

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

Факультет математики, фізики та інформаційних технологій

Кафедра механіки, автоматизації та інформаційних технологій

Кваліфікаційна робота

на здобуття ступеня вищої освіти «магістр»

«Інформаційна технологія прогнозування пожежонебезпеки та динаміки розвитку пожежі»

«Information technology for forecasting fire danger and dynamics of fire development»

Виконала: здобувачка заочної форми навчання спеціальності 126 Інформаційні системи та технології

Освітня програма Інформаційні системи та технології

Молошнюк Ірина Олександрівна

(прізвище, ім'я, по-батькові здобувача)

Керівник к.ф.-м.н, доцент Рачинська А.Л. _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент д.т.н, професор Волков В.Е.
(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
*механіки, автоматизації та
інформаційних технологій*
№ ____ від ____ . ____ . 20 ____ р.

Захищено на засіданні ЕК № ____
протокол № __ від ____ . ____ . 20 ____ р.

Оцінка _____ / _____ / _____
(за національною шкалою/шкалою ECTS/ бали)

Завідувачка кафедри
_____ Алла РАЧИНСЬКА
(підпис)

Голова ЕК
_____ Володимир ВИЧУЖАНІН
(підпис)

Одеса 2023

АНОТАЦІЯ

У кваліфікаційній роботі проведено дослідження теми "Інформаційна технологія прогнозування пожежонебезпеки та динаміки розвитку пожежі". Зокрема, розроблено та впроваджено інформаційну технологію, спрямовану на удосконалення методів прогнозування пожежної небезпеки та моделювання динаміки їхнього розвитку.

Об'єкт дослідження – пожежонебезпека.

Предмет дослідження – фізико-хімічні властивості горючих речовин.

У процесі виконання роботи здійснено аналіз існуючих методів та технологій у сфері прогнозування пожеж та вивчено основні фактори, що впливають на їх розвиток. На основі цього аналізу запропоновано і розроблено інформаційну технологію, спрямовану на ефективне прогнозування пожежонебезпеки та моделювання динаміки їхнього розвитку.

Результати роботи включають розроблену інформаційну технологію, яка може бути використана для прогнозування пожежонебезпеки та ефективного управління в ситуаціях пожежі. Дана технологія пропонує комплексний підхід до вирішення проблеми та має потенціал для подальшого розвитку та вдосконалення в галузі безпеки та запобігання пожежам.

ANNOTATION

In the qualification work, the topic "Information technology for predicting fire hazard and fire development dynamics" was studied. In particular, an information technology aimed at improving the methods of fire hazard forecasting and modeling the dynamics of their development has been developed and implemented.

The object of research is fire hazard.

The subject of research is the physical and chemical properties of combustible substances.

In the course of the work, an analysis of existing methods and technologies in the field of fire forecasting was carried out and the main factors influencing their development were studied. On the basis of this analysis, an information technology aimed at effective forecasting of fire hazards and modeling the dynamics of their development was proposed and developed.

The results of the work include the developed information technology that can be used to predict fire hazards and effective management in fire situations. This technology offers a comprehensive approach to solving the problem and has the potential for further development and improvement in the field of fire safety and prevention.

ЗМІСТ

ВСТУП	6
1 МАТЕМАТИЧНІ МОДЕЛІ В ПРОЦЕСАХ ГОРІННЯ	8
1.1 Аспекти молекулярної фізики	9
1.2 Дослідження математичної моделі теоретичної кількості повітря для горіння речовини.	11
1.3 Дослідження математичної моделі теоретичної температури горіння і тиску вибуху горючої рідини	14
1.4 Визначення концентраційних меж поширення полум'я	16
1.4.1 Математична модель концентраційних меж поширення полум'я на основі апроксимаційної формули.....	17
1.4.2 Математична модель концентраційних меж поширення полум'я за граничною теплотою згорання	18
1.4.3 Математична модель концентраційних меж поширення полум'я для суміші горючих речовин.....	19
1.5 Дослідження математичної моделі температурних меж поширення полум'я.....	19
2 ПРОГНОЗУВАННЯ ДИНАМІКИ РОЗВИТКУ ПОЖЕЖІ НА ОБ'ЄКТІ	21
2.1 Модель динаміки розвитку площі пожежі	23
2.2 Модель теплового балансу внутрішньої пожежі.....	25
2.3 Модель динаміки зміни висоти нейтральної зони у часі.....	26
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	28
3.1 Структура інформаційної технології	28
3.2 Головне вікно frmContainer.....	30
3.3 Вікно Air_quantity.	30
3.4 Вікно Concent_limits	32
3.5 Вікно Explosion	33
3.6 Вікно Thermal_intervals	33
3.7 Вікно Dynamic_fire	34
3.8 Бібліотека класів Exp_Hazard_Library	35
4 ТЕСТУВАННЯ РОБОТИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....	36
ВИСНОВКИ.....	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	38
ДОДАТОК А. КОД КЛАСУ frmContainer.....	39
ДОДАТОК Б. КОД КЛАСУ Air_quantity.....	42
ДОДАТОК В. КОД КЛАСУ Concent_limits.	42

ДОДАТОК Г. КОД КЛАСУ Explosion.....	50
ДОДАТОК Д. КОД КЛАСУ Thermal_intervals	52
ДОДАТОК Е. КОД КЛАСУ Dynamic_fire.....	52
ДОДАТОК Ж. БІБЛІОТЕКА КЛАСІВ Exp_Hazard_Library	58
ДОДАТОК З. ДОВІДКА ПРО ВПРОВАДЖЕННЯ.....	61

ВСТУП

Зростання обсягів виробництва та неперервне збільшення кількості пожеж стали причиною загострення проблем неорганізованого горіння. Сучасне сприйняття пожеж розглядає їх як складний комплекс взаємопов'язаних процесів і явищ, що вимагає системного підходу. Розвиток нових вогнегасних речовин та вдосконалення методів їх застосування стали актуальними завданнями в цьому контексті.

Боротьба з пожежами та розвиток науки про пожежі стали стратегічними напрямками в сучасному суспільстві. Дослідження процесів горіння на пожежі та їх вивчення набули великого значення для ефективного контролю над пожежами.

Оволодіння основними закономірностями фізичних та хімічних процесів надає змогу ефективно вирішувати завдання профілактики та гасіння пожеж. Горіння є основним процесом на пожежі. Знання суті цього явища, законів горіння, механізмів та засобів його припинення є необхідними для успішної роботи інженера пожежної безпеки в будь-якій сфері його діяльності.

Впровадження інформаційних технологій та математичних моделей у сферу прогнозування може значно підвищити точність та швидкість виявлення потенційних пожежних загроз. Використання інформаційних технологій та автоматизованих систем дозволяє швидше та точніше аналізувати великі обсяги даних, поліпшуючи можливості прогнозування та реагування.

В кваліфікаційній роботі досліджується та розробляється інформаційна технологія прогнозування пожежонебезпеки та динаміки розвитку пожежі.

Метою роботи є розробка інструментів для забезпечення прогнозу пожежонебезпеки, який може допомогти у попередженні можливих загроз та мінімізації ризиків пожежі.

Для досягнення поставленої мети необхідно виконати:

- збір та аналіз початкових даних;
- створити та розрахувати математичні моделі процесів горіння;

- розробити та застосувати методи прогнозування динаміки розвитку пожежі, використовуючи отримані математичні моделі;
- розробити інформаційну технологію розрахунку зазначених моделей для подальшого аналізу та реалізації прогнозування динаміки розвитку пожежі;
- візуалізувати отримані результати для подальшого аналізу.

Об'єкт дослідження кваліфікаційної роботи – пожежонебезпека.

Предмет дослідження – фізико-хімічні властивості горючих речовин.

Розробка інформаційної технології прогнозування пожежонебезпеки має практичне значення та велику соціальну важливість в сучасному світі.

Тематика кваліфікаційної роботи входить до плану науково-дослідних робіт кафедри механіки, автоматизації та інформаційних технологій Одеського національного університету імені І.І. Мечникова на 2023-2027 роки «Теорія і практика застосування математичного моделювання та інформаційних технологій в технологічному та організаційному керуванні» (номер держ. реєстрації 0123U103362).

1 МАТЕМАТИЧНІ МОДЕЛІ В ПРОЦЕСАХ ГОРІННЯ

Процес горіння як одна з форм фізико-хімічної взаємодії атомів і молекул може бути по-справжньому зрозумілий тільки на основі вивчення молекулярно-кінетичної теорії будови матерії.

