане на основі джерела [11].

3.6 Висновки до розділу

Оскільки більшість джерел забруднень діють безперервно, тобто постійно випаровують забруднюючі речовини та метеорологічні умови є змінними, для подальшої роботи була обрана Гауссова модель повітряної дисперсії, оскільки вона більше підходить для поставленого завдання та приймає мінімальну кількість параметрів на вхід, на відміну від інших моделей.

4 ЗАСОБИ РЕАЛІЗАЦІЇ

Перед початком розробки програмного продукту поставленої задачі необхідно було обрати засоби реалізації, які найефективніше зможуть допомогти в розробці та будуть мати увесь необхідний для цього функціонал.

Після аналізу існуючих засобів та усіх їх переваг і недоліків було обрано:

- Мову програмування C#;
- Середовище розробки Visual Studio;
- Програмна технологія .NET Framework;
- Інтерфейс програмування Windows Form;
- Технологія COM Interop;
- Мова структурованих запитів SQL;
- Система керування базами даних MySQL;
- Система контролю версій Git;

4.1 Мова програмування C#

Мова програмування C# [4] – є проста, сучасна, об'єктно – орієнтована та типізована мова. Вона походить з сімейства мов програмування C і буде одразу знайомою для програмістів, які використовують такі мови програмування, як: Java, C++ та C. C# стандартизовано ISO/IEC, як стандарт ISO/IEC 23270 і ECMA International, як стандарт ECMA – 334. Компілятор C# для .NET Framework від Microsoft є відповідною реалізацією обох цих стандартів.

Деякі функції C# допомагають створювати надійні та довговічні програми, а саме [4]:

- автоматичне відновлення пам'яті, зайнятої невикористаними об'єктами, за допомогою збору сміття;

- обробка винятків забезпечує структурований і розширений підхід до виявлення та виправлення помилок;
- унеможливлення читання з неініціалізованих змінних, індексування масивів за їх межами або виконання неперевірених перетворень типів .

Щоб гарантувати, що бібліотеки та програми написані на C# розвиваються з часом відповідним чином, велика увага приділена версійності в дизайні C#. Більшість мов програмування не приділяють достатньої уваги цьому питанню. В результаті програми, які були написані за допомогою інших мов програмування, ламаються набагато частіше, при вводі нових версій залежних бібліотек [4].

4.2 Середовище розробки Visual Studio

Інтегроване середовище розробки (IDE) – це програма, яка має великий функціонал та підтримує багато різних аспектів розробки програмного забезпечення.

Visual Studio [7] – це креативна стартова панель, яка може бути використана для створення коду, його редагування та налагодження, а надалі і для публікації програми. Окрім налагоджувача та стандартного редактора, які мають більшість IDE, Visual Studio включає в себе інструменти завершення коду, компілятори, графічні дизайнери та багато інших функцій, які покращують процес розробки ПЗ.

Visual Studio є доступною для ОС Mac і Windows. Visual Studio для Mac має велику кількість функцій, що й Visual Studio для Windows і є оптимізованою для розробки мобільних і кросплатформних додатків. У Visual Studio існує три версії: Community, Professional і Enterprise [7].

Функції Visual Studio, які значно допомагають покращити продуктивність під час розробки ПЗ [7]:

1. Завитки та швидкі дії. Завитки – це хвилясті підкреслення, які сповіщають про проблеми або помилки у коді. Лампочка, яка може з'явитися зліва від них, показує швидкі дії, які можна виконати, щоб виправити помилку;
2. Очищення коду. Натиснувши спеціальну кнопку «Очищення коду», можна відформувувати код та застосувати виправлення, які були запропоновані налаштуваннями стилю коду;
3. Рефакторинг. Процес рефакторингу охоплює: заміну порядку параметрів методу, вилучення одного або кількох рядків коду в новий метод та інтелектуальне перейменування змінних;
4. Спільна робота. Функція «Спільна робота» дозволяє спільно редагувати та налагоджувати код з іншими людьми в режимі реального часу, незалежно від типу програми чи мови програмування;
5. «Ієрархія викликів». У вікні «Ієрархія викликів» показано методи, які викликає вибраний метод;
7. «Перейти до визначення». Функція «Перейти до визначення» перенесе вас безпосередньо до розташування визначення функції або типу.

4.3 Програмна технологія .NET Framework

.NET Framework [9] – це безкоштовна платформа для розробників з відкритим кодом для створення багатьох різних типів додатків. .NET Framework допомагає швидше розробляти високоякісні програми. У поєднанні з великими бібліотеками класів, звичайними API, багатомовною підтримкою та потужними інструментами сімейства Visual Studio, .NET Framework є найпродуктивнішою платформою для розробників.

4.4 Інтерфейс програмування Windows Forms

Windows Forms [3] – це платформа інтерфейсу користувача, яка використовується для створення типових додатків Windows. За допомогою візуального конструктора в Visual Studio, технологія Windows Forms може забезпечити один з найефективніших способів створення типових програм. Процес створення класичних додатків значно полегшується з використанням функції розміщення всіляких візуальних елементів керування методом їх перетягування.

Windows Forms дає можливість розробляти графічно складні програми, які можна легко використовувати, оновлювати і з якими зручно працювати як в автономному режимі, так і в мережі [3].

Windows Forms – є технологією інтерфейсу для .NET, яка представляє з себе певний набір бібліотек, що допомагають спрощувати виконання типових завдань, наприклад як: запис в файлову систему або читання з неї. За допомогою використання середовищ розробки, таких як Visual Studio, з'являється можливість у простому створенні інтелектуальних клієнтських програм Windows Forms, які запитують користувача інформацію на введення, відображають інформацію у різних формах і можуть взаємодіяти з віддаленими комп'ютерами [3].

4.5 Технологія COM Interop

COM Interop – це технологія, яка включена до .NET CLR і дозволяє об'єктам .NET взаємодіяти з об'єктами COM, і навпаки [2].

Завданням Interop полягає у забезпеченні доступу до компонентів COM без необхідності модифікації оригінальних компонентів. Ця технологія допомагає зробити типи .NET еквівалентними COM. Окрім цього, Interop дає можливість розробникам COM отримувати доступ до об'єктів, які керуються, так само просто, як і доступ до всіх інших об'єктів COM [2].

.NET Framework створює записи в реєстрі та бібліотеку типів під час реєстрації компонента. Це виконується за допомогою спеціальної утиліти RegAsm.exe, яка виконує процес експорту керованих типів в бібліотеку типів і реєструє керовані компоненти як звичайний COM – компонент [2].

При використанні програми Tlbimp.exe, Interop може бути створено для вже існуючого компонента COM. Ця програма постачається разом із .NET Framework SDK. Вона створює набір проміжних класів, які реалізуються мовою CIL, а потім інкапсулюються у складання [2].

4.6 Мова структурованих запитів SQL

Мова структурованих запитів SQL – це стандартизована мова програмування, яку використовують для управління реляційними базами даних та для виконання різних операцій над даними цих баз [5].

Основні цілі використання мови SQL [5]:

- створення, додавання, оновлення та видалення рядків даних;
- зміна структури індексів та таблиць бази даних;
- отримання різних підмножин інформації з систем управління реляційними базами даних.

4.7 Система управління базами даних MySQL

MySQL – найпопулярніша система керування базами даних SQL з відкритим вихідним кодом, розроблена, поширена та підтримувана Oracle Corporation [8].

Переваги використання MySQL [8]:

1. Розмір і швидкість – MySQL може працювати на дуже скромному обладнанні і створювати невелике навантаження на системні ресурси.

2. Простота встановлення – MySQL можна встановити без великої кількості складних конфігурацій.
3. Увага до стандартів – переміщення коду з однієї бази даних до іншої ніколи не є тривіальним, але MySQL робить розумну роботу з забезпечення стандартного середовища та розвиває більше функцій.
4. Увага до користувачів – MySQL дуже гнучка організація, яка постійно стежить за потребами користувачів.
5. Простий інтерфейс – MySQL легко використовувати як частину більшої програмної системи, більшість мов програмування мають бібліотеки функцій для використання MySQL.

4.8 Система контролю версій Git

Git – це безкоштовна розподілена система контролю версій з відкритим вихідним кодом, призначена для швидкої та ефективної роботи з усім, від малих до дуже великих проектів [1].

Особливості Git [1]:

- Git дозволяє мати кілька локальних гілок, які можуть бути повністю незалежними одна від одної. Створення, об'єднання та видалення цих ліній розробки займають секунди;
- З Git майже всі операції виконуються локально, що дає йому величезну перевагу в швидкості в централізованих системах, які постійно повинні спілкуватися з сервером;
- Модель даних, яку використовує Git, забезпечує криптографічну цілісність кожного біта вашого проекту.

4.9 Висновки до розділу

В даному розділі були детально описані засоби реалізації, які використовуються для розробки системи моделювання шлейфу повітряної дисперсії забруднюючих речовин. Вибір кожного засобу був обґрунтований, зважаючи на сучасні вимоги до розробки ПЗ.

5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Під час розробки програмного забезпечення було створено класичний додаток для пристроїв, які базуються на ОС Windows. Користувацький інтерфейс був розроблений за допомогою Windows Forms. Програма складається з семи форм, а саме: головна форма, форма розрахунку концентрації шкідливої речовини, форма з таблицею речовин, форма генерації звіту, форма звіту, допоміжна форма з класами стабільності атмосфери і допоміжна форма з інформацією про координати розрахунків.

5.1 Структура програмного забезпечення

При розробці додатку використовувались наступні бібліотеки:

- `MySQL.Data.MySqlClient` – містить в собі класи, які допомагають працювати з `MySQL`;
- `System` – містить в собі базові і фундаментальні класи, які визначають типи значень, обробники подій і самі події, атрибути і інтерфейси, які часто використовуються під час розробки додатків;
- `System.Windows.Forms` – містить в собі класи, які допомагають при створенні додатків для ОС Windows і дозволяють ефективно використовувати можливості інтерфейсу, призначеного для користувача;
- `System.Reflection` – містить в собі типи, які призначені для отримання відомостей про збірки, модулі, члени, параметри та інші об'єкти в керованому коді шляхом обробки їх метаданих [5];
- `Microsoft.Office.Interop.Word` – містить в собі класи, які допомагають взаємодіяти об'єктам `.NET` з об'єктами `COM`.

Загальна структура розробленого програмного забезпечення представлена на

рисунку 5.1.

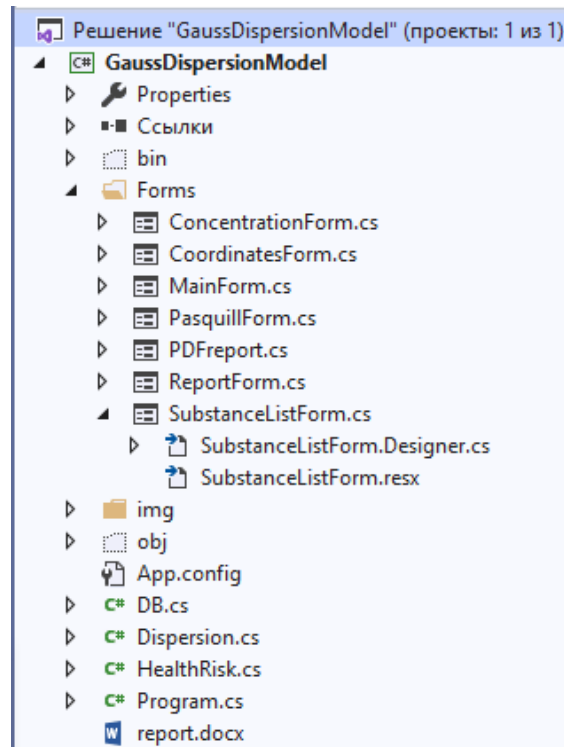


Рисунок 5.1 – Загальна структура ПЗ

Для розробки додатку бути створені наступні класи:

- Program.cs – містить метод Main(), який представляє собою точку входу роботи;
- DB.cs – містить в собі методи для роботи з БД;
- Dispersion.cs – містить атрибути опису викидів забруднюючих речовин та умов, за яких відбуваються ці викиди, а також методи для розрахунку концентрації забруднюючих речовин;
- HealthRisk.cs – містить методи для розрахунку коефіцієнта небезпеки для життя людини та рівня впливу на її здоров'я;
- MainForm.cs – містить методи для роботи користувача з головною формою;
- ConcentrationForm.cs – містить методи для роботи користувача з формою

- розрахунку концентрації шкідливої речовини;
- ReportForm.cs – містить методи для роботи користувача з формою генерації звіту;
 - PDFreport.cs – містить методи для роботи користувача з формою звіту;
 - SubstanceListForm.cs – містить методи для роботи користувача зі списком речовин;
 - CoordinatesForm.cs – містить методи для роботи користувача з допоміжною формою з інформацією про координати розрахунків;
 - PasquillForm.cs – містить методи для роботи користувача з допоміжною формою з класами стабільності атмосфери.

На рисунку 5.2 представлена діаграма прецедентів «Система моделювання гауссового шлейфу повітряної дисперсії», яка демонструє відношення між прецедентами та акторами.