Необхідно уявляти, що в хімічних процесах, перш ніж утворюються нові молекули, руйнуються старі. Енергія, необхідна для розриву зв'язків у молекулах пального й окислювача, називається енергією активації. Руйнування або ослаблення хімічних зв'язків у молекулах відбувається під дією теплового руху атомів. Що вища температура, то вища частка активних молекул, то ефективніші зіткнення і більша їх кількість. Для реакції горіння, як і для багатьох інших хімічних реакцій, справедливим є положення: підвищення температури на $10\text{ }^{\circ}\text{C}$ призводить до збільшення її швидкості вдвічі (закон Арреніуса). Крім того, швидкість реакції відповідно до закону діючих мас збільшується зі зростанням концентрації реагентів. Швидкість горіння максимальна при стехіометричному складі суміші – коли відношення реагентів відповідає коефіцієнтам у рівнянні реакції.

Горінню як хімічній реакції, що йде з великим виділенням тепла, відповідає низка фізичних явищ. Наприклад, відбувається перенесення тепла реагуючих речовин і продуктів горіння теплопровідністю за рахунок молекулярної та турбулентної дифузії. Тепломасообмін також здійснюється за рахунок конвекції. Промениста теплопередача також сприяє теплообміну. Усі перераховані процеси взаємопов'язані. Швидкість хімічної реакції горіння визначається процесами теплопередачі, взаємною дифузією пального, окислювача, продуктів горіння. Своєю чергою, температура, швидкість горіння залежать від інтенсивності хімічної реакції. Важлива особливість явища горіння – здатність до просторового поширення (лісові та степові пожежі).

У горінні фізичні процеси й умови є домінуючими і визначають вид горіння, його параметри, але в основі горіння лежить хімічне перетворення

пального й окислювача на продукти горіння. На пожежі найчастіше спостерігається турбулентне, дифузійне, гомогенне горіння.

Показники пожежо- та вибухонебезпечних властивостей матеріалів визначають на всіх стадіях процесу горіння: займання, горіння і гасіння з урахуванням їхнього агрегатного стану. Так, наприклад, температуру спалаху і температурні межі поширення полум'я вивчають під час горіння рідин, температуру самозаймання – під час розгляду механізму цього явища тощо.

Для оцінки пожежонебезпечних властивостей речовин і матеріалів використовують розрахункові та експериментальні методи. Більшість розрахункових методів є наближеними, мають характер емпіричних залежностей.

Пожежа являє собою комплекс взаємопов'язаних процесів горіння, теплообміну і газообміну, які спричиняють такі вторинні явища, як задимлення, об'ємний спалах, вибухи, обвалення тощо. Під час аналізу пожежі її заведено розбивати на три зони: горіння, теплового впливу і задимлення. Кожна з них має свої особливості та характеристики, що залежать від виду пожежі. За характером тепло- і газообміну пожежі ділять на два види: відкриті та внутрішні. Відкритими називаються пожежі, в яких присутній теплогазообмін тільки зони горіння з навколишнім середовищем. Це пожежі, що протікають на відкритому просторі. Внутрішніми називаються пожежі, що протікають у приміщеннях. У цих випадках під час аналізу пожежі доводиться враховувати також процеси тепло- і газообміну зони горіння з огорожувальними конструкціями і приміщення із зовнішнім середовищем.

1.1 Аспекти молекулярної фізики

Визначимо деякі ключові елементи, які часто стають значущими при вирішенні різноманітних практично важливих завдань.

1 моль речовини – кількість речовини, маса якої чисельно дорівнює молекулярній масі, вираженій у грамах. Наприклад, маса 1 моля водню (H_2) дорівнює 2 г/моль або $2 \cdot 10^{-3}$ кг/моль, а маса 1 моля метану – 16 г/моль або

$16 \cdot 10^{-3}$ кг/моль. 1 кіломоль речовини в тисячу разів більший за 1 моль, тому його маса в 10^3 більше.

У зв'язку з тим, що маса 1 моля речовини чисельно дорівнює її молекулярній масі, то в 1 молі будь-якої речовини міститься однакова кількість молекул $N = 6,02 \cdot 10^{23}$ молекул/моль (число Авогадро). У газоподібному стані 1 моль будь-якої речовини за нормальних умов займає об'єм, що дорівнює 22,4 л. Таким чином, об'єм 1 моля будь-якого газу за $T_0 = 273$ К і $P_0 = 1,013 \cdot 10^5$ Па становить $22,4 \cdot 10^{-3}$ м³/моль. Об'єм 1 кіломоля в тисячу разів більший і дорівнює 22,4 м³/кмоль.

Об'єм 1 моля газу можна визначити з рівняння стану (Клапейрона – Менделєєва):

$$pV = \frac{m}{M} RT \quad (1.1)$$

де m - маса речовини; M - маса одного моля; R - універсальна газова стала,

$$R = 8,31 \text{ [Дж/(моль} \cdot \text{К)]}.$$

Для 1 моля ($m/M = 1$) об'єм газу за нормальних умов

$$V_0 = \frac{RT_0}{P_0} = \frac{8,31 \text{ Дж/(моль} \cdot \text{К)} \cdot 273 \text{ К}}{1,013 \cdot 10^5 \text{ Па}} = 22,4 \cdot 10^{-3} \text{ м}^3/\text{моль} \quad (1.2)$$

(Дж = Н · м, Па = Н/м²).

Густина речовини в газоподібному стані за нормальних умов

$$\rho = \frac{M}{V_0} \quad (1.3)$$

Якщо умови відрізняються від нормальних, необхідно ввести поправки на температуру і тиск. Для цього можна використовувати об'єднаний газовий закон:

$$\frac{pV}{T} = \frac{p_0 V_0}{T_0} \rightarrow V = V_0 \frac{p_0 T}{p T_0} \quad (1.4)$$

Використання таких величин, як маса й об'єм 1 моля речовини, дає змогу значно спростити розв'язання багатьох практично важливих задач. Наприклад, при визначенні об'єму пари за відомою масою рідини, що випарувалася, або

визначенні об'єму, займаного газом, що вийшов з балона, коли відома втрата маси в ньому.

1.2 Дослідження математичної моделі теоретичної кількості повітря для горіння речовини.

Узагальнений запис бруutto-рівняння матеріального і теплового балансу реакції горіння має вигляд:

$$n_r[\Gamma] + n_o[o] = n_{\text{пгг}}[\text{пг}] + Q \quad (1.5)$$

де n_r , n_o , $n_{\text{пгг}}$ – стехіометричні коефіцієнти при відповідних речовинах, ($[\Gamma]$ – пальне, $[o]$ – окислювач, $[\text{пг}]$ – продукти горіння); Q – тепловий ефект хімічної реакції.

Для складання матеріального балансу процесів горіння використовують рівняння елементарних реакцій окиснення елементів, що входять до складу пального, виражений у масових відсотках кожного хімічного елемента на одиницю маси пального.

Для індивідуальних речовин запис рівняння матеріального балансу здійснюється відразу, але принцип залишається той самий.

Для сумішей горючих газів записують рівняння матеріального балансу кожного газу.

Під час горіння реакції між паливом, продуктами його термічного розкладання протікають в об'ємі полум'я – у зоні горіння.

Мінімальна (теоретична) кількість повітря, необхідна для повного спалювання 1 кг твердого чи рідкого горючого матеріалу або 1 м³ горючого газу, – називається теоретичною кількістю повітря і позначається $V_{\text{п}}^0$. Дійсна кількість повітря $V_{\text{п}}^{\text{пр}}$ є добутком коефіцієнта надлишку повітря α і теоретичної кількості повітря $V_{\text{п}}^0$.

Коефіцієнт надлишку повітря (α) показує, у скільки разів об'єм повітря, що надійшов на горіння, більший за теоретичний об'єм повітря, необхідний для повного згорання одиниці кількості речовини в стехіометричній суміші.

Повний, дійсний об'єм продуктів горіння знаходиться з урахуванням надлишку повітря ($\alpha > 1$):

$$V_{\text{пр}} = V_{\text{пр}}^0 + (\alpha - 1)V_{\text{п}}^0 \quad (1.6)$$

Розрахунок ведеться на 1 кг (1 кмоль, 1 м³) твердого, рідкого або газоподібного пального.

Адіабатичною температурою горіння називається температура, до якої нагріваються продукти горіння, коли все тепло, що виділилося в результаті згоряння, йде на їх нагрівання.

Дійсна температура горіння завжди нижча за адіабатичну, оскільки частина тепла втрачається на випромінювання.

Під час горіння на пожежах витрата повітря відрізняється від теоретично необхідної. Дійсна витрата повітря на одиницю пального записується як $V_{\text{п}} = \alpha V_{\text{п}}^0$. Множник α називається коефіцієнтом надлишку повітря. Різниця між дійсною і теоретично необхідною кількістю повітря називається надлишком повітря.

$$\Delta V = V_{\text{п}} - V_{\text{п}}^0$$

або

$$\Delta V_{\text{п}} = V_{\text{п}}^0 (\alpha - 1) \quad (1.7)$$

Визначення кількості повітря та об'єму продуктів горіння виявляється можливим на підставі закону збереження маси за рівняннями хімічних реакцій.

Під час складання рівняння матеріального балансу процесів горіння заведено враховувати не тільки кисень, що бере участь у реакції окиснення, а й азот, що входить до складу повітря. Для практичних розрахунків приймають, що повітря складається з 21 % кисню і 79 % азоту. Таким чином, об'ємне співвідношення азоту і кисню в повітрі становитиме:

$$\frac{\varphi_{N_2}}{\varphi_{O_2}} = \frac{79}{21} = 3,76 \quad (1.8)$$

де φ_{N_2} , φ_{O_2} – відповідно (% об.) вміст азоту та кисню в окислювальному середовищі.

Отже, на 1 м³ (кмоль) кисню в повітрі припадає 3,76 м³ (кмоль) азоту.

Вагове співвідношення азоту і кисню в повітрі можна визначити, виходячи зі співвідношення:

$$\frac{\varphi_{N_2}}{\varphi_{O_2}} = \frac{M_{N_2}}{M_{O_2}} = \frac{79 \cdot 28}{21 \cdot 32} = 3,29 \quad (1.9)$$

де M_{N_2} , M_{O_2} – молекулярні маси відповідно кисню та азоту.

У загальному вигляді формулу для визначення кількості повітря можна записати:

$$V_{\text{п}}^0 \frac{n_{O_2} + n_{N_2}}{n_{\text{г}}} \quad (1.10)$$

Де n_{O_2} , n_{N_2} , $n_{\text{г}}$ – число кмолів кисню, азоту й горючої речовини в рівнянні матеріального балансу, [кмоль].

Аналогічно обсяг продуктів горіння:

$$V_{CO_2} = \frac{n_{CO_2}}{n_{\text{г}}}, \quad V_{H_2O} = \frac{n_{H_2O}}{n_{\text{г}}} \quad \text{та} \quad V_{N_2}^0 = \frac{n_{N_2}}{n_{\text{г}}} \quad (1.11)$$

де $V_{N_2}^0$ – об'єм азоту, розрахований з рівняння матеріального балансу, [кмоль/кмоль].

Об'єм продуктів горіння:

$$V_{\text{пг}}^0 = V_{CO_2} + V_{H_2O} + V_{N_2}^0 \quad (1.12)$$

Якщо горіння відбувається з надлишком повітря ($\alpha > 1$), то азоту в продуктах горіння буде більше:

$$V_{N_2} = V_{N_2}^0 + 0,79 \Delta V_{\text{в}} \quad (1.13)$$

і в продуктах горіння з'явиться кисень

$$V_{O_2} = 0,21 \Delta V_{\text{в}} \quad (1.14)$$

У цьому випадку повний об'єм продуктів горіння

$$V_{\text{пг}} = V_{\text{пг}}^0 + \Delta V_{\text{п}} \quad \text{або} \quad (1.15)$$

$$V_{\text{пг}} = V_{CO_2} + V_{H_2O} + V_{N_2} + V_{O_2}$$

Для газоподібних горючих речовин розрахунок об'ємів повітря і продуктів горіння найчастіше проводять у м³/м³. Оскільки 1 кмоль будь-якого

газу в однакових умовах займає один і той самий об'єм (за нормальних умов - $22,4 \text{ м}^3$), то об'єм, розрахований у $\text{м}^3/\text{м}^3$, чисельно буде таким самим, як і в $\text{кмоль}/\text{кмоль}$.

Якщо горюча речовина перебуває в конденсованому стані (рідкому або твердому), то, як правило, розрахунки об'ємів повітря і продуктів горіння проводять у $\text{м}^3/\text{кг}$. Для цього кмоль повітря або продукту горіння переводять у м^3 , множачи на об'єм одного кмоль газу ($22,4 \text{ м}^3/\text{кмоль}$), а кмоль пального переводять у кг , множачи на масу одного кіломоля пального (M_r , $\text{кг}/\text{кмоль}$).

$$V_{\text{п}}^0 = \frac{(n_{\text{O}_2} + n_{\text{N}_2})22,4}{n_r M_r} \quad (1.16)$$

У разі згоряння суміші індивідуальних хімічних речовин розрахунок необхідної кількості повітря та об'єму продуктів горіння проводять за кожною з речовин, а потім підсумовують з урахуванням її процентного вмісту в суміші.

Якщо в процесі горіння було задано іншу температуру, то об'єм продуктів горіння і повітря розраховують з урахуванням об'єму, що займає один кмоль газу за цієї температури:

$$V = \frac{22,4 P_0 T}{T_0 P} \quad (1.17)$$

Де T і P – задані температура та тиск.

1.3 Дослідження математичної моделі теоретичної температури горіння і тиску вибуху горючої рідини

Під температурою горіння розуміють максимальну температуру, до якої нагріваються продукти горіння. Заведено розрізняти адіабатичну температуру горіння, що розраховується без урахування втрат тепла в навколишній простір, і дійсну температуру горіння, що враховує ці тепловтрати.

Температуру горіння можна розрахувати, якщо відома кількість теплоти, що виділяється під час згоряння горючої речовини, склад і об'єм продуктів горіння, їхня теплоємність або тепловміст. Тепловий ефект хімічної реакції горіння (тобто кількість теплоти, що виділяється під час екзотермічної реакції) називається нижчою теплотою згоряння речовини $Q_{\text{н}}$.

Якщо пальне має у своєму складі водень, то під час його згоряння утворюється вода в газоподібному стані, яка, охолоджуючись, буде конденсуватися. Оскільки в процесі конденсації парів тепло виділяється, то загальна кількість теплоти, отримана під час згоряння речовини, буде більшою за величину теплоти випаровування $Q_{\text{вип}}$:

$$Q_{\text{в}} = Q_{\text{н}} + Q_{\text{вип}} \quad (1.18)$$

де $Q_{\text{в}}$ – вища теплота згоряння речовини.

При розрахунках температури горіння користуються величиною $Q_{\text{н}}$, оскільки при температурі горіння вода перебуває в газоподібному стані. Значення нижчої теплоти згоряння речовини (тепловий ефект хімічної реакції) наводиться в довідковій літературі. Для деяких горючих речовин це значення наведено в додатку (див. показники пожежної небезпеки газів, рідин і твердих речовин). Ця величина може бути розрахована за законом Гесса,

$$Q_{\text{н}} = (\sum \Delta H_i n_i)_{\text{прод}} - (\sum \Delta H_i n_i)_{\text{вих}} \quad (1.19)$$

який говорить про те, що тепловий ефект хімічної реакції дорівнює різниці сум теплоти утворення продуктів реакції і теплоти утворення вихідних речовин.

Для розрахунку температури горіння складемо рівняння теплового балансу, вважаючи, що тепло, яке виділилося в результаті згоряння, нагріває продукти горіння (CO_2 , H_2O , N_2) від початкової температури T_0 до температури $T_{\text{г}}$.

$$Q_{\text{н}}(1 - \eta) = \sum c_{\text{рпгі}} V_{\text{пгі}} (T_{\text{г}} - T_0) \quad (1.20)$$

де η - коефіцієнт тепловтрат (частка втрат тепла на випромінювання, а також унаслідок неповноти згоряння); $c_{\text{рпгі}}$ і $V_{\text{пгі}}$ - теплоємність і об'єм i -го продукту горіння.

Визначаємо тиск вибуху парів речовини в суміші з повітрям на стехіометричній концентрації і за заданих умов за формулою:

$$P_{\text{тиск}} = \frac{P_0 T_{\text{тиск}} \sum_{i=1}^n n_i}{T_0 \sum_{i=1}^n m_i} \quad (1.21)$$

1.4 Визначення концентраційних меж поширення полум'я

Суміші горючих газів і парів з повітрям здатні займатися і поширювати полум'я тільки тоді, коли концентрація пального в них перебуває в певних обмежених межах, які називаються нижньою і верхньою межами поширення полум'я.

У межах одного гомологічного ряду зі збільшенням відносної молекулярної маси межі поширення полум'я, виражені в об'ємних відсотках, зменшуються. На нижній межі теплота згоряння більшості вуглеводнів відповідає приблизно величині 1830...2050 кДж/м³, а адиабатична температура горіння дорівнює 1500 К ± 110 К. Ці дані свідчать про те, що для займання горючих речовин і поширення полум'я необхідні певні умови.

Концепції граничних температур і теплоти згоряння використані при створенні методик розрахунку нижньої концентраційної межі поширення полум'я.

Зі збільшенням температури, як і впливає із сутності граничних явищ, область поширення полум'я розширюється, а зі зменшенням тиску звужується. Концентраційні межі поширення полум'я змінюються і від низки інших причин: присутності в суміші флегматизатора або інгібітора, потужності джерела запалювання.