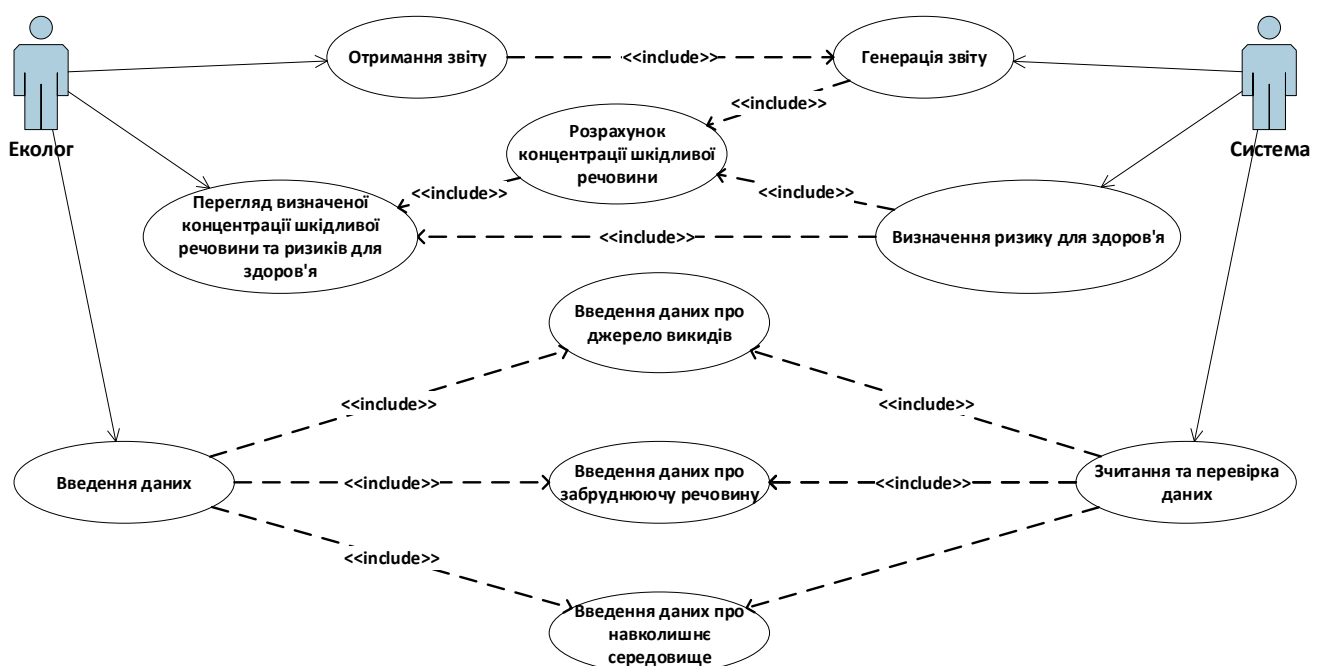


Рисунок 5.2 – Діаграма прецедентів «Система моделювання гауссового шлейфу повітряної дисперсії»

На рисунку 5.3 представлена діаграма діяльності системи моделювання гауссового шлейфу повітряної дисперсії забруднюючих речовин.

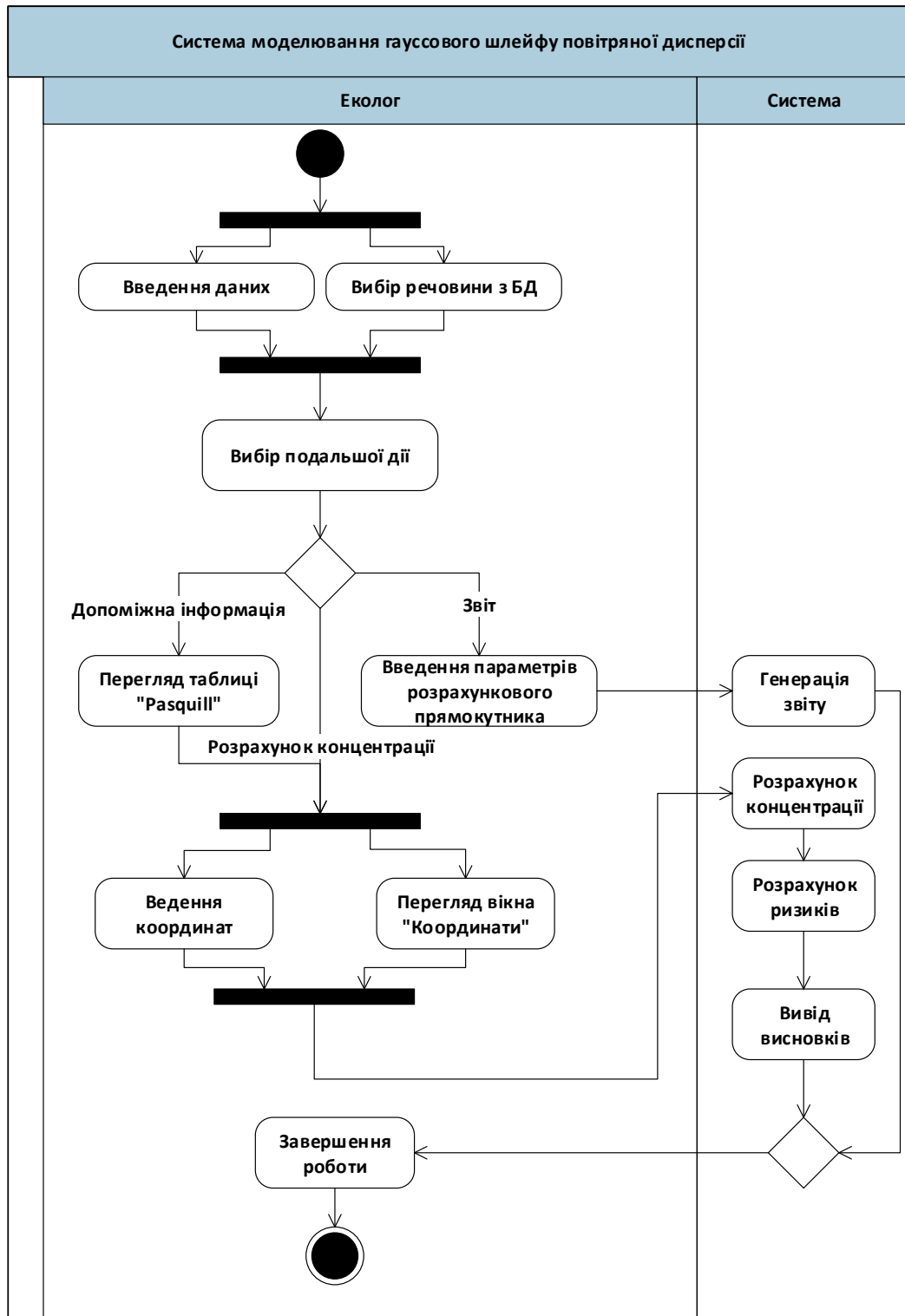


Рисунок 5.3 – Діаграма діяльності програмного забезпечення

Дана діаграма діяльності демонструє аспекти поведінки розробленої системи.

5.2 Опис бази даних

В ході розробки бази даних було створено 6 таблиць, а саме: «atmospheric_Stability», «p», «cd», «ab», «locality», «substance». Всі таблиці повністю відповідають нормальним формам. Основною метою було виключити надлишкове дублювання даних, які є причиною протиріччя в базі даних чи значно ускладнюють обробку бази.

Опис таблиці «atmospheric_Stability», яка містить в собі інформацію щодо класів стабільності атмосфери, представлений в таблиці 5.1.

Таблиця 5.1 – Опис таблиці «atmospheric_Stability»

Назва поля	Тип даних	Значення
id	int	Унікальний номер класу стабільності атмосфери
name	varchar(45)	Назва класу стабільності атмосфери

Опис таблиці «p», яка містить в собі інформацію про поправочного коефіцієнту p , представлений в таблиці 5.2.

Таблиця 5.2 – Опис таблиці «p»

Назва поля	Тип даних	Значення
id	int	Унікальний номер коефіцієнта
atmStab_id	int	Клас стабільності атмосфери
locality_id	int	Місцевість

Продовження таблиці 5.2

value	double	Значення коефіцієнта p
-------	--------	--------------------------

Опис таблиці «cd», яка містить в собі інформацію про коефіцієнти c та d , представлений в таблиці 5.3.

Таблиця 5.3 – Опис таблиці «cd»

Назва поля	Тип даних	Значення
id	int	Унікальний номер коефіцієнта
atmStab_id	int	Клас стабільності атмосфери
c_value	double	Значення коефіцієнта c
d_value	double	Значення коефіцієнта d

Опис таблиці «ab», яка містить в собі інформацію про коефіцієнти a та b , представлений в таблиці 5.4.

Таблиця 5.4 – Опис таблиці «ab»

Назва поля	Тип даних	Значення
id	int	Унікальний номер коефіцієнта
atmStab_id	int	Клас стабільності атмосфери
x_min	double	Початок діапазона значення змінної x
x_max	double	Кінець діапазона значення змінної x
a_value	double	Значення коефіцієнта a
b_value	double	Значення коефіцієнта b

Опис таблиці «locality», яка містить в собі інформацію про типи місцевості, представлений в таблиці 5.5.

Таблиця 5.5 – Опис таблиці «locality»

Назва поля	Тип даних	Значення
id	int	Унікальний номер місцевості
name	varchar(45)	Назва місцевості

Опис таблиці «substance», яка містить в собі інформацію речовини, представлений в таблиці 5.6.

Таблиця 5.6 – Опис таблиці «substance»

Назва поля	Тип даних	Значення
id	int	Унікальний номер речовини
name	varchar(45)	Назва місцевості
rfc	double	Безпечний рівень впливу
gdk	double	Гранично допустима концентрація речовини

Первинний ключ – унікальне, непусте значення в межах зазначеного стовпця таблиці. Серед створених первинних ключів:

- атрибут 'id' в таблиці «atmospheric_Stability»;
- атрибут 'id' в таблиці «р»;
- атрибут 'id' в таблиці «cd»;
- атрибут 'id' в таблиці «ab»;
- атрибут 'id' в таблиці «locality»;
- атрибут 'id' в таблиці «substance».

Зовнішній ключ – ключ, який використовується для з'єднання двох таблиць одна з одною. Серед створених зовнішніх ключів:

- атрибут 'atmStab_id' в таблиці «р», який посилається на атрибут 'id' в таблиці «atmospheric_Stability».
- атрибут 'atmStab_id' в таблиці «cd», який посилається на атрибут 'id' в таблиці «atmospheric_Stability».

- атрибут 'atmStab_id' в таблиці «ab», який посилається на атрибут 'id' в таблиці «atmospheric_Stability».
- атрибут 'locality_id' в таблиці «р», який посилається на атрибут 'id' в таблиці «locality».

Сервер MySQL має спеціальний об'єкт, який дозволяє зберігати запити. Цей об'єкт називається збережена процедура.

Процедура – це відкомпільований запит, який знаходиться на сервері. Вони потрібні для того, щоб підвищити продуктивність роботи з базами даних. Якщо використовувати процедури, клієнту потрібно буде відправляти лише одну маленьку частину коду для того, щоб викликати збережену процедуру та отримати запит. Також запит, який знаходиться в процедурі, компілюється лише один раз, що значно економить час на отримання результату.

Серед створених збережених процедур :

- 1) «atm_stab» – процедура, яка визначає унікальний номер класу стабільності атмосфери по її назві;
- 2) «a» – процедура, яка видає значення коефіцієнта a , зважаючи на обраний клас стабільності атмосфери та значення змінної x ;
- 3) «b» – процедура, яка видає значення коефіцієнта b , зважаючи на обраний клас стабільності атмосфери та значення змінної x ;
- 4) «c» – процедура, яка видає значення коефіцієнта c , зважаючи на обраний клас стабільності атмосфери;
- 5) «d» – процедура, яка видає значення коефіцієнта d зважаючи на обраний клас стабільності атмосфери;
- 6) «locality» – процедура, яка визначає унікальний номер місцевості по її назві.

Всі збережені процедури використовуються для розрахунку концентрації шкідливої речовини в атмосфері.

Концептуальна модель бази даних представлена на рисунку 5.4.

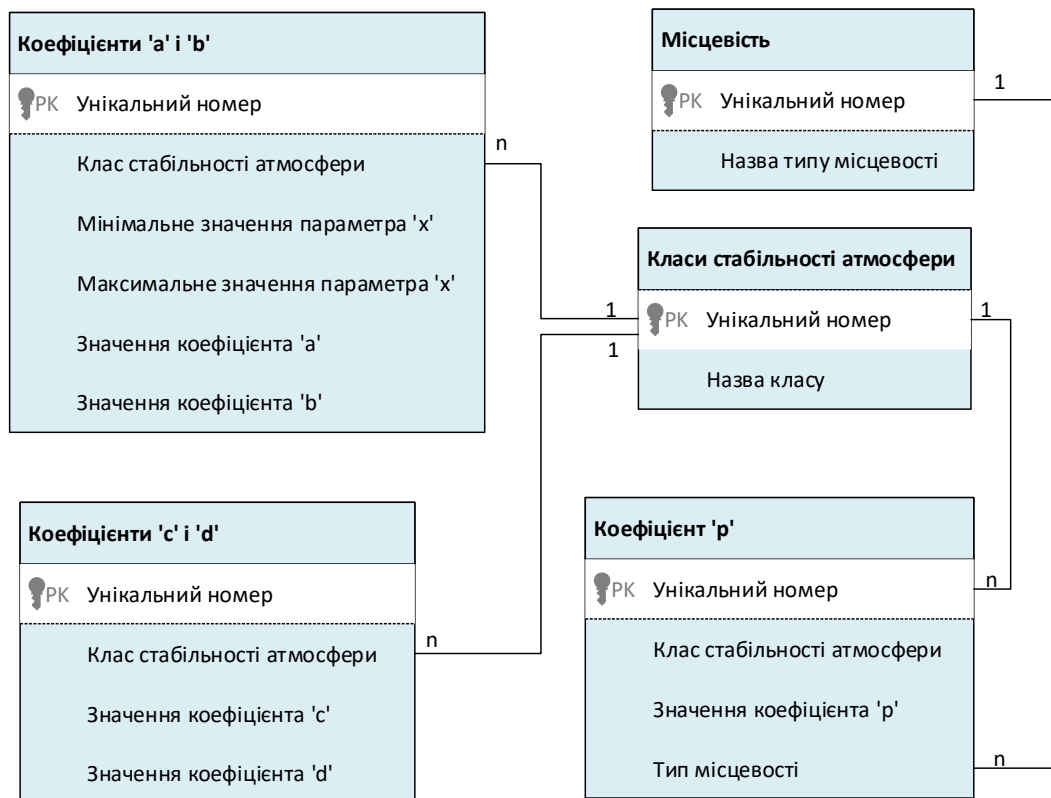


Рисунок 5.4 – Концептуальна модель БД

Концептуальна модель БД – це модель предметної області. Мета побудови цієї моделі – наочна демонстрація зв'язку між об'єктами. Саме об'єкти та взаємозв'язки являються основними компонентами моделі.