Концентраційні межі поширення полум'я є чи не основною характеристикою пожежонебезпечних властивостей речовин і матеріалів. Потужність запалювальної іскри, температура самозаймання, температура горіння мають сенс тільки всередині концентраційної області поширення полум'я.

Межі поширення полум'я речовин і матеріалів визначають розрахунковими та експериментальними методами.

Поряд із прийнятою системою визначення меж за концентрацією пального в суміші, останніми роками для твердих горючих матеріалів почали доволі широко застосовувати межу поширення полум'я за киснем, що виявляється експериментально і має назву кисневого індексу. Цей параметр

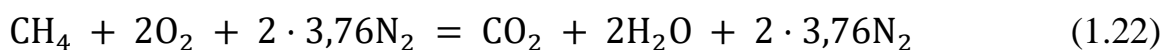
також залежить від тиску, температури. Кисневий індекс характеризує той мінімальний вміст кисню в суміші з азотом, за якого можливе займання горючого матеріалу з його подальшим горінням.

Межі поширення полум'я застосовують на практиці для класифікації виробництв за ступенем їхньої пожежної небезпеки, під час розрахунку гранично припустимих концентрацій горючих парів і газів у приміщеннях під час виконання вогневих робіт, для розрахунку вибухонебезпечних режимів робіт у середовищі, яке містить горючі гази і пари.

1.4.1 Математична модель концентраційних меж поширення полум'я на основі апроксимаційної формули

Газова суміш пального з окислювачем здатна спалахувати і поширювати полум'я тільки за певних концентрацій пального. Мінімальна концентрація пального, за якої суміш здатна спалахувати і поширювати полум'я, називається нижньою концентраційною межею поширення полум'я, а максимальна концентрація пального – верхньою концентраційною межею поширення полум'я. За концентрацій пального, нижчих за нижню концентраційну межу і вищих за вищу, його суміші з повітрям негорючі. Швидкість поширення полум'я та його температура мінімальні в граничних сумішах, тобто за концентрацій пального, що дорівнюють нижній концентраційній межі і верхній концентраційній межі. Максимальні значення швидкості та температури полум'я, як правило, мають суміші стехіометричного складу. Тому ці суміші найбільш пожежовибухонебезпечні. Концентрацію пального в стехіометричній суміші розраховують за рівнянням горіння.

Наприклад, із рівняння горіння видно, що стехіометрична суміш метану[9] з повітрям містить 1 моль метану, 2 моля кисню і 2·3,76 молей азоту. Концентрація пального в такій суміші



$$\varphi_{\text{стех}} = \frac{n_{\text{CH}_4} \cdot 100}{n_{\text{CH}_4} + n_{\text{O}_2} + n_{\text{N}_2}} = \frac{1 \cdot 100}{1 + 2 + 2 \cdot 3,76} = 9,5\%$$

За апроксимаційною формулою можна розраховувати значення як нижньої, так і верхньої концентраційних меж поширення полум'я:

$$\varphi_{н(в)} = \frac{100}{an+b} \quad (1.23)$$

де n – число молей кисню, необхідне для повного згорання одного моля горючої речовини, знаходимо з рівняння реакції горіння (стехіометричний коефіцієнт за наявності кисню); a і b – константи, які мають певні значення для нижньої та верхньої меж залежно від значення n , наведені в табл. 1

Таблиця 1

Область застосування	a	b
Для обчислення нижньої межі	8,684	4,679
Для обчислення верхньої межі		
за умови $n < 7,5$	1,550	0,560
за умови $n > 7,5$	0,768	6,554

1.4.2 Математична модель концентраційних меж поширення полум'я за граничною теплотою згорання

Відповідно до теплової теорії, наявність меж поширення полум'я можна пояснити тим, що для поширення полум'я в горючому паро- або газоповітряному середовищі необхідне виділення певної кількості тепла. Експериментально встановлено, що кількість тепла, що виділяється під час горіння сумішей із вмістом пального, що відповідає нижній концентраційній межі поширення полум'я, для більшості речовин є приблизно величиною постійною, що дорівнює 1830 кДж/м^3 . Цю величину називають граничною теплотою згорання $Q_{пр}$. Якщо відомо значення нижньої теплоти згорання речовини $Q_{н}$, то можна розрахувати концентрацію пального, %, у граничній за горючістю суміші, тобто величину нижньої концентраційної межі:

$$\varphi_{н} = \frac{Q_{пр}}{Q_{н}} 100 \quad (1.24)$$

$Q_{пр}$ - це кількість теплоти, що виділяється під час згоряння 1 м^3 граничної суміші пального з повітрям, а Q_H - кількість теплоти, що виділяється під час згоряння 1 м^3 пального [9].

1.4.3 Математична модель концентраційних меж поширення полум'я для суміші горючих речовин

Якщо пальне є сумішшю індивідуальних речовин з відомими концентраційними межами поширення полум'я, то для такої суміші можна встановити значення нижньої і верхньої межі за формулою Ле-Шательє

$$\varphi_{H(B)}^{CM} = \frac{1}{\sum \frac{\mu_i}{\varphi_{H(B)}i}} \quad (1.25)$$

де μ_i – мольна частка i -ї речовини в суміші; $\varphi_{H(B)}i$ - значення нижньої або верхньої концентраційної межі поширення полум'я i -ї речовини.

Ця формула справедлива для більшості сумішей речовин, хоча в деяких випадках можуть спостерігатися значні відхилення, пов'язані з хімічною взаємодією речовин[3].

1.5 Дослідження математичної моделі температурних меж поширення полум'я.

Температурними межами поширення полум'я характеризують пожежну небезпеку рідких горючих речовин. Нижня температурна межа займання - це така температура рідини, за якої концентрація насичених парів над її поверхнею дорівнює нижній концентраційній межі поширення полум'я. Аналогічно за верхньою температурною межею займання концентрація насичених парів дорівнює верхній концентраційній межі поширення. З цього випливає, що якщо для горючої речовини відомі концентраційні межі поширення полум'я, то за залежністю тиску насиченої пари від температури можуть бути знайдені температурні межі поширення.

Припустимо, відома нижня концентраційна межа поширення полум'я парів горючої рідини. Необхідно знайти нижню температурну межу займання, тобто

температуру, за якої концентрація насичених парів над поверхнею цієї рідини дорівнюватиме нижній концентраційній межі. За концентрацією парів φ_n (значення нижньої концентраційної межі поширення полум'я) можна знайти парціальний тиск насичених парів:

$$P_n = \frac{\varphi_n P_0}{100} \quad (1.26)$$

де P_0 - атмосферний тиск (загальний тиск пароповітряної суміші).

За тиском насиченої пари неважко встановити відповідну температуру рідини, яка і буде значенням нижньої температурної межі займання. Залежність тиску насиченої пари від температури для більшості рідин відома і наведена в довідковій літературі у вигляді таблиць (див. дод.), графіків або задана рівнянням Антуана $\lg P = f(T)$ [1].

Температурні межі займання можна визначити також за емпіричною формулою, що пов'язує їх із температурою кипіння індивідуальних рідин і константами K і l , характерними для цього гомологічного ряду

$$T_{n(v)} = K T_{\text{кип}} - l \quad (1.27)$$

де $T_{n(v)}$ - нижня або верхня температурна межа займання; $T_{\text{кип}}$ - температура кипіння, °С.

Значення K і l наведено в додатку 10.

2 ПРОГНОЗУВАННЯ ДИНАМІКИ РОЗВИТКУ ПОЖЕЖІ НА ОБ'ЄКТІ

Динамікою будь-якого процесу, в тому числі пожежі, називається залежність його параметрів від часу. Знання закономірностей зміни параметрів пожежі необхідне для гарантування безпеки людей, розроблення заходів для запобігання поширенню пожеж, для проектування систем автоматичного виявлення і гасіння, планування сил і засобів, необхідних під час ліквідації пожеж.

Безпосередньою причиною виникнення процесу горіння і, отже, пожежі є джерело запалювання. Ним може бути в принципі будь-яке джерело тепла достатньої потужності (побутовий кип'ятильник, рефлектор тощо). У багатьох випадках пожежа починається з прогрівання локальної ділянки поверхні горючого матеріалу й утворення над нею горючої суміші, яка потім запалюється. Променистий тепловий потік від факела, що утворився, нагріває сусідні ділянки поверхні, спричиняючи утворення і займання над ними горючої суміші, тобто починається процес поширення полум'я поверхнею, збільшення площі пожежі, а з ростом площі пожежі збільшується об'єм зони горіння і, відповідно, площа випромінювання, конвективні потоки. Усе це призводить до підвищення середньооб'ємної температури і, як наслідок, температури всіх предметів і матеріалів, що знаходяться в приміщенні. Своєю чергою, зростання температури матеріалів супроводжується збільшенням лінійної швидкості поширення фронту полум'я.

Якщо приміщення має великий вільний об'єм (зазвичай понад 1000 м³) або великі відкриті прорізи, процес поширення пожежі протікає шляхом поступового охоплення полум'ям предметів (виробів, матеріалів) одного за іншим. У приміщеннях меншого об'єму при досягненні середньооб'ємної температури 300-350 °С і наявності повітря вміст газоподібних продуктів термічного розкладання досягає нижньої концентраційної межі поширення полум'я. У цьому випадку швидкість поширення полум'я досягає максимального значення, оскільки процес відбувається вже не на поверхні, а в

об'ємі газового середовища. Швидкість поширення полум'я по газу становить від кількох метрів до сотень метрів на секунду залежно від складу газу. Зовні це сприймається як об'ємний спалах, тобто одночасне займання всіх горючих матеріалів, які перебувають у приміщенні. У результаті вільний об'єм приміщення перетворюється на зону горіння (об'ємна пожежа). Зазвичай це супроводжується руйнуванням скління прорізів унаслідок різкого наростання тиску. Відповідно в приміщення починає надходити свіже повітря.

Якщо вільний об'єм приміщення малий, то під час об'ємної пожежі швидко витрачається запас кисню, масова швидкість вигорання і, відповідно, інтенсивність тепловиділення починають поступово зменшуватися. Температура також знижується. У герметичному приміщенні може статися мимовільне загасання пожежі. Однак на практиці таке буває рідко. Як правило, завжди є приплив повітря через нещільно закриті отвори, щілини, а температура газового середовища в приміщенні встигає досягти температури займання горючих матеріалів. Тому через якийсь час, залежно від швидкості припливу повітря, у приміщенні створюється горюча суміш продуктів піролізу з повітрям і горіння інтенсифікується.

Якщо під час "загасання" забезпечити доступ повітря в приміщення (наприклад, відчинивши двері), станеться миттєве утворення і займання горючої суміші. А оскільки об'єм продуктів згорання більший за об'єм вихідної суміші, у приміщенні різко зросте тиск і відбудеться викид нагрітих газів. Такий спалах може бути причиною травмування або загибелі пожежників під час гасіння пожеж у підвалах, лазнях-саунах, гаражах, холодильниках. Ця стадія пожежі, протягом якої триває поширення фронту полум'я, називається початковою.

Після того як зростання площі пожежі в даному приміщенні припиняється, починається друга стадія - стадія розвитку, під час якої, за наявності достатнього припливу повітря, збільшується швидкість вигорання горючих речовин, теплота пожежі, зростає температура газового середовища. Наприкінці цієї стадії припиняється зміна параметрів процесів тепло- і

газообміну, температура досягає максимального значення і починається третя стадія пожежі – стаціонарна. За вільного розвитку пожежі горючі матеріали поступово вигорають, і пожежа переходить у стадію загасання.

2.1 Модель динаміки розвитку площі пожежі

На практиці для прогнозування можливої обстановки під час пожежі при розробці оперативного плану пожежогасіння приймають ряд припущень:

- Пожежна навантага у приміщенні розподілена рівномірно, отже, пожежа поширюється у всіх напрямках рівномірно.
- Залежно від місця виникнення горіння на початковій стадії зона горіння може набувати форми прямокутника, кола, півкола або кутової форми (рис. 2.1). В міру розвитку пожежі форма зони горіння може змінюватися і переходити у прямокутну. Перехід із однієї форми розвитку пожежі в іншу відбувається при підході фронту пожежі до огорожуючих конструкцій. При поширенні горіння в декількох приміщеннях складну форму пожежі умовно розбивають на декілька простих.
- При досягненні протипожежних конструкцій його форма відразу стає прямокутною.

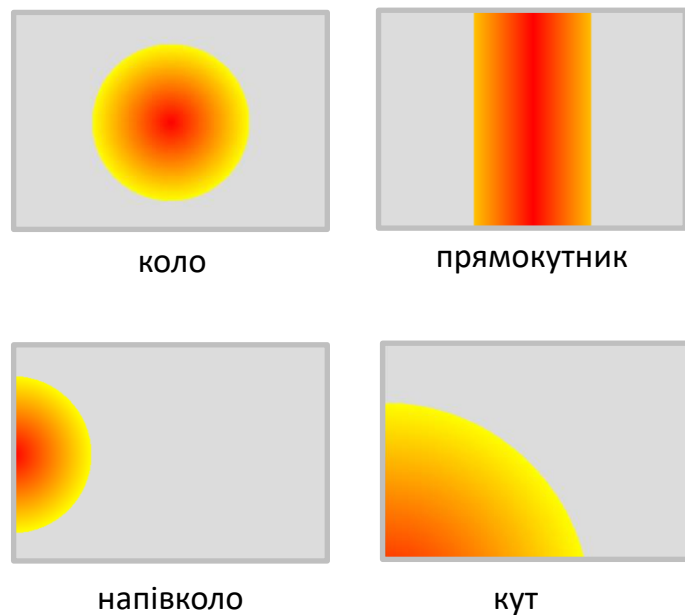


Рисунок 2.1 – Форми розвитку пожежі

У випадку, коли радіус розвитку пожежі менший за ширину приміщення, форма зони горіння буде чверть кола. Якщо радіус розвитку пожежі обмежений відстанню до стіни, зона горіння матиме форму півкола. Якщо осередок пожежі знаходиться у центрі приміщення і радіус розвитку пожежі є меншим за відстань до ближчої стіни, то форма зони горіння – коло. У випадку, коли радіус розвитку пожежі дорівнює або є більшим за відстань до ближчої стіни, то форма зони горіння – прямокутник.

Коли радіус розвитку пожежі перевищує відстань до протипожежних конструкцій, важливо враховувати час їхньої вогнестійкості. Після того, як протипожежні конструкції, такі як двері, прогорять, пожежа буде поширюватися у вільному напрямку із середньою швидкістю.

Фронт пожежі за кругової форми розвитку дорівнює периметру кола, за кутової – довжині дуги, обмеженої дугою, за прямокутної – ширині приміщення, помноженій на кількість напрямків розвитку пожежі. У таблиці 2 представлені вирази для обчислення площі пожежі[1]

Таблиця 2

Форма площі пожежі		
кругова	кутова	прямокутна
$S_{\Pi} = \pi R^2$ <i>При розвитку пожежі у формі півкола</i> $S_{\Pi} = 0,785D^2$	$S_{\Pi} = 0,5 aR^2$	$S_{\Pi} = ab$ <i>При розвитку пожежі у двох напрямках</i> $S_{\Pi} = a(b_1 + b_2)$

При дослідженні динамічної моделі будемо вважати, що в початковій стадії розвитку пожежі (тривалість якої приймають рівною 10 хв) $V_L = 0,5v_L^{\text{табл}}$ (м/хв), шлях, пройдений фронтом полум'я до моменту τ , дорівнюватиме:

$$L_{\tau} = 0,5v_L^{\text{табл}} \text{ при } \tau \leq 10 \text{ хв} \quad (2.1)$$

$$L_{\tau} = L_{10} + L_{\tau-10} \text{ при } \tau > 10 \text{ хв}$$

L_{10} – шлях, який проходить фронт полум'я за перші 10 хв;

$$L_{10} = 0,5v_L^{\text{табл}} * 10 = 5v_L^{\text{табл}}, [\text{м}]$$

$L_{\tau-10}$ – шлях, який проходить фронт полум'я за час, що залишився після 10 хв:

$$L_{\tau-10} = v_L^{\text{табл}} (\tau - 10), [\text{м}], \text{ тогдa при } \tau > 10 \text{ хв}$$

$$L = 5v_L^{\text{табл}} + v_L^{\text{табл}} (\tau - 10), [\text{м}]$$

Розрахунок площі пожежі на заданий момент часу доцільно проводити в такій послідовності (рис. 2.2):

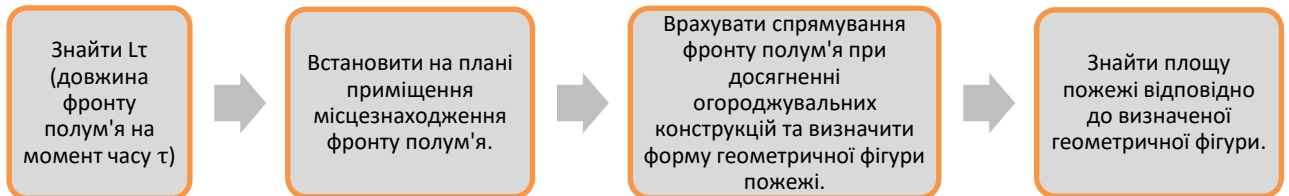


Рисунок 2.2 – Послідовність розрахунку площі пожежі

Такий підхід дозволяє систематично та ефективно проводити розрахунки та візуалізувати розвиток пожежі в приміщенні, враховуючи особливості її поширення та геометричні параметри.