5.3 Початкове вікно користувача

Початкове вікно користувача – це головна форма, в якій користувачу необхідно ввести дані про забруднюючу речовину, джерело викиду та навколишнє середовище. Далі користувач має можливість обрати які саме дії він хоче зробити: подивитись допоміжну інформацію, отримати розрахунок концентрації забруднюючої речовини або згенерувати звіт.

Фрагмент структури класу «MainForm» представлений на рисунку 5.5.

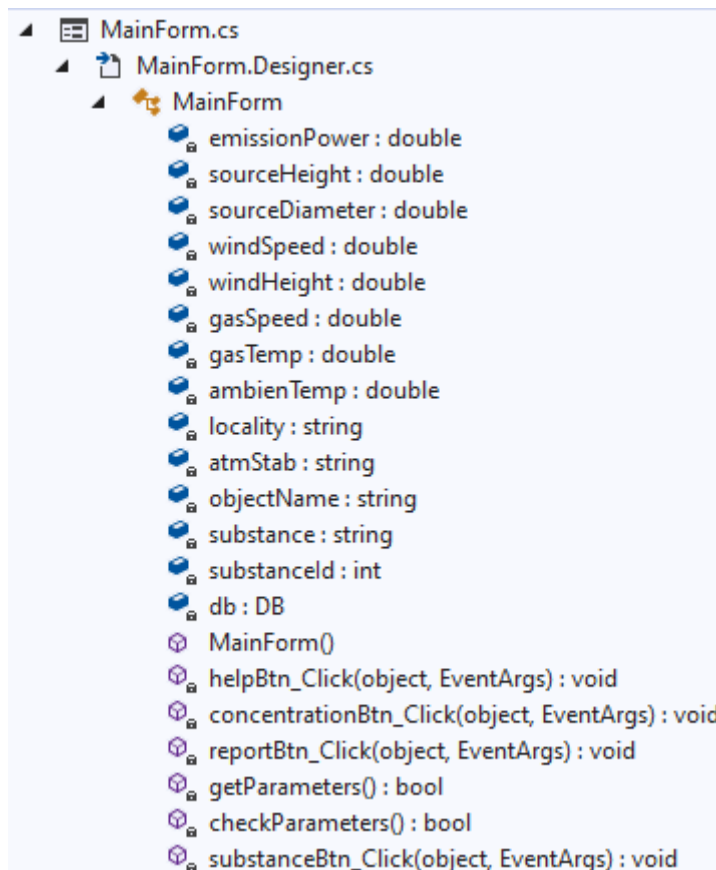


Рисунок 5.5 – Фрагмент структури класу «MainForm»

Клас «MainForm» складається з наступних функцій:

- helpBtn_Click – відбувається при натисканні кнопки «helpBtn» і викликає допоміжну форму «PasquillForm»;
- concentrationBtn_Click – відбувається при натисканні кнопки «concentrationBtn» і викликає форму розрахунку концентрації забруднюючої речовини «ConcentrationForm»;
- reportBtn_Click – відбувається при натисканні кнопки «reportBtn» і викликає форму розрахунку концентрації забруднюючої речовини «ReportForm»;
- substanceBtn_Click – відбувається при натисканні кнопки «substanceBtn» і викликає форму з списком речовин «SubstanceListForm»;
- getParameters – зчитує дані, які ввів користувач і в разі помилки генерує

- вікно з повідомленням про цю помилку;
- `checkParameters` – перевіряє чи коректні дані ввів користувач і в разі помилки генерує вікно з повідомленням про цю помилку.

5.4 Вікно розрахунку концентрації забруднюючої речовини

Вікно розрахунку концентрації забруднюючої речовини – форма, яка дозволяє користувачу на основі вводу координат, розрахувати концентрацію забруднюючої речовини в точці простору та отримати висновок, щодо її впливу на здоров'я людини. Користувач має можливість подивитися як саме необхідно обирати координати для розрахунку.

Фрагмент структури класу «`ConcentrationForm`» представлений на рисунку 5.6.

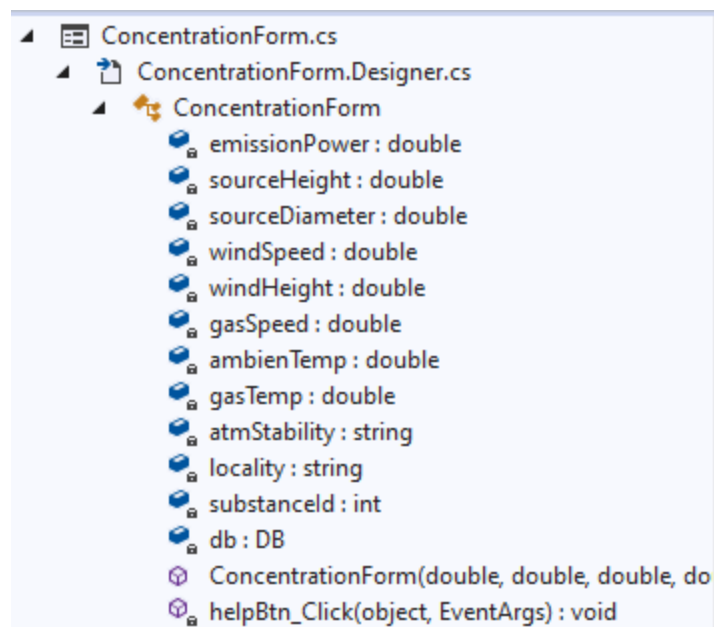


Рисунок 5.6 – Фрагмент структури класу «`ConcentrationForm`»

Клас «`ConcentrationForm`» складається з наступних функцій:

- ConcentrationForm – конструктор класу, який приймає в себе параметри, які описують забруднюючу речовину, джерело викиду та навколишнє середовище;
- helpBtn_Click – відбувається при натисканні кнопки «helpBtn» і викликає допоміжну форму «CoordinatesForm»;
- calcBtn_Click – відбувається при натисканні кнопки «calcBtn», проводить розрахунок концентрації та коефіцієнту небезпеки для життя людини, повертає ці значення та значення гранично допустимої концентрації і генерує висновок, щодо отриманих результатів.

5.5 Вікно генерації звіту

Вікно генерації звіту – це форма, в якій користувач має вказати інформацію щодо джерела викиду та забруднюючої речовини, а також заповнити дані для розрахункового прямокутника. Далі користувач отримує змогу згенерувати звіт, який відкривається в новій формі – форма звіту.

Фрагмент структури класу «ReportForm» представлений на рисунку 5.7.

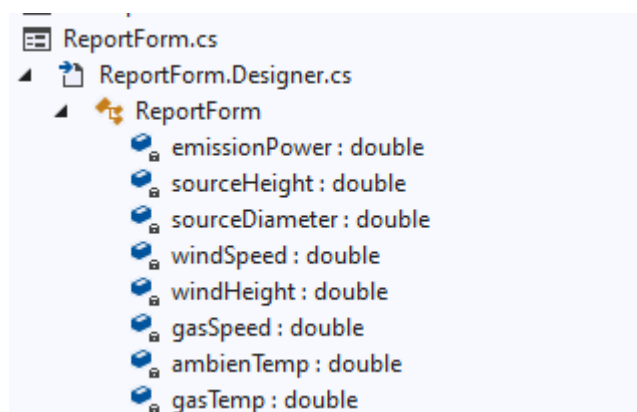


Рисунок 5.7 – Фрагмент класу «ReportForm»

Клас «ReportForm» складається з наступних функцій:

- ReportForm – конструктор класу, який приймає в себе параметри, які описують забруднюючу речовину, джерело викиду та навколишнє середовище;
- reportBtn_Click – відбувається при натисканні кнопки «reportBtn» і викликає форму «PDFreport».

Фрагмент структури класу «PDFreport» представлений на рисунку 5.8.

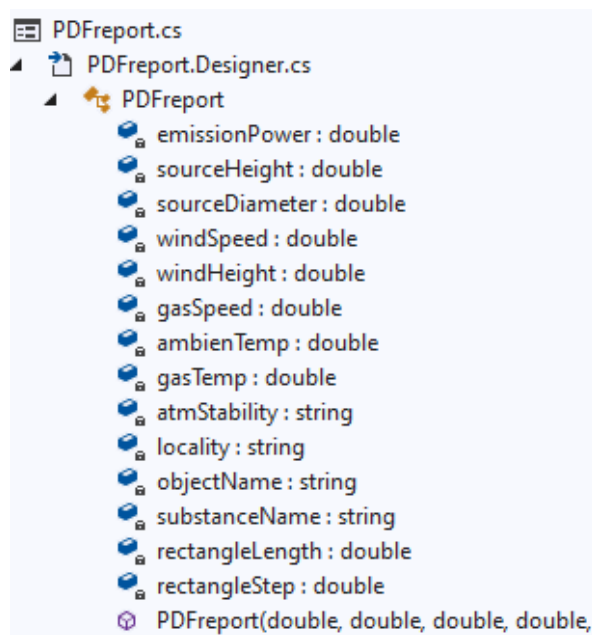


Рисунок 5.8 – Фрагмент структури класу «PDFreport»

Клас «PDFreport» складається з наступних функцій:

- PDFreport – конструктор класу, який приймає в себе параметри, які описують забруднюючу речовину, джерело викиду та навколишнє середовище;
- PDFreport_Load – генерує на основі отриманої інформації звіт в форматі PDF;
- FindAndReplase – шукає ключові слова в файлі, та замінює їх значення на необхідні параметри.

5.6 Допоміжні вікна

Допоміжні вікна – це форми, які містять в собі зображення з допоміжною інформацією для користувача. До таких вікон належать:

- допоміжна форма, яка дозволяє обрати речовину зі списку і описується класом «SubstanceListForm»;
- допоміжна форма з класами стабільності атмосфери, яка описується класом «PasquillForm»;
- допоміжна форма з інформацією про координати розрахунків, яка описується класом «CoordinatesForm».

Структура класу «PasquillForm» представлена на рисунку 5.9.

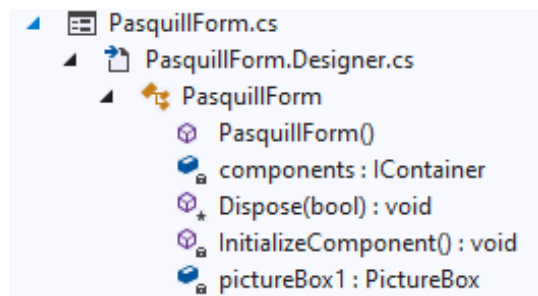


Рисунок 5.9 – Структура класу «PasquillForm»

Структура класу «CoordinatesForm» представлена на рисунку 5.10.

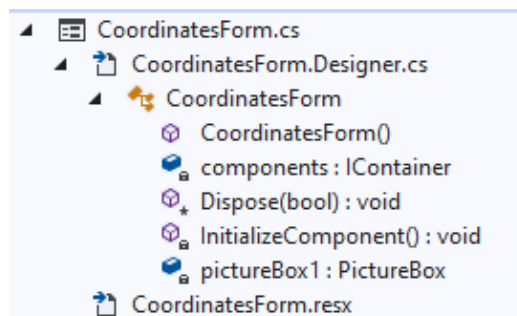


Рисунок 5.10 – Структура класу «CoordinatesForm»

Структура класу «SubstanceListForm» представлена на рисунку 5.11.

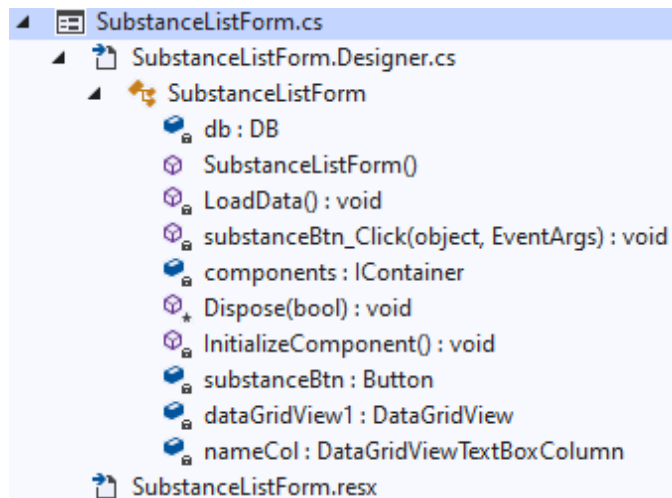


Рисунок 5.11 – Структура класу «ubstanceListForm»

Клас «PasquillForm» складається з наступних функцій:

- LoadData – отримує значення про речовини з БД та видає їх у формі таблиці користувачу;
- substanceBtn_Click – передає назву речовини в поле «substanceBox» головної форми та закриває поточне вікно.