2.2 Модель теплового балансу внутрішньої пожежі

Під температурою внутрішньої пожежі розуміється середньооб'ємна температура газового середовища в приміщенні в даний момент часу. Вона визначається співвідношенням кількості тепла, що виділяється внаслідок горіння і видаляється з приміщення. Якби був відсутній теплообмін вогнища пожежі з навколишнім середовищем, огорожувальними конструкціями, то максимальна температура, як і на відкритій пожежі, досягала б температури полум'я приблизно 1150 °С для твердих горючих матеріалів, приблизно 1250 °С для рідин і 1350 °С для газів.

У загальному вигляді тепловий баланс внутрішньої пожежі в кожен момент часу може бути представлений таким рівнянням:

$$q_n = q_{\text{вид}} + q_{\text{ср}} + q_k + q_{\text{вип}} \quad (2.2)$$

де q_n – інтенсивність виділення тепла (теплота пожежі); $q_{\text{вид}}$ – інтенсивність видалення тепла з приміщення разом із продуктами горіння; $q_{\text{ср}}$ – інтенсивність накопичення тепла газовим середовищем усередині

приміщення; q_k – інтенсивність поглинання тепла огорожувальними конструкціями; $q_{\text{вип}}$ – інтенсивність випромінювання тепла за межі приміщення через прорізи.

Усі величини, що входять до цього рівняння, змінюються в часі. Вони залежать від виду горючого матеріалу, його кількості, характеристик приміщення і будівельних матеріалів та багатьох інших чинників. Наприклад, q_k на стаціонарній стадії розвитку пожежі може досягати 50% від q_n . Загалом втрати тепла змінюються від 10 до 80 % q_n залежно від умов газообміну і тривалості горіння.

З усіх складових q_n , що входять до рівняння (2.2), тільки $q_{\text{ср}}$ безпосередньо витрачається на розвиток пожежі усередині цього приміщення, проте не можна сказати, що інші складові не становлять пожежної небезпеки. Величина $q_{\text{вип}}$ визначає можливість поширення пожежі на сусідні об'єкти, становить небезпеку для особового складу під час гасіння, призводить до підвищення температури будівельних конструкцій, знижує їхню несучу здатність і створює загрозу обвалення. Тепло, що видаляється з приміщення з продуктами горіння, сприяє поширенню пожежі на верхні поверхи будівлі.

Таким чином, тільки величина $q_{\text{ср}}$ визначає в кінцевому рахунку температуру внутрішньої пожежі. Рівняння теплового балансу використовується як для аналізу пожежі, так і для розрахунку середньооб'ємної температури

$$Q = \frac{\chi V_m F_n Q_n}{F_{\text{огр}}} [\text{КВТ/М}^2] \quad (2.3)$$

2.3 Модель динаміки зміни висоти нейтральної зони у часі

Під час пожежі в приміщенні газообмін виникає через різницю тисків між газовим середовищем всередині приміщення та зовнішнім повітрям. При нагромадженні газоподібних продуктів горіння та збільшенні середньооб'ємної температури тиск усередині приміщення починає змінюватися в порівнянні з тиском зовнішнього повітря. У верхній частині

приміщення тиск стає дещо більшим за атмосферний, у нижній частині - меншим, і на якійсь висоті h_0 - рівним атмосферному. Горизонтальна площина, розташована на висоті h_0 , на якій тиск газів усередині приміщення і повітря зовні рівні, називається площиною рівних тисків. Вона ділить приміщення на дві частини. Нижче площини рівних тисків у приміщення через отвори надходить повітря, вище - з приміщення виходять нагріті гази.

Залежно від відношення маси горючих матеріалів та кількості повітря, яке забезпечує їх згоряння, пожежа на третій стадії може протікати в двох режимах. У випадках, коли приплив повітря достатній для досягнення максимальної повноти згоряння, залежно від виду горючого матеріалу, масова швидкість вигорання залишається сталим параметром, невідомим від зміни інтенсивності газообміну. Оскільки тепловиділення пожежі, позначене як q_n , прямо пропорційне масовій швидкості вигорання, цей параметр також залишається сталим і не залежить від зміни інтенсивності газообміну. Такі пожежі відомі як регульовані навантаженням.

Під час необмеженого розвитку пожежі в такому режимі рівень інтенсивності газообміну між приміщенням та навколишнім середовищем формується відповідно до масової швидкості вигорання, тобто темпу утворення продуктів горіння. Збільшення подачі повітря за допомогою вентилятора або димососа призводить до зниження температури газового середовища в приміщенні за рахунок холодного зовнішнього повітря. Відкриття додаткових отворів або видалення продуктів горіння також може сприяти зниженню середньооб'ємної температури.

У випадках, коли інтенсивність газообміну обмежує масову швидкість вигорання, тобто приплив повітря не забезпечує максимальну повноту згоряння горючих матеріалів, пожежа відома як регульована вентиляція. Збільшення припливу повітря під час регульованої вентиляції призводить до зростання масової швидкості вигорання, інтенсивності тепловиділення та температури газового середовища в приміщенні.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

Інформаційна технологія, реалізована у вигляді Windows-застосунку з інтегрованою графічною візуалізацією даних.

Проект було розроблено у середовищі розробки Visual Studio на мові програмування C# із використанням 2D-графіки та компоненту Chart. Visual Studio, як інтегроване середовище розробки (IDE), надає широкий набір інструментів та зручностей, які значно спрощують процес розробки програмного забезпечення на платформі Microsoft. Зокрема, підтримка мови C# у Visual Studio дозволяє швидко і зручно створювати додатки для Windows, з легкістю взаємодіючи з різними компонентами та бібліотеками.

Вибір 2D-графіки вказує на спрощення та оптимізацію роботи з графічними об'єктами. У випадку, коли зображення та візуалізація не потребують тривимірної просторової інтерпретації, використання 2D-графіки є зручним та ефективним рішенням, яке дозволяє концентруватися на суті завдання без зайвих складнощів.

Компонент Chart від Microsoft надає потужний інструмент для візуалізації даних у вигляді графіків та діаграм. Цей компонент вбудований у Visual Studio та пропонує гнучкість та розширені можливості для створення різноманітних типів графіків. Використання Chart дозволяє ефективно представляти дані користувачу, використовуючи зручний інтерфейс та візуально привабливий дизайн.

Отже, обрана комбінація Visual Studio, мови програмування C#, 2D-графіки та компонента Chart визначена прагматичним підходом, спрямованим на досягнення високої ефективності, зручності розробки та якісної візуалізації результатів застосунку.

3.1 Структура інформаційної технології

Головне вікно програми є MDI-контейнером, що функціонує як основна частина інтерфейсу, а також містить робочу панель з п'ятьма кнопками. Кожна кнопка ініціює відкриття відповідного вікна, де користувач може вводити дані,

отримувати результати та робити висновок, спираючись на отримані результати.

На рис. 2.3 зображені модулі інтерфейсу з яких складається інформаційна технологія

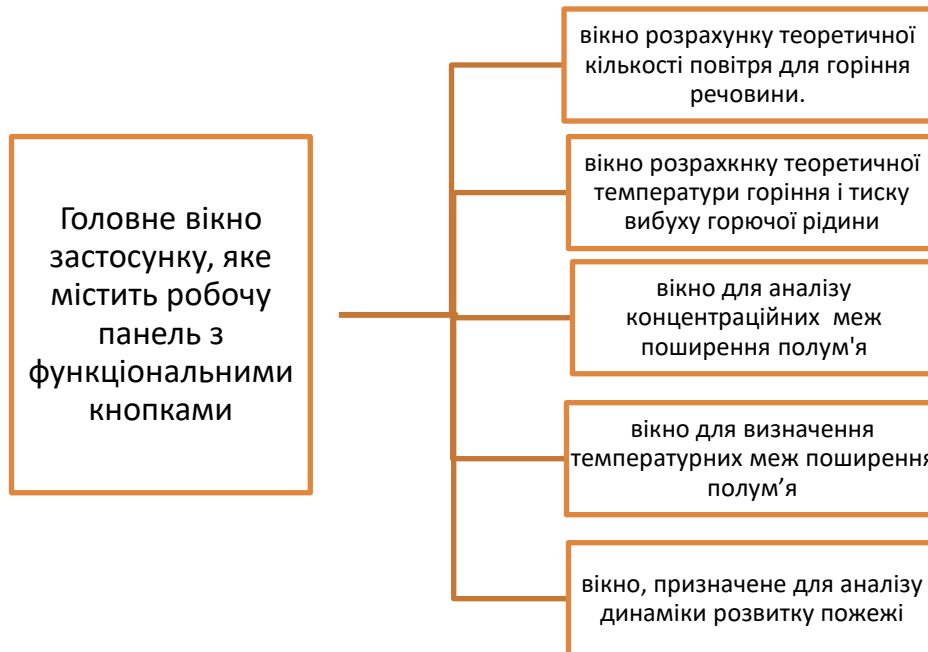


Рисунок 2.3 – Структура інформаційної технології

Для більшого розуміння було створено діаграму для відображення функціональності інформаційної технології та її взаємодії з користувачем застосунку (рис. 2.4).