5.7 Висновки до розділу

Було розроблено програмний продукт, який відповідає поставленій задачі. В даному розділі була описана структура програмного забезпечення, його база даних та основні модулі. Також для кращого розуміння було розроблено UML діаграми, а саме: діаграма прецедентів та діаграма діяльності.

6 МЕТОДИКА РОБОТИ КОРИСТУВАЧА

Розроблений додаток має три основні вікна для роботи з користувачем, а саме: головне вікно, вікно для розрахунку концентрації забруднюючої речовини та коефіцієнта небезпеки і вікно для побудови звіту.

6.1 Головна сторінка

При запуску додатку одразу з'являється головне вікно, основною задачею якого є збір інформації про джерело викиду, речовину і навколишнє середовище.

Вигляд головної сторінки користувача представлено на рисунку 6.1.

Гауссове моделювання повітряної дисперсії

Назва об'єкту:

Q - Потужність викиду, г/с v - Швидкість газу в гирлі, м/с

h - Висота джерела викиду, м ts - Температура газів, °C

d - Діаметр джерела викиду, м t - Температура навколишнього середовища, °C

u - Середня швидкість вітру, м/с Стабільність атмосфери

на висоті, м Місцевість

Речовина

Розрахунок концентрації Звіт

Рисунок 6.1 – Головна сторінка

Користувач має ввести дані та обрати варіанти подальшої роботи, а саме:

провести розрахунок концентрації забруднюючої речовини в точці простору, натиснувши кнопку «Розрахунок концентрації», або згенерувати звіт, натиснувши кнопку «Звіт».

Для полегшення роботи користувач може скористатися кнопкою «?» біля поля «Стабільність атмосфери», яка відкриває нове вікно «Класи стабільності атмосфери». У вікні представлена таблиця «Pasquill», яка допомагає правильно обрати клас стабільності атмосфери. Вигляд вікна «Класи стабільності атмосфери» представлений на рисунку 6.2.



Швидкість вітру	Денний час			Нічний час	
	Рівень сонячного освітлення			Хмарність	
	Сильне	Середнє	Слабке	> 50%	<= 50%
м/с					
< 2	A	A-B	B	E	F
[2 - 3)	A-B	B	C	E	F
[3 - 5)	B	B-C	C	D	E
[5 - 6)	C	C-D	D	D	D
>= 6	C	D	D	D	D

Клас	Стан	Типова швидкість вітру	Опис погоди
A	Максимально нестабільний	1	Безхмарно
B	Середньо нестабільний	2	Сонячно і тепло
C	Помірно нестабільний	5	Змінна хмарність протягом дня
D	Нейтральний	5	Хмарний день або ніч
E	Помірно стабільний	3	Змінна хмарність протягом ночі
F	Середньо стабільний	2	Ясна ніч

Рисунок 6.2 – Вікно «Класи стабільності атмосфери»

Стабільність атмосфери та тип місцевості користувач обирає з випадуючого списку. Вигляд випадуючого списку «Стабільність атмосфери» представлений на рисунку 6.3, а вигляд випадуючого списку «Місцевість» на рисунку 6.4.

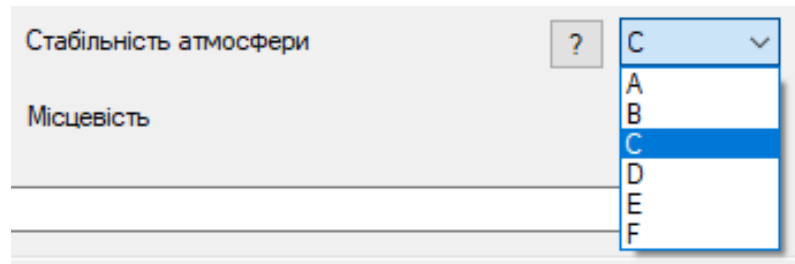


Рисунок 6.3 – Список «Стабільність атмосфери»

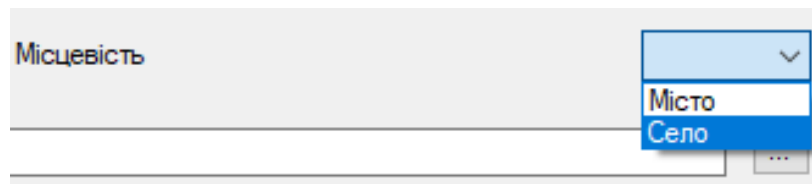


Рисунок 6.4 – Список «Місцевість»

Також користувач має ввести назву речовини, яка буде використовуватись для подальших дій. Він має два варіанти: ввести власноруч або обрати речовину з бази даних.

Натиснувши на кнопку «...», яка знаходиться навпроти поля «Речовина», відкриється нове вікно «Список речовин», яке демонструє список речовин, які знаходяться в БД. За допомогою кнопки «Обрати», користувач може обрати речовину, назва якої автоматично передається в поле «substanceBox» головного меню. Вигляд вікна «Список речовин» представлений на рисунку 6.5.

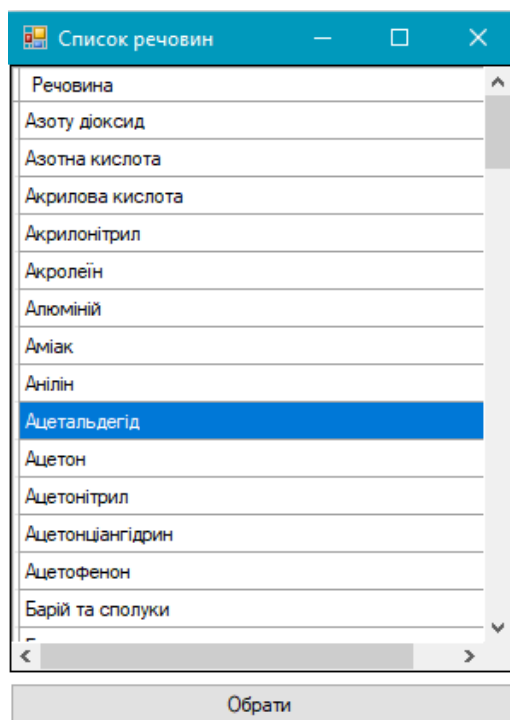


Рисунок 6.5 – Вікно «Список речовин»

Також додаток передбачає можливість перевірки правильності вводу даних. Якщо пропустити якесь поле або ввести дані неправильного типу, то додаток виведе повідомлення з попередженням, яке представлено на рисунку 6.6.

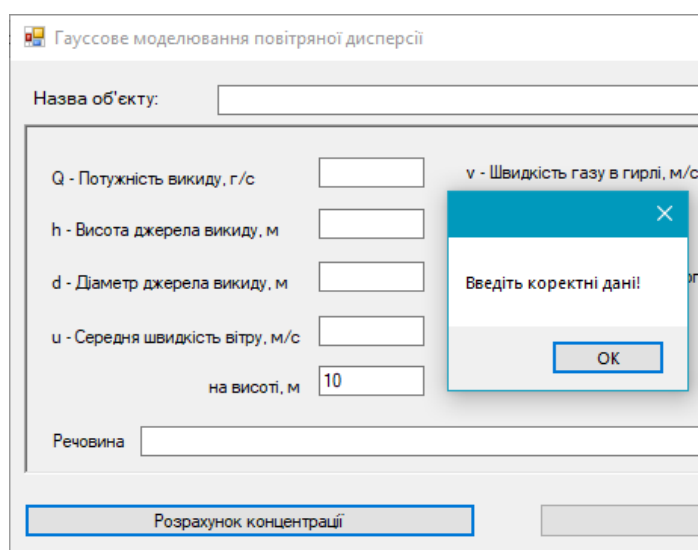


Рисунок 6.6 – Повідомленням про помилку, при пропуску поля

Якщо ввести дані правильного типу, але не в коректному діапазоні, також будуть з'являтися повідомлення про помилку стосовно кожного поля. Приклад повідомлення при вводі від'ємної потужності викиду представлена на рисунку 6.7.

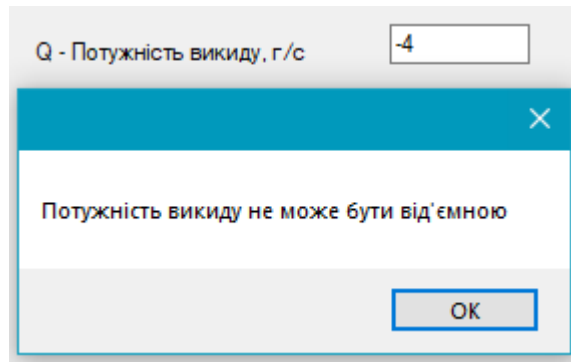


Рисунок 6.7 – Помилка при вводі від'ємної потужності

На рисунку 6.8 показано початкове вікно з правильно заповненими даними.

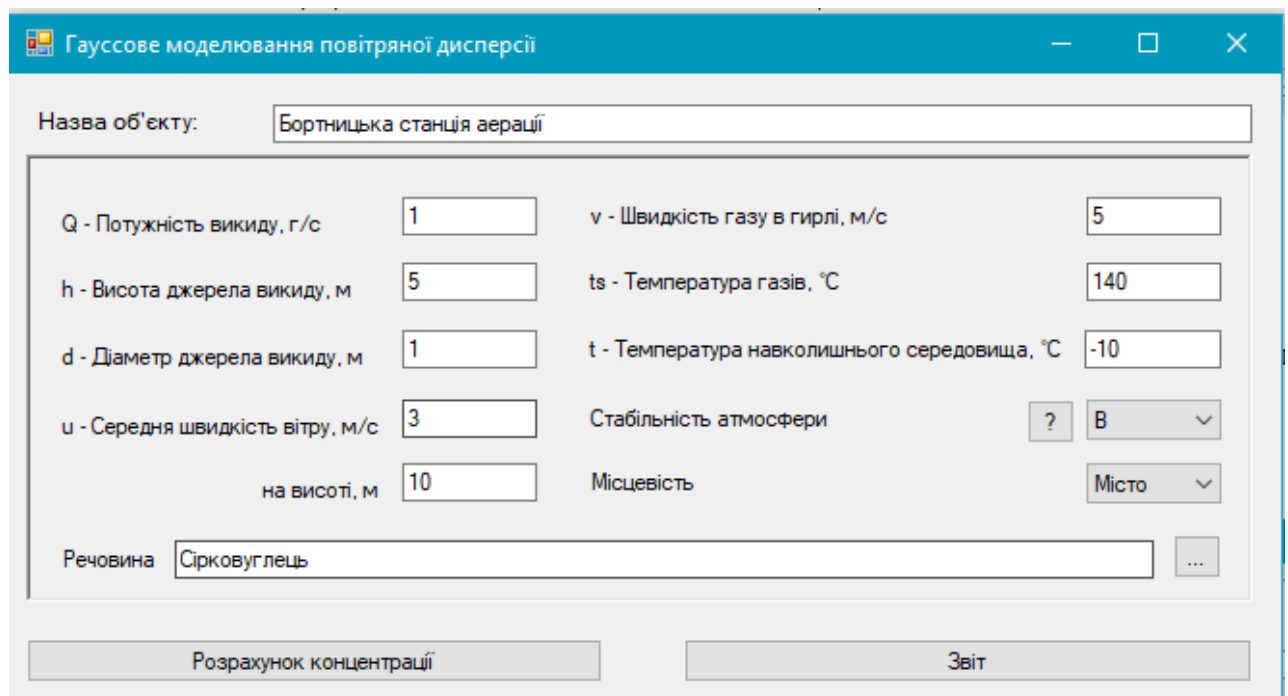
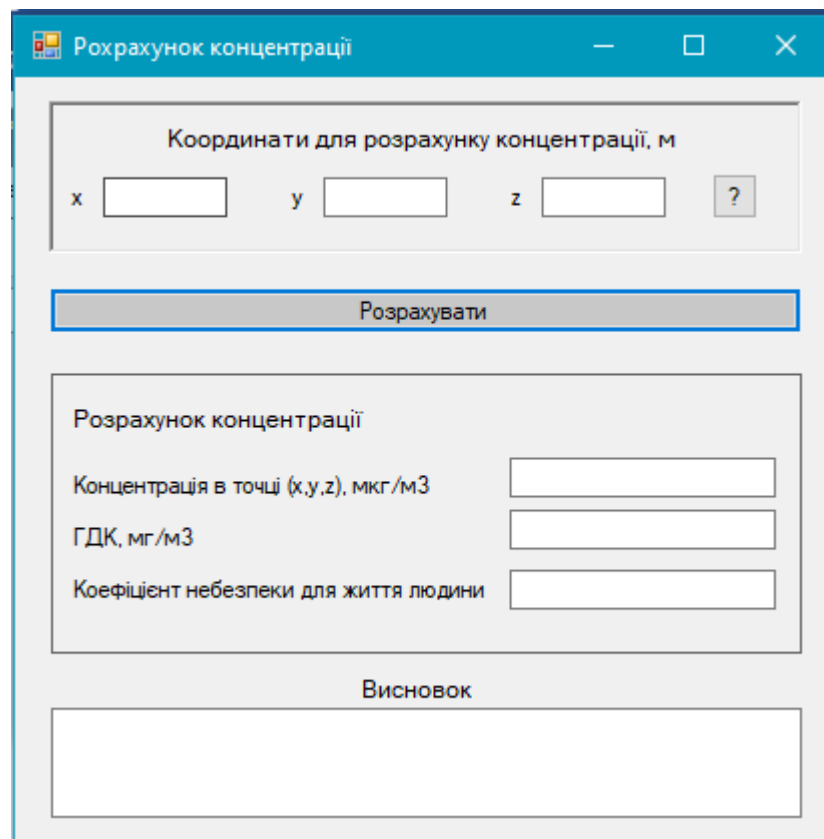


Рисунок 6.8 – Заповнене початкове вікно

Таким чином після коректного заповнення головного вікна, додаток дає можливість обрати наступну дію та продовжити роботу.