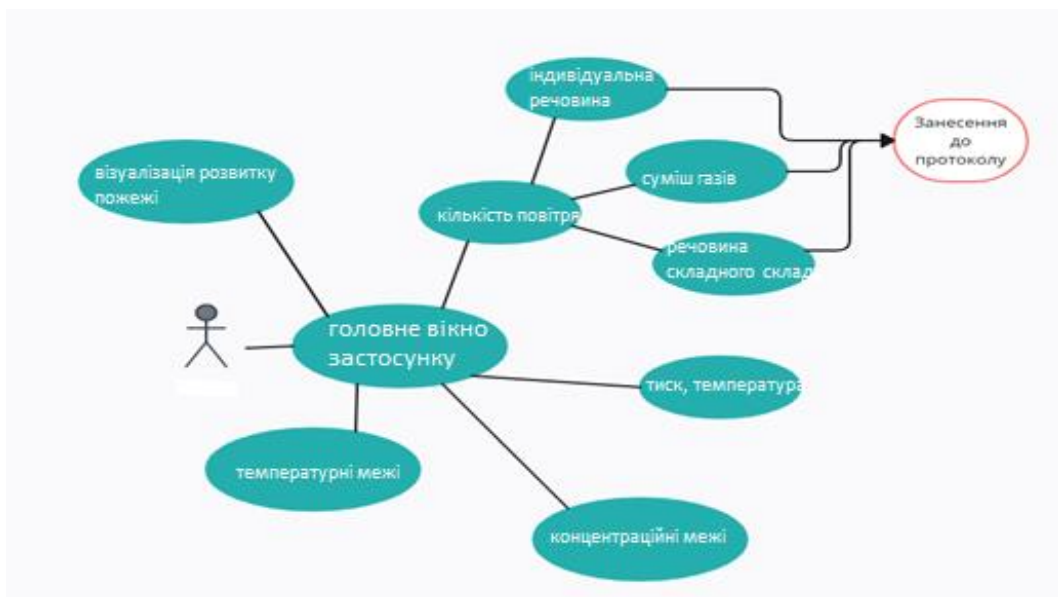


Рисунок 2.4 – Діаграма взаємодії користувача з технологією

3.2 Головне вікно frmContainer

Головне вікно frmContainer визначає центральний вузол інтерфейсу застосунку. Забезпечує навігацію між різними функціональними модулями та виконання основних операцій. Містить панель із зручним доступом до розрахункових вікон та основних функцій програми (рис. 2.5).

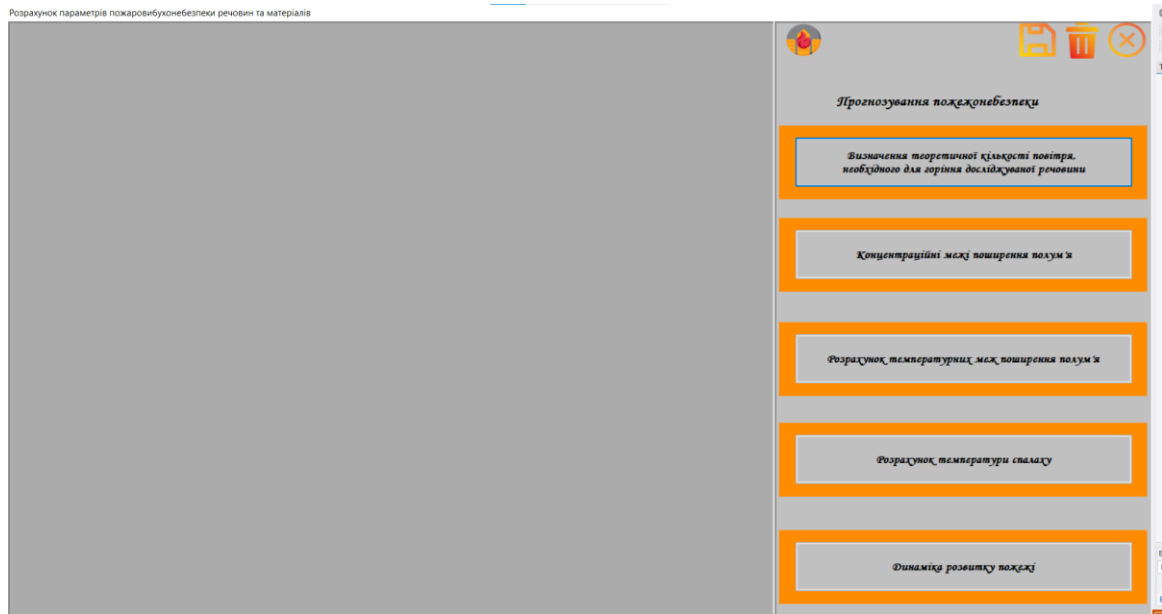


Рисунок 2.5 – Головне вікно застосунку

Кожна з кнопок (рис 2.5) головного вікна викликає відкриття спеціального вікна для проведення розрахунків та аналізу. Кожне з п'яти вікон забезпечує користувачу можливість введення даних для проведення розрахунків. Відображає результати у зручному для розуміння вигляді.

Клас Form1 виступає в ролі координатора та включає в себе функції обробки подій, які визивають інші вікна (дод. А).

3.3 Вікно Air_quantity.

Вікно Air_quantity призначено для розрахунку теоретичної кількості повітря, необхідної для горіння різних речовин. Це вікно є частиною інформаційної технології, яка включає в себе модулі для розрахунків та аналізу та являється дочірнім до головного вікна застосунку (рис 2.6).

Головні елементи вікна включають у себе панель з випадаючим списком `comboBox_fuel`, де користувач може вибрати тип речовини для розрахунку. Після вибору відбувається динамічне створення групи елементів у вікні відповідно до обраного типу речовини.

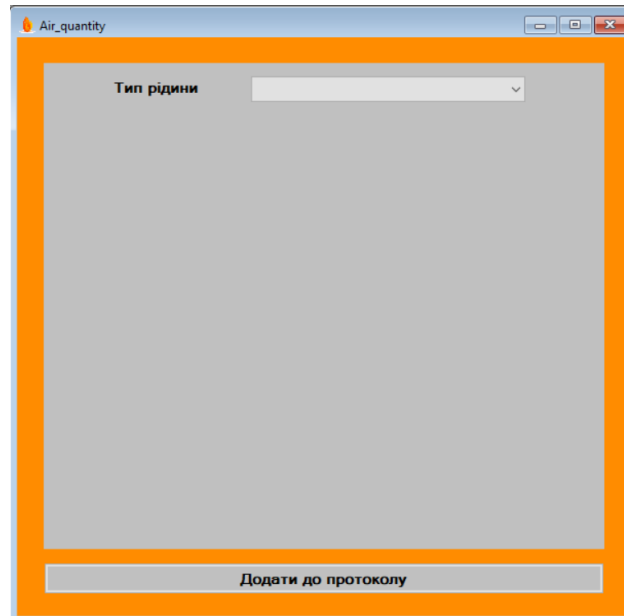


Рисунок 2.6 – Вікно розрахунку теоретичної кількості повітря

Для *Індивідуальної речовини* відображається список доступних горючих рідин, де вибір конкретної речовини викликає розрахунок та відображення результатів щодо кількості кисню, азоту та горючої рідини. Список завантажується з файлу.

Для *Речовини складного складу* доступні текстові поля для введення значень різних компонентів, а також кнопки для розрахунку та збереження результатів.

Суміш газів включає елементи `CheckBox` для вибору окремих газів у суміші.

Кожен розрахунок відбувається при взаємодії з відповідною кнопкою або виборі елемента у випадаючому списку. Результати розрахунків виводяться на вікно у вигляді текстових елементів та лейблів, а також можуть бути додані до протоколу за допомогою відповідної кнопки (див. дод. Б).

Для ефективного розрахунку теоретичної кількості повітря для горіння різних типів речовин була створена бібліотека класів `Exp_Hazard_Library`, яка

включає в себе класи для різних категорій речовин – Hydrocarbons.cs, Organic.cs, MixGaz.cs (дод Б). Ці класи реалізують необхідні методи та алгоритми для виконання розрахунків залежно від вибраного типу речовини.

3.4 Вікно Concent_limits

Вікно Concent_limits, яке є дочірнім до головного вікна застосунку, спроектоване для виконання розрахунків та аналізу концентраційних меж поширення полум'я (рис 2.7). Воно містить три елементи GroupBox, кожен із яких представляє собою окремий розділ для введення та аналізу даних та містить функціональну кнопку для подальшого розрахунку введених даних.

Контролі, що розташовані у кожному з трьох GroupBox дозволяють визначити концентраційні межі поширення полум'я за допомогою апроксимаційної формули, за граничною теплотою згоряння та для суміші горючих речовин. Необхідні методи реалізують спеціальні класи залежно від обраного типу речовини та способу визначення концентраційних меж (дод).

Також вікно містить 3 кнопки RadioButton, призначені для вибору способу розрахунку. Є можливість зберегти отримані результати у файлі.

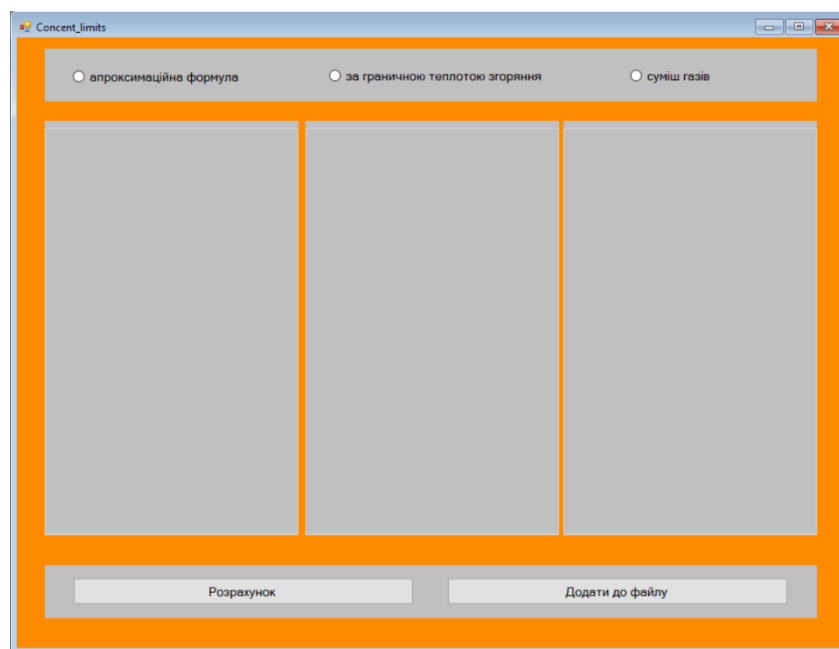


Рисунок 2.7 – Вікно розрахунку теоретичної кількості повітря

3.5 Вікно Explosion

Вікно розроблено для проведення розрахунків теоретичної температури горіння та тиску вибуху горючої рідини в залежності від введених значень температури та тиску навколишнього середовища. Хімічні методи (див. розділ 1.3), використовуються для формування хімічного рівняння реакції горіння, яке визначається спеціальними методами.

Вікно містить текстові віконця для вводу значень температури та тиску навколишнього середовища. На формі також розміщена кнопка розрахунку та виводу отриманих результатів у вікно RichTextBox з подальшою можливістю збереження цих даних у файлі

Це вікно є дочірнім вікном головного вікна та слугує інструментом для проведення розрахунків і визначення ключових параметрів, пов'язаних з горінням та вибухом горючої рідини в заданих умовах.

3.6 Вікно Thermal_intervals

Вікно для визначення температурних меж поширення пламені представляє собою інструмент для розрахунку та аналізу пожежної безпеки рідинних горючих речовин. Це вікно дозволяє користувачеві визначити нижні та верхні температурні межі займання, що є важливими параметрами для оцінки пожежної небезпеки речовин.

Користувач може ввести значення нижньої концентраційної межі поширення полум'я для конкретної речовини, і вікно автоматично розрахує відповідні температурні межі займання. Для цього використовуються формули, що зв'язують тиск насичених парів з температурою.

Вікно також надає можливість користувачеві обирати різні речовини, для яких відбувається розрахунок. Дані результати можуть бути використані для подальших наукових досліджень, проектування систем пожежогасіння та планування заходів з ліквідації пожеж.

3.7 Вікно `Dynamic_fire`

Вікно `Dynamic_fire` відображає графічне та числове представлення динаміки пожежі, що є ключовим етапом у розумінні та управлінні пожежним процесом. Динаміка пожежі визначається залежністю її параметрів від часу.

Для побудови графіків динаміки розвитку пожежі використовується бібліотека `System.Windows.Forms.DataVisualization.Charting`. У цій бібліотеці використовується клас `Chart` та його складові (`Series`, `ChartArea`, `DataPoint`), які дозволяють ефективно створювати та налаштовувати графіки на формі. Метод `ChartDynamicsFire` визначає, які дані відображати на графіку.

Метод `fire_2D_Paint` використовує об'єкт типу `Graphics` для реалізації малюнку форми поширення пожежі в просторі. За допомогою цього методу створюється елемент, який відображає область поширення вогню. Метод використовує `LinearGradientBrush` для створення градієнтного фону. Такий градієнт відображається як плавний перехід від червоного до жовтого кольору. Це може слугувати символічним представленням зони поширення пожежі в просторі. Ширина області поширення пожежі визначається за допомогою спеціальної математичної функції, що робить її динамічною і залежною від часу. Така візуальна репрезентація служить для ілюстрації певних аспектів розвитку пожежі.

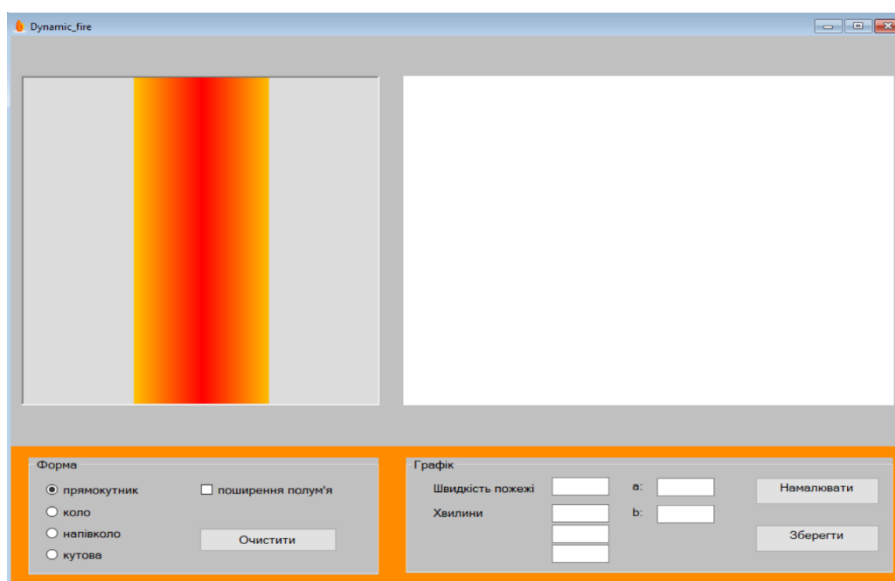


Рисунок 2.8 – Вікно для візуалізації розвитку пожежі

3.8 Бібліотека класів `Exp_Hazard_Library`

Бібліотека класів `Exp_Hazard_Library` містить набір класів, необхідних для роботи з даними, пов'язаними з характеристиками речовин, що можуть брати участь у хімічних реакціях.

Класи `Hydrocarbons.cs`, `Organic.cs`, та `MixGaz.cs` виконують роль моделей для представлення вуглеводнів, складних органічних речовин і змішаних газів відповідно (див. дод. Ж). Кожен з цих класів має специфічні властивості та методи, призначені для розрахунку характеристик саме цього типу речовин.

Класи `Writer.cs` та `Reader.cs` в бібліотеці відповідають за запис і читання даних в/з файлів. Клас `Writer.cs` надає можливість створювати та записувати інформацію у вказаний файл. Зокрема, він використовується для формування протоколів з визначеним ім'ям і додаванням вмісту з інших файлів.

Клас `Reader.cs` служить для завантаження даних з файлів. Цей клас реалізований з можливістю читання файлу і створення об'єктів класів `Hydrocarbons.cs`, `Organic.cs`, та `MixGaz.cs` з отриманих даних.

Обидва класи сприяють зручній взаємодії з даними, які можуть використовуватися для подальших розрахунків та аналізу в контексті вибухової безпеки та інших додаткових досліджень.

4 ТЕСТУВАННЯ РОБОТИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

Головне вікно frmContainer

Процес роботи:

Головне вікно відображається з великою