6.2 Розрахунок концентрації забруднюючої речовини

При натисканні кнопки «Розрахунок концентрації» в головному вікні, відкривається вікно «Розрахунок концентрації», яке представлено на рисунку 6.9.



The image shows a Windows-style dialog box titled "Розрахунок концентрації". It contains the following elements:

- A section titled "Координати для розрахунку концентрації, м" with three input fields for x, y, and z, and a question mark button.
- A "Розрахувати" button.
- A section titled "Розрахунок концентрації" with three output fields: "Концентрація в точці (x,y,z), мкг/м3", "ГДК, мг/м3", and "Коефіцієнт небезпеки для життя людини".
- A "Висновок" section with a large empty text area.

Рисунок 6.9 – Вікно «Розрахунок концентрації»

Користувач має ввести координати точки простору, в якій буде розраховуватись концентрація речовини. Для більшої зрозумілості як правильно обрати координати, користувач може скористатися кнопкою «?», при натисканні

якої з'являється вікно «Координати», яке показано на рисунку 6.10.

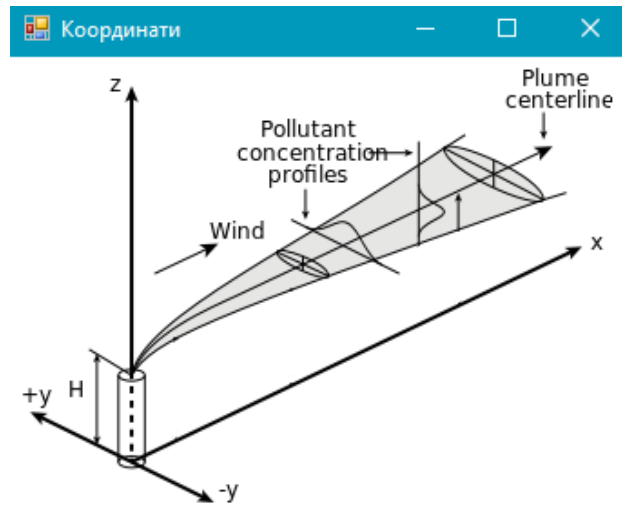


Рисунок 6.10 – Вікно «Координати»

Приклад результату розрахунку можна побачити на рисунку 6.11.

Координати для розрахунку концентрації, м

x	1000	y	3000	z	0	?
---	------	---	------	---	---	---

Розрахувати

Розрахунок концентрації 'Бутилацетат':

Концентрація в точці (x,y,z), мкг/м ³	1,68518452438469E-05
ГДК, мг/м ³	0,1
Коефіцієнт небезпеки для життя людини	0,024074064634067

Висновок

Концентрація речовини в точці (1000; 3000; 0) - не перевищує гранично допустиму концентрацію. Неканцерогенний ризик впливу на здоров'я людини - малий

Рисунок 6.11 – Результат розрахунку концентрації

Щоб отримати результат розрахунку концентрації забруднюючої речовини в точці простору необхідно натиснути кнопку «Розрахувати». Окрім значення концентрації, користувач може побачити гранично допустиму концентрацію обраної речовини, коефіцієнт небезпеки для життя людини і висновок щодо впливу речовини на здоров'я людини.

Також користувач може ввести речовину, якої нема в базі даних, приклад розрахунку в цьому випадку представлений на рисунку 6.12.

Рисунок 6.12 – Розрахунок концентрації речовини, якої немає в БД

Додаток все одно проводить розрахунок концентрації речовин в точці простору, але вказує користувачу, що значення ГДК та коефіцієнту небезпеки не можуть бути розраховані, через відсутність необхідних параметрів.

6.3 Побудова звіту

При натисканні кнопки «Звіт» в головному вікні, відкривається вікно «Отримати звіт», яке представлено на рисунку 6.13.

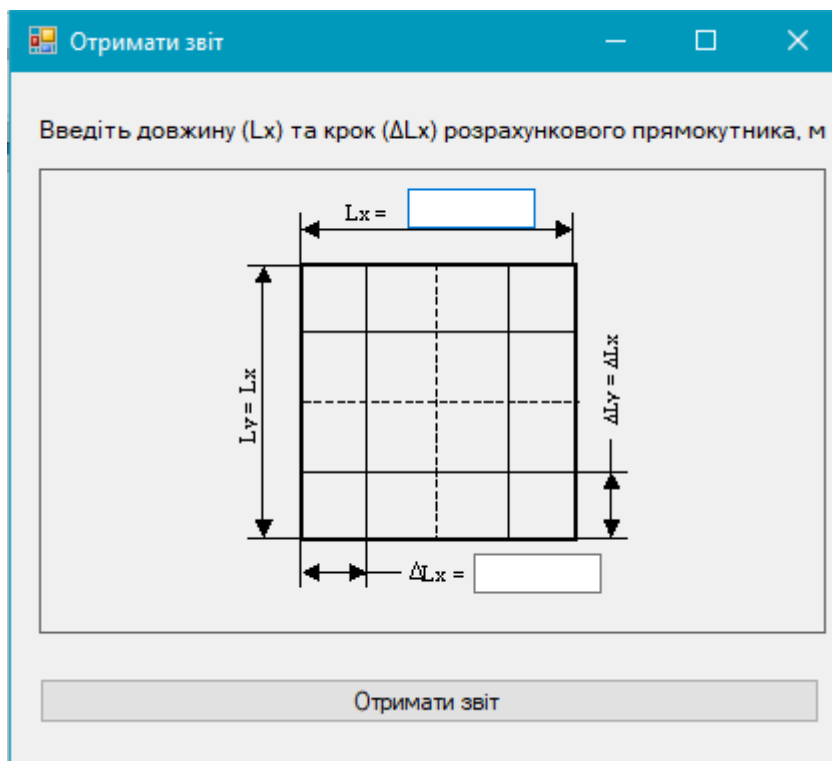


Рисунок 6.13 – Вікно «Отримати звіт»

Звіт генерується на основі параметрів розрахункового прямокутника, тобто використовує його периметр для розрахунку концентрації шкідливої речовини в межах цього прямокутника.

Додаток передбачає можливість введення помилкових даних. Наприклад при вводі довжини прямокутника, яка є меншою за крок розрахунку, з'явиться повідомлення про помилку, яке можна побачити на рисунку 6.14, а при вводі від'ємного значення – з'явиться повідомлення про помилку, яке можна побачити на рисунку 6.15.

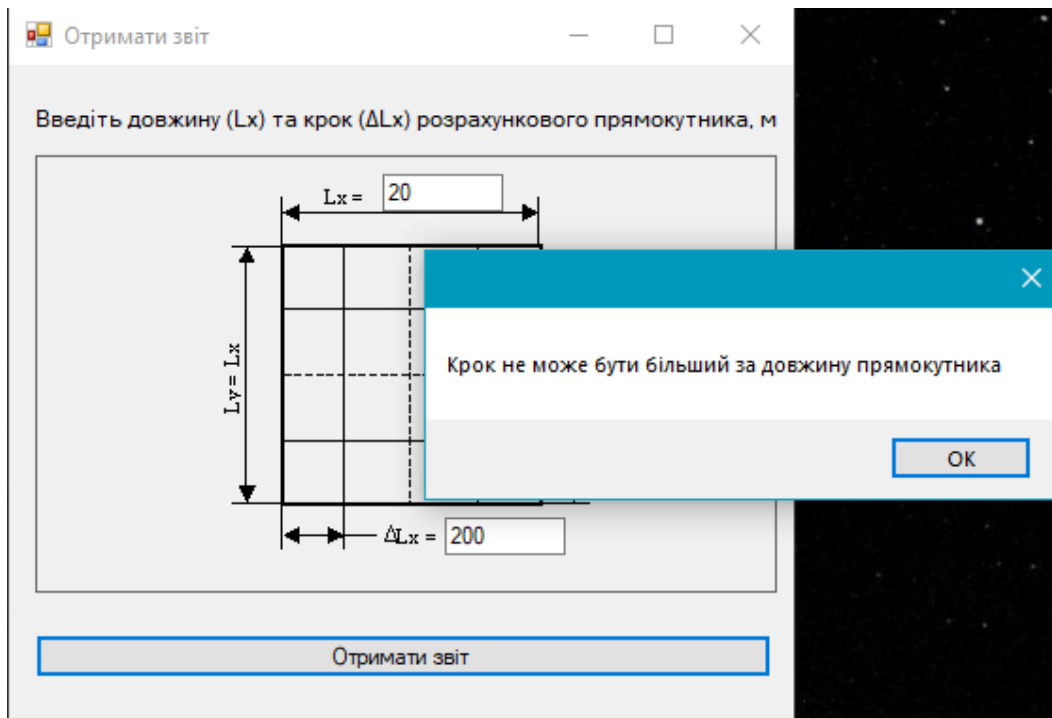


Рисунок 6.14 – Повідомлення про помилку «Крок не може бути більший за довжину прямокутника»

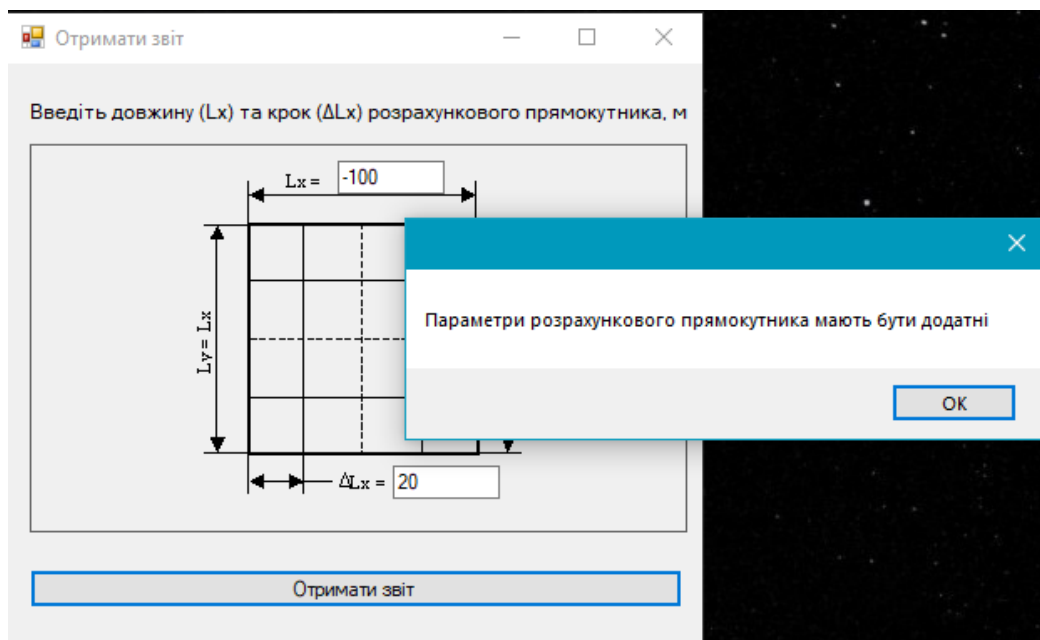


Рисунок 6.15 – Повідомлення про помилку «Параметри розрахункового прямокутника мають бути додатні»

Приклад правильно заповненого вікна «Отримати звіт» представлено на рисунку 6.16.

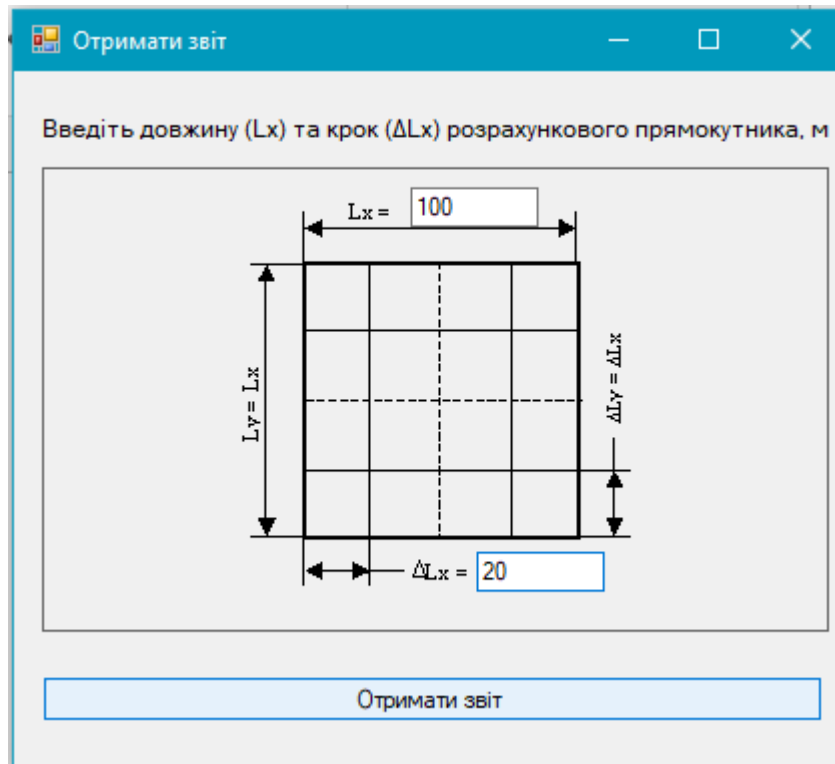


Рисунок 6.16 – Заповнене вікно «Отримати звіт»

При натисканні кнопки «Отримати звіт», відкривається нове вікно «Звіт», в якому користувач отримує звіт на основі даних, які він ввів. На рисунку 6.17 можна побачити, що звіт має інформацію про вхідні дані, а саме: назва об'єкту розрахунку, назва шкідливої речовини, потужність викиду, висота джерела викиду, діаметр джерела викиду, швидкість газоподібної суміші в гирлі джерела викиду, температура газоподібної суміші, температура навколишнього повітряного середовища, середня швидкість вітру та висота, на якій проводились виміри швидкості вітру, клас стабільності атмосфери та тип місцевості, а також значення розрахункового прямокутника.

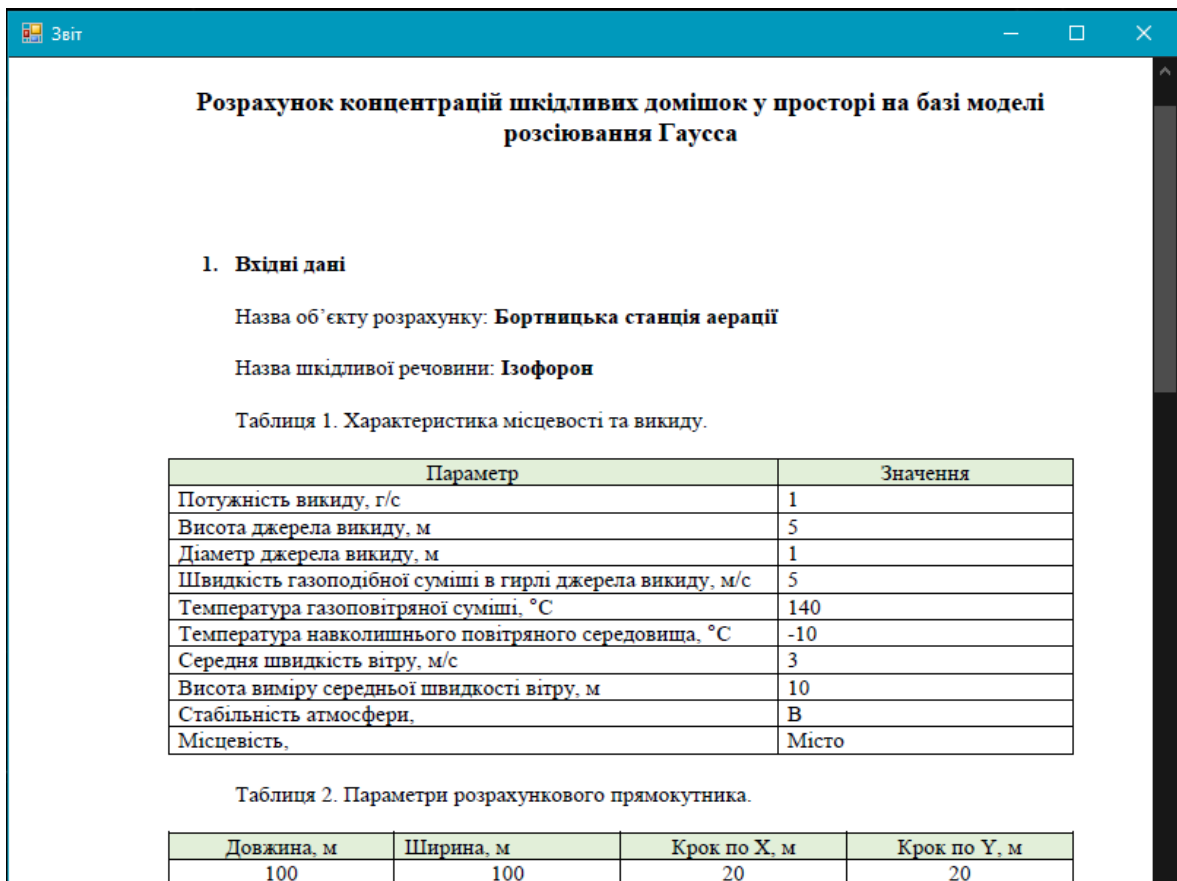


Рисунок 6.17 – Вхідні дані в звіті

На рисунку 6.18 можна побачити фрагмент результатів розрахунків.

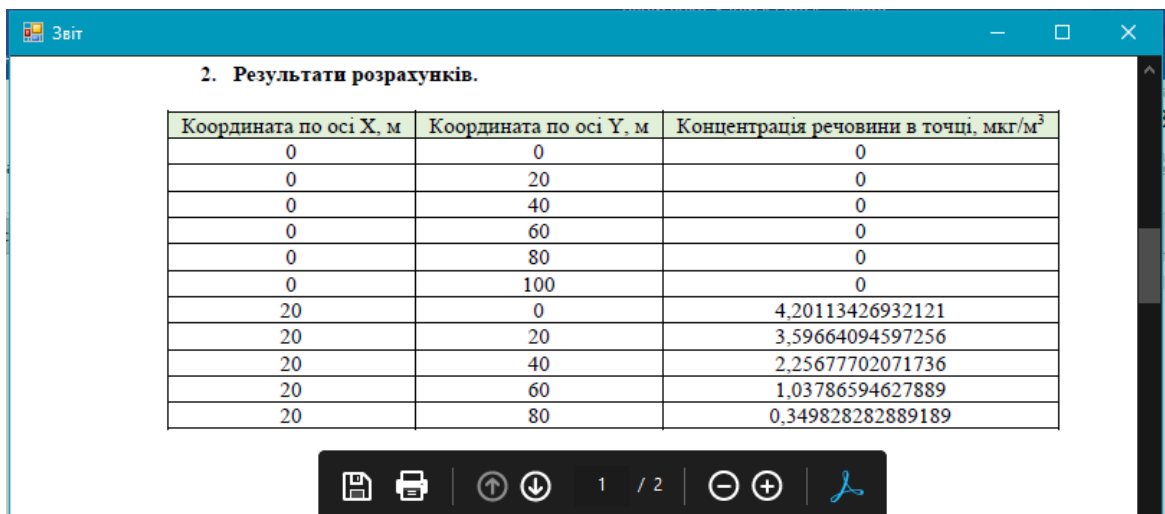


Рисунок 6.18 – Фрагмент результатів розрахунків

Звіт формується в форматі PDF. Користувач має можливість переглянути звіт, зберегти і роздрукувати його.

6.4 Висновки до розділу

В даному розділі було детально описано як користувач має та може взаємодіяти з системою. Були наведені приклади роботи та описаний алгоритм взаємодії користувача та системи. Система має зрозумілий інтерфейс, що сприяє легкості в роботі з даним програмний забезпеченням.

Основними користувачами системи є експерти – екологи, які працюють з підприємствами та установами пов'язаними з викидами небезпечних речовин в атмосферне повітря.

ВИСНОВКИ

В ході роботи була проаналізована предметна область та поставлені задачі до програмного продукту.

Для ефективного проектування програмного забезпечення був проведений огляд подібних систем, які вже були розроблені раніше.

На основі цього аналізу були обрані технології розробки, які максимально ефективно допомагають вирішувати поставлені задачі, а також обрана математична модель для будування повітряної дисперсії.

У цій роботі була обґрунтована актуальність розробки додатку для моделювання шлейфу повітряної дисперсії забруднюючих речовин в атмосферному повітрі від різноманітних підприємств при певних погодних умовах.

Були спроектовані та реалізовані функції для розрахунку концентрації забруднюючих речовин, аналізу впливу речовини на життя людини та моделювання звіту на основі отриманої інформації.

Функціонал програмного забезпечення і варіанти його використання були протестовані.

В майбутньому розробка додатка може бути розширена, а саме:

1. Додати систему адміністрування.
2. Створити клієнт – серверну архітектуру.
3. Додати базу даних для зберігання інформації про джерела забруднення.
4. Додати функціонал для роботи користувача з БД.
5. Додати нові функції для вирішення інших завдань моделювання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. About – git: web–site. URL: <https://git-scm.com/about/info-assurance>: 14.04.2022).
2. COM Interop: веб–сайт. URL: https://uk.wikijaa.ru/wiki/COM_Interop (дата звернення: 16.04.2022).
3. Desktop Guide (Windows Forms .NET): web–site. URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-6.0> (дата звернення: 12.04.2022).
4. Hejlsberg A., Torgersen M., Wiltamuth S., Golde P. The C# Programing Language: tutorial, New–Jersey: 2010. 1085 p.
5. Structured Query Language (SQL): web–site. URL: <https://www.techtarget.com/searchdatamanagement/definition/SQL> (дата звернення: 18.04.2022).
6. System.Reflection Namespace: web–site. URL: <https://docs.microsoft.com/en-us/dotnet/api/system.reflection?view=net-6.0>: 12.04.2022).
7. Visual Studio: web-site. URL: <https://visualstudio.microsoft.com/> (дата звернення: 13.04.2022).
8. What is MySQL?: web–site. URL: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>: 13.04.2022).
9. Why choose .NET?: web–site. URL: <https://dotnet.microsoft.com/en-us/platform/why-choose-dotnet> (дата звернення: 20.04.2022).
10. Why MySQL?: web–site. URL: <https://www.mysql.com/why-mysql/> (дата звернення: 15.04.2022).
11. Бекетов В.Е. Конспект лекций по разделу «Рассеивание загрязняющих веществ в атмосферном воздухе и методики расчета приземных концентраций» к дисциплине «Инженерная аэроэкология городов». Харьков: ХНАГХ, 2011. 74 с. URL:

http://eprints.kname.edu.ua/21234/1/%D0%A1%D0%AE_%D0%BF%D0%B5%D1%87_%D0%B2%D0%B0%D1%80_%D0%BF%D0%BB_2009_35%D0%9B_24_09_2010_%D0%A0%D0%90%D0%A1%D0%A1%D0%95%D0%98%D0%92%D0%90%D0%9D%D0%98%D0%95_5_%D0%9A_%D0%9B%D0%B5%D0%BA%D1%86%D0%B8%D0%B8_%D0%BE%D1%81%D0%B5%D0%BD%D0%BD%D0%B8%D0%B9_%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80_%D0%B7%D0%B0%D0%BC.pdf (дата звернення: 10.04.2022).

12. Грицюк Ю.І., Грицюк М.Ю. Моделі оцінювання технологічних чинників під час виникнення пожеж на сховищах нафтопродуктів. Львів: РВВ НЛТУ України, 2016. 12 с. URL: <http://fasu.nltu.edu.ua/index.php/nplanu/article/download/69/39/> (дата звернення: 15.04.2022).
13. Караєва Н.В. Характеристика можливостей комп'ютерних систем і програмних засобів для економіка–екологічного аналізу господарської діяльності: стаття. Київ: НТУУ КПІ, 2016. 5 с. URL: <http://global-national.in.ua/archive/14-2016/208.pdf> (дата звернення: 11.04.2022).
14. Кирсанов Ю.Г. Оценка взаимодействия выбросов вредных веществ на атмосферный воздух. Екатеринбург: Издательство Уральского Университета, 2018. 116 с. URL: <http://fasu.nltu.edu.ua/index.php/nplanu/article/download/69/39/> (дата звернення: 15.04.2022).
15. ОНД - 86: веб-сайт. URL: <https://docs.cntd.ru/document/1200000112> (дата звернення: 16.04.2022).
16. Програма «Атмосфера – ПДВ»: веб-сайт. URL: http://logosoft.ru/programmy/atmosfera/atmosfera_pdv.html (дата звернення: 21.04.2022).

17. Программный комплекс «Призма–предприятие»: веб–сайт. URL: <https://allsoft.ru/software/vendors/logus/programmnyu-kompleks--prizma---predpriyatie-/> (дата звернения: 21.04.2022).
18. Программный комплекс «НОСТРАДАМУС»: веб–сайт. URL: <http://ibrae.ac.ru/contents/950/> (дата звернения: 21.04.2022).
19. Программный комплекс «ЭРА» : веб–сайт. URL: <https://lpp.ru/about/o-nas/> (дата звернения: 21.04.2022).
20. ЭПК РОСА. Воздух: веб–сайт. URL: <https://ecolida.ru/epk-rosa/vozduh> (дата звернения: 21.04.2022).

ДОДАТОК А

Моделювання гауссового шлейфу повітряної дисперсії

Специфікація

УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТМ-81253_22Б

Аркушів 3

Київ 2022

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕП С_ТМ-81253_22Б 81-1	Записка.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕП С_ТМ-81253_22Б 12-1	DB.cs	Компонент для роботи з БД
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕП С_ТМ-81253_22Б 12-2	Dispersion.cs	Компонент для роботи з викидами
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕП С_ТМ-81253_22Б 12-3	HealthRisk.cs	Компонент для розрахунку ризиків для здоров'я
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕП С_ТМ-81253_22Б 12-4	ConcentrationForm.cs	Компонент для розрахунку концентрації речовини

УКР.НТУУ"КПІ імені Ігоря Сікорського" _ТЕФ_АПЕП С_ТМ-81253_22Б 12-5	CoordinatesForm.cs	Компонент для роботи з координатами
УКР.НТУУ"КПІ імені Ігоря Сікорського" _ТЕФ_АПЕП С_ТМ-81253_22Б 12-6	MainForm.cs	Компонент для збору інформації
УКР.НТУУ"КПІ імені Ігоря Сікорського" _ТЕФ_АПЕП С_ТМ-81253_22Б 12-7	PDFreport.cs	Компонент для роботи з звітом
УКР.НТУУ"КПІ імені Ігоря Сікорського" _ТЕФ_АПЕП С_ТМ-81253_22Б 12-8	ReportForm.cs	Компонент для генерації звіту

ДОДАТОК Б

Моделювання гауссового шлейфу повітряної дисперсії

Лістинг програми

УКР.НТУУ"КПІ імені Ігоря Сікорського" _ТЕФ_АПЕПС_ТМ-81253_22Б 12-1

Аркушів 15

Київ 2022

Dispersion.cs:

```
using MySql.Data.MySqlClient;
using System;

namespace GaussDispersionModel
{
    class Dispersion
    {
        private double emissionPower; // Потужність викиду
        private double sourceHeight; // Висота джерела викину
        private double sourceDiameter; // Діаметр джерела викиду
        private double windSpeed; // Середня швидкість вітру
        private double windHeight; // на висоті
        private double gasSpeed; // Швидкість газу в гирлі
        private double gasTemp; // Температура газу
        private double ambienTemp; // Тампература навколишньої середи
        private int atmStability; // Стабільність атмосфери
        private int locality; // Місцеість

        DB db = new DB();

        public Dispersion(double emissionPower, double sourceHeight, double sourceDiameter,
            double windSpeed,
            double windHeight, double gasSpeed, double gasTemp, double ambienTemp, String
            atmStability, String locality)
        {
            this.emissionPower = emissionPower;
            this.sourceHeight = sourceHeight;
            this.sourceDiameter = sourceDiameter;
            this.windSpeed = windSpeed;
            this.windHeight = windHeight;
            this.gasSpeed = gasSpeed;
            this.gasTemp = gasTemp;
            this.ambienTemp = ambienTemp;

            MySqlCommand command = new MySqlCommand("call atm_stab (@atmStability)",
            db.getConnection());
            command.Parameters.Add("@atmStability", MySqlDbType.String).Value =
            atmStability[0];

            db.openConnection();
            this.atmStability = Convert.ToInt32(command.ExecuteScalar());
            db.closeConnection();

            if (locality[0] == 'M') { this.locality = 2; }
            else { this.locality = 1; }
        }

        // Значення коефіцієнта p
        private double p()
        {
            MySqlCommand command = new MySqlCommand("call p (@atmStability, @locality)",
            db.getConnection());
            command.Parameters.Add("@atmStability", MySqlDbType.Int32).Value = atmStability;
            command.Parameters.Add("@locality", MySqlDbType.Int32).Value = locality;

            db.openConnection();
            double p = Convert.ToDouble(command.ExecuteScalar());
        }
    }
}
```

```

        db.closeConnection();

        return p;
    }

    // Значення коефіцієнта c
    private double c()
    {
        MySqlCommand command = new MySqlCommand("call c (@atmStability)",
db.getConnection());
        command.Parameters.Add("@atmStability", MySqlDbType.Int32).Value = atmStability;

        db.openConnection();
        double c = Convert.ToDouble(command.ExecuteScalar());
        db.closeConnection();

        return c;
    }

    // Значення коефіцієнта d
    private double d()
    {
        MySqlCommand command = new MySqlCommand("call d (@atmStability)",
db.getConnection());
        command.Parameters.Add("@atmStability", MySqlDbType.Int32).Value = atmStability;

        db.openConnection();
        double d = Convert.ToDouble(command.ExecuteScalar());
        db.closeConnection();

        return d;
    }

    // Значення коефіцієнта a
    private double a(double x)
    {
        MySqlCommand command = new MySqlCommand("call a (@atmStability, @x)",
db.getConnection());
        command.Parameters.Add("@atmStability", MySqlDbType.Int32).Value = atmStability;
        command.Parameters.Add("@x", MySqlDbType.Int32).Value = x;

        db.openConnection();
        double a = Convert.ToDouble(command.ExecuteScalar());
        db.closeConnection();

        return a;
    }

    // Значення коефіцієнта b
    private double b(double x)
    {
        MySqlCommand command = new MySqlCommand("call b (@atmStability, @x)",
db.getConnection());
        command.Parameters.Add("@atmStability", MySqlDbType.Int32).Value = atmStability;
        command.Parameters.Add("@x", MySqlDbType.Int32).Value = x;

        db.openConnection();
        double b = Convert.ToDouble(command.ExecuteScalar());
        db.closeConnection();

        return b;
    }
}

```

```

// Швидкість вітру на ефективній висоті
private double us()
{
    double p = this.p(); ;
    double zref = windHeight;
    double hs = sourceHeight;
    double uref = windSpeed;

    return uref * Math.Pow((hs / zref), p);
}

// Ефективна висота джерела викиду
private double he(double x)
{
    double vs = gasSpeed;
    double ds = sourceDiameter;
    double ts = gasTemp;
    double ta = ambientTemp;
    double hs = sourceHeight;
    double atmStab = atmStability;
    double us = this.us();
    double xf, he;
    double hs1;

    // Параметр Брига
    double fb = 9.8 * vs * Math.Pow(ds, 2) * ((ts - ta) / (4 * ts));

    // Розрахунок модифікованої висоти джерела
    if (vs < 1.5 * us)
    {
        hs1 = hs + 2 * ds * ((vs / us) - 1.5);
    }
    else { hs1 = hs; }

    // Відстань з максимальною концентрацією
    if (fb < 55) { xf = 49 * Math.Pow(fb, 0.625); }
    else { xf = 119 * Math.Pow(fb, 0.4); }

    // Розрахунок ефективної висоти
    if (atmStab == 1 || atmStab == 2 || atmStab == 3 || atmStab == 4) // Якщо
стабільність атм = A/B/C/D
    {
        if (x < xf) { he = hs1 + 1.6 * (Math.Pow(fb, 0.33333) * Math.Pow(xf, 0.33333)
/ us); }
        else { he = hs1 + 1.6 * (Math.Pow(fb, 0.33333) * Math.Pow(x, 0.33333) / us); }
    }
    else
    { // Якщо стабільність атм = E/F
    double s;

    // Коефіцієнт стабільності
    if (atmStab == 5) { s = 9.8 * 0.02 / ta; }
    else { s = 9.8 * 0.035 / ta; }

    if (1.84 * us * Math.Pow(s, -0.5) >= xf)
    {
        if (x < xf) { he = hs1 + 1.6 * (Math.Pow(fb, 0.33333) * Math.Pow(xf,
0.33333) / us); }
        else { he = hs1 + 1.6 * (Math.Pow(fb, 0.33333) * Math.Pow(x, 0.33333) /
us); }
    }
    }
}

```

```

        else if (1.84 * us * Math.Pow(s, -0.5) >= x) { he = hs1 + 2.4 * Math.Pow(fb /
(us * s), 0.33333); }
        else { he = hs1 + 1.6 * (Math.Pow(fb, 0.33333) * Math.Pow(x, 0.33333) / us); }
    }
    return he;
}

//Розрахунок cy
private double cy(double x)
{
    return 456.11628 * x * Math.Tan(0.017453293 * (c() - d() * Math.Log(x)) * Math.PI
/ 180);
}

//Розрахунок cz
private double cz(double x)
{
    double cz;
    cz = a(x) * Math.Pow(x, b(x));

    if ((cz > 5000 && (atmStability == 1 || atmStability == 2 || atmStability == 3))
|| (atmStability == 1 && x > 3.11))
    {
        cz = 5000;
    }
    return cz;
}

//Розрахунок вертикальних умов розсіювання
private double v(double x, double z)
{
    double v, h1, h2, h3, h4, l, sum = 0;
    double u = windSpeed;
    double he = this.he(x);

    l = 320 * u;

    if (atmStability == 5 || atmStability == 6)
    {
        v = Math.Exp(-0.5 * Math.Pow(z - he, 2) / Math.Pow(cz(x), 2)) + Math.Exp(-0.5
* Math.Pow(z + he, 2) / Math.Pow(cz(x), 2));
    }
    else
    {
        for (int m = 1; m <= 3; m++)
        {
            h1 = z - (2 * m * l - he);
            h2 = z + (2 * m * l - he);
            h3 = z - (2 * m * l + he);
            h4 = z + (2 * m * l + he);
            sum += Math.Exp(-0.5 * Math.Pow(h1, 2) / Math.Pow(cz(x), 2))
+ Math.Exp(-0.5 * Math.Pow(h2, 2) / Math.Pow(cz(x), 2))
+ Math.Exp(-0.5 * Math.Pow(h3, 2) / Math.Pow(cz(x), 2))
+ Math.Exp(-0.5 * Math.Pow(h4, 2) / Math.Pow(cz(x), 2));
        }

        v = Math.Exp(-0.5 * Math.Pow(z - he, 2) / Math.Pow(cz(x), 2)) + Math.Exp(-0.5
* Math.Pow(z + he, 2) / Math.Pow(cz(x), 2)) + sum;
    }
    return v;
}
}

```

```

public double dotConcentration(double x, double y, double z)
{
    double concentration;
    double q = emissionPower;
    if (x < 0.6)
    {
        return 0;
    }
    else
    {
        concentration = (q * Math.Pow(10, 6) * v(x, z) * Math.Exp(-0.5 * Math.Pow(y,
2) / Math.Pow(cy(x), 2))) / (2 * Math.PI * us() * cy(x) * cz(x));
        return concentration;
    }
}
}
}
}

```

DB.cs:

```

using MySql.Data.MySqlClient;

namespace GaussDispersionModel
{
    class DB
    {
        // СТрока підключення
        MySqlConnection connection = new MySqlConnection("server = localhost; port = 3307; " +
"username = root; password = root; "
+
"database = gauss_model");

        // Відкрити з'єднання
        public void openConnection()
        {
            if (connection.State == System.Data.ConnectionState.Closed)
            {
                connection.Open();
            }
        }

        // Закрити з'єднання
        public void closeConnection()
        {
            if (connection.State == System.Data.ConnectionState.Open)
            {
                connection.Close();
            }
        }

        public MySqlConnection getConnection()
        {
            return connection;
        }
    }
}

```

PDFreport.cs:

```
using System;
using System.Reflection;
using System.Windows.Forms;
using Microsoft.Office.Interop.Word;

namespace GaussDispersionModel.Forms
{
    public partial class PDFreport : Form
    {
        private readonly double emissionPower;
        private readonly double sourceHeight;
        private readonly double sourceDiameter;
        private readonly double windSpeed;
        private readonly double windHeight;
        private readonly double gasSpeed;
        private readonly double ambienTemp;
        private readonly double gasTemp;
        private readonly string atmStability;
        private readonly string locality;

        private readonly string objectName;
        private readonly string substanceName;
        private readonly double rectangleLength;
        private readonly double rectangleStep;

        public PDFreport(double emissionPower, double sourceHeight, double sourceDiameter,
            double windSpeed, double windHeight, double gasSpeed, double gasTemp, double
            ambienTemp,
            string atmStability, string locality, string objectName, string substanceName,
            double rectangleLength, double rectangleStep)
        {
            InitializeComponent();
            this.emissionPower = emissionPower;
            this.sourceHeight = sourceHeight;
            this.sourceDiameter = sourceDiameter;
            this.windSpeed = windSpeed;
            this.windHeight = windHeight;
            this.gasSpeed = gasSpeed;
            this.gasTemp = gasTemp;
            this.ambienTemp = ambienTemp;
            this.atmStability = atmStability;
            this.locality = locality;

            this.objectName = objectName;
            this.substanceName = substanceName;
            this.rectangleLength = rectangleLength;
            this.rectangleStep = rectangleStep;
        }

        Microsoft.Office.Interop.Word.Application app;
        Document doc;
        object objMiss = Missing.Value;
        object TmpFile = System.IO.Path.GetTempPath() + "report.pdf";
        object FileLocation =
            @"D:\Диплом\GaussDispersionModel\GaussDispersionModel\report.docx";

        private void PDFreport_Load(object sender, EventArgs e)
        {
            try
```

```

{
    app = new Microsoft.Office.Interop.Word.Application();
    doc = app.Documents.Open(ref FileLocation, ref objMiss, ref objMiss,
        ref objMiss, ref objMiss, ref objMiss, ref objMiss, ref objMiss,
        ref objMiss, ref objMiss, ref objMiss, ref objMiss, ref objMiss,
        ref objMiss, ref objMiss, ref objMiss);

    // Підставляємо значення в файл
    FindAndReplace("<objName>", objectName);
    FindAndReplace("<substanceName>", substanceName);
    FindAndReplace("<emissionPower>", emissionPower);
    FindAndReplace("<sourceHeight>", sourceHeight);
    FindAndReplace("<sourceDiameter>", sourceDiameter);
    FindAndReplace("<gasSpeed>", gasSpeed);
    FindAndReplace("<gasTemp>", gasTemp);
    FindAndReplace("<ambienTemp>", ambienTemp);
    FindAndReplace("<windSpeed>", windSpeed);
    FindAndReplace("<windHeight>", windHeight);
    FindAndReplace("<atmStability>", atmStability);
    FindAndReplace("<locality>", locality);
    FindAndReplace("<rectangleLength>", rectangleLength);
    FindAndReplace("<rectangleStep>", rectangleStep);

    // Додаємо таблицю в звіт
    Table tab = doc.Tables[3];

    int i = 2;
    double c;
    double z = 0;

    Dispersion release = new Dispersion(emissionPower, sourceHeight,
sourceDiameter,
        windSpeed, windHeight, gasSpeed, gasTemp, ambienTemp, atmStability,
locality);

    for (double x = 0; x <= rectangleLength; x += rectangleStep)
    {
        for (double y = 0; y <= rectangleLength; y += rectangleStep)
        {
            if (i != 2)
            {
                tab.Rows.Add(ref objMiss);
            }
            c = release.dotConcentration(x, y, z);

            tab.Cell(i, 1).Range.Text = Convert.ToString(x);
            tab.Cell(i, 2).Range.Text = Convert.ToString(y);
            tab.Cell(i, 3).Range.Text = Convert.ToString(c);

            i++;
        }
    }

    // Конвертуємо в пдф
    doc.ExportAsFixedFormat(TmpFile.ToString(), WdExportFormat.wdExportFormatPDF);
    PDFreader.src = TmpFile.ToString();
    PDFreader.Show();
}
catch (Exception ex) { MessageBox.Show(ex.Message); }
finally
{

```

```

        doc.Close(WdSaveOptions.wdDoNotSaveChanges,
WdOriginalFormat.wdOriginalDocumentFormat, false);
        app.Quit(WdSaveOptions.wdDoNotSaveChanges);
    }

    private void FindAndReplace(object findText, object replaceText)
    {
        app.Selection.Find.Execute(ref findText, true, true, false, false, false, true,
false, 1,
        ref replaceText, 2, false, false, false, false);
    }
}
}

```

HealthRisk.cs:

```

using System;
using MySql.Data.MySqlClient;

namespace GaussDispersionModel
{
    class HealthRisk
    {
        private double c; // Рівень впливу
        private double rfc; // Безпечний рівень впливу

        DB db = new DB();

        public HealthRisk(double c, int substanceId)
        {
            this.c = c;

            // Отримання значення rfc з БД
            MySqlCommand getRfcCommand = new MySqlCommand("select rfc from substance where id
= @substanceId; ", db.getConnection());
            getRfcCommand.Parameters.Add("@substanceId", MySqlDbType.String).Value =
substanceId;

            db.openConnection();
            this.rfc = Convert.ToDouble(getRfcCommand.ExecuteScalar());
            db.closeConnection();
        }

        // Коефіцієнт небезпеки
        public double dangerCoeff()
        {
            return c / rfc;
        }

        // Висновок щодо ризику
        public string conclusion()
        {
            string conclusion = "Неканцерогенний ризик впливу на здоров'я людини - ";

            if (dangerCoeff() < 1)
            {
                conclusion += "малий";
            }
        }
    }
}

```

```

        else if (dangerCoeff() > 1)
        {
            conclusion += "великий";
        }
        else
        {
            conclusion += "гранично допустимий";
        }
        return conclusion;
    }
}
}

```

ConcentrationForm.cs:

```

using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

```

```

namespace GaussDispersionModel.Forms
{
    public partial class ConcentrationForm : Form
    {
        private readonly double emissionPower;
        private readonly double sourceHeight;
        private readonly double sourceDiameter;
        private readonly double windSpeed;
        private readonly double windHeight;
        private readonly double gasSpeed;
        private readonly double ambienTemp;
        private readonly double gasTemp;
        private readonly string atmStability;
        private readonly string locality;
        private readonly int substanceId;

        DB db = new DB();

        public ConcentrationForm(double emissionPower, double sourceHeight, double
sourceDiameter,
            double windSpeed, double windHeight, double gasSpeed, double gasTemp, double
ambienTemp,
            string atmStability, string locality, int substanceId)
        {
            InitializeComponent();

            this.emissionPower = emissionPower;
            this.sourceHeight = sourceHeight;
            this.sourceDiameter = sourceDiameter;
            this.windSpeed = windSpeed;
            this.windHeight = windHeight;
            this.gasSpeed = gasSpeed;
            this.gasTemp = gasTemp;
            this.ambienTemp = ambienTemp;
            this.atmStability = atmStability;
            this.locality = locality;
            this.substanceId = substanceId;
        }

        // Кнопка Координати
    }
}

```

```

private void helpBtn_Click(object sender, EventArgs e)
{
    CoordinatesForm coordinatesForm = new CoordinatesForm();
    coordinatesForm.Show();
}

// Кнопка розрахунку
private void calcBtn_Click(object sender, EventArgs e)
{
    try
    {
        double x = Convert.ToDouble(xBox.Text);
        double y = Convert.ToDouble(yBox.Text);
        double z = Convert.ToDouble(zBox.Text);
        double c;

        Dispersion release = new Dispersion(emissionPower, sourceHeight,
sourceDiameter,
        windSpeed, windHeight, gasSpeed, gasTemp, ambienTemp, atmStability,
locality);

        c = release.dotConcentration(x, y, z);
        dotConcentrationBox.Text = Convert.ToString(c);

        //Якщо речовина є в БД
        if (substanceId != 0)
        {
            double gdk;
            string conclusion = "Концентрація речовини в точці (x; y; z) - ";

            HealthRisk risk = new HealthRisk(c / 0.001, substanceId);

            // Отримуємо значення ГДК
            MySqlCommand getGdkCommand = new MySqlCommand("select gdk from substance
where id = @substanceId; ", db.getConnection());
            getGdkCommand.Parameters.Add("@substanceId", MySqlDbType.String).Value =
substanceId;

            db.openConnection();
            gdk = Convert.ToDouble(getGdkCommand.ExecuteScalar());
            db.closeConnection();

            // Порівнюємо з ГДК
            if (gdk < c / 0.001)
            {
                conclusion += "перевищує гранично допустиму концентрацію. ";
            }
            else
            {
                conclusion += "не перевищує гранично допустиму концентрацію. ";
            }

            // Висновок щодо ризику
            conclusion += risk.conclusion();

            // Повертаємо значення
            gdkBox.Text = Convert.ToString(gdk);
            dangerCoeffBox.Text = Convert.ToString(risk.dangerCoeff());
            conclusionBox.Text = conclusion;
        }
        else // Якщо речовини немає в БД
        {

```



```

        db.openConnection();
        this.substanceId = Convert.ToInt32(command.ExecuteScalar());
        db.closeConnection();

        ConcentrationForm concentrationForm = new ConcentrationForm(emissionPower,
sourceHeight,
        sourceDiameter, windSpeed, windHeight, gasSpeed, gasTemp, ambienTemp, atmStab,
locality, substanceId);

        concentrationForm.Show();
    }
}

// Генерація звіту
private void reportBtn_Click(object sender, EventArgs e)
{
    if (getParameters() && checkParameters())
    {
        ReportForm reportForm = new ReportForm(emissionPower, sourceHeight,
locality, objectName, substance);

        reportForm.Show();
    }
}

// Функція отримання значення з форми
private bool getParameters()
{
    try
    {
        emissionPower = Convert.ToDouble(emissionPowerBox.Text);
        sourceHeight = Convert.ToDouble(sourceHeightBox.Text);
        sourceDiameter = Convert.ToDouble(sourceDiameterBox.Text);
        windSpeed = Convert.ToDouble(windSpeedBox.Text);
        windHeight = Convert.ToDouble(windHeightBox.Text);
        gasSpeed = Convert.ToDouble(gasSpeedBox.Text);
        gasTemp = Convert.ToDouble(gasTempBox.Text);
        ambienTemp = Convert.ToDouble(ambienTempBox.Text);
        locality = localityBox.Text;
        atmStab = atmStabilityBox.Text;
        substance = substanceBox.Text;
        objectName = objectNameBox.Text;

        return true;
    }
    catch (Exception)
    {
        MessageBox.Show("Введіть коректні дані!");
        return false;
    }
}

// Перевірка коректності даних
private bool checkParameters()
{
    bool ind = true;

    if (emissionPower < 0)
    {

```

```

        ind = false;
        MessageBox.Show("Потужність викиду не може бути від'ємною");
    }
    if (sourceHeight < 0)
    {
        ind = false;
        MessageBox.Show("Висота джерела викиду не може бути від'ємною");
    }
    if (sourceDiameter <= 0)
    {
        ind = false;
        MessageBox.Show("Діаметр джерела викиду має бути додатнім");
    }
    if (windSpeed < 0)
    {
        ind = false;
        MessageBox.Show("Швидкість вітру не може бути від'ємною");
    }
    if (windHeight < 0)
    {
        ind = false;
        MessageBox.Show("Висота виміру швидкості вітру не може бути від'ємною");
    }
    if (gasSpeed < 0)
    {
        ind = false;
        MessageBox.Show("Швидкість газу в гирлі не може бути від'ємною");
    }
    }

    return ind;
}

// Вибір речовини з списку
private void substanceBtn_Click(object sender, EventArgs e)
{
    SubstanceListForm sub = new SubstanceListForm();
    sub.Owner = this;
    sub.Show();
}
}
}

```

SubstanceListForm.cs:

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace GaussDispersionModel.Forms
{
    public partial class SubstanceListForm : Form
    {
        DB db = new DB();

        public SubstanceListForm()
        {
            InitializeComponent();
            LoadData();
        }
    }
}

```

```

    }

    private void LoadData()
    {
        MySqlCommand command = new MySqlCommand("select `name` from substance ; ",
        db.getConnection());

        db.openConnection();

        MySqlDataReader reader = command.ExecuteReader();

        List<string[]> data = new List<string[]>();

        while (reader.Read())
        {
            data.Add(new string[1]);
            data[data.Count - 1][0] = reader[0].ToString();
        }

        reader.Close();
        db.closeConnection();

        foreach (string[] s in data)
        {
            dataGridView1.Rows.Add(s);
        }
    }

    private void substanceBtn_Click(object sender, EventArgs e)
    {
        MainForm mainF = this.Owner as MainForm;

        if (mainF != null)
        {
            mainF.substanceBox.Text = dataGridView1.CurrentCell.Value.ToString(); ;
            this.Close();
        }
    }
}
}
}

```

ДОДАТОК В

Модуль юриста в системі комплексного екологічно-економічного моніторингу

Опис програмного модулю

УКР.НТУУ"КПІ імені Ігоря Сікорського" _ТЕФ_АПЕПС_ТМ-81253_22Б 13-1

Аркушів 9

Київ 2022

АНОТАЦІЯ

Даний розділ містить в собі опис розробленого програмного продукту та загальні відомості про нього. В додатку реалізована можливість моделювання концентрації шкідливих речовин в повітрі та аналізування впливу цих речовин на здоров'я людини.

Програмне забезпечення було розроблено мовою програмування C# з використанням системи керування базами даними MySql.

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ.....	3
2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯІ.....	4
3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ.....	5
4. ВИКОРИСТУВАНІ ТЕХНІЧНІ ЗАСОБИ.....	6
5. ВИКЛИК І ЗАВАНТАЖЕННЯ.....	7
6. ВХІДНІ І ВИХІДНІ ДАНІ.....	8

ЗАГАЛЬНІ ВІДОМОСТІ

Програмне забезпечення було створене для моделювання гауссового шлейфу повітряної дисперсії забруднюючих речовин в атмосферному повітрі. Для коректної роботи додатка необхідно мати пристрій на основі операційної системи Windows 7 та вище. В встановлені додаткового програмного забезпечення немає необхідності. Додаток створений за допомогою використання мови програмування C# і інтерфейсу програмування Windows Forms в середовищі розробки Visual Studio 2019. Для роботи з БД було обрано СКБД MySQL.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Додаток був розроблений з метою здійснення екологічного моніторингу та прискорення прийняття важливих рішень при забрудненні атмосферного повітря різними забруднюючими речовинами та підвищення точності розрахунків.

Основні можливості програмного продукту це: розрахунок концентрації речовини, оцінка ризиків впливу на здоров'я людини та формування звітної документації.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Програмне забезпечення складається з 7 форм, які забезпечують весь необхідний функціонал, до цих форм належать:

1. Головне вікно, в якому користувач має ввести вхідні параметри та обрати подальшу дію;
2. Форма з списком речовин в БД;
3. Вікно розрахунку концентрації, в який користувач отримує значення концентрації речовини в точці простору, значення гранично допустимої концентрації, коефіцієнт впливу та висновок, щодо ризику впливу на здоров'я людини;
4. Вікно генерації звіту, де користувач має ввести параметри розрахункового прямокутника;
5. Вікно звіту, в якому користувач отримує звіт в форматі PDF;
6. Вікно з класами стабільності атмосфери, яке допомагає правильно обрати клас;
7. Вікно з наочним відображенням напрямку координат, яке допомагає правильно обрати координати для розрахунків.

ВИКОРИСТУВАНІ ТЕХНІЧНІ ЗАСОБИ

Програмний продукт був написаний мовою програмування C# в середовищі розробки Visual Studio. Для розробки користувацького інтерфейсу, було використано Windows Forms. Технологія COM Interop використовувалась для роботи з файлами. Робота з базою даних проводилась за допомогою СКБД MySQL та з використанням простору імен MySql.Data.MySqlClient .

ВИКЛИК І ЗАВАНТАЖЕННЯ

Розроблене програмне забезпечення працює без необхідності в інсталяції. Щоб почати роботу з програмою, необхідно відкрити папку з програмним продуктом та запустити файл GaussDispersionModel.exe.

Після запуску файла GaussDispersionModel.exe відкривається початкове вікно. Повний функціонал системи доступний всім її користувачам.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними для програмного забезпечення є інформація про викиди, джерело викидів забруднюючих речовин і навколишнє середовище, а саме:

- Назва об'єкту дослідження
- Назва речовини
- Потужність викиду
- Висота джерела викиду
- Діаметр джерела викиду
- Середня швидкість вітру
- Висота виміру середньої швидкості вітру
- Швидкість газу в гирлі
- Температура газів
- Температура навколишнього середовища
- Клас стабільності атмосфери
- Тип місцевостям

Вихідними даними є : значення концентрації речовини в точці простору, значення гранично допустимої концентрації, оцінка ризику впливу на здоров'я людини, звіт на основі отриманої інформації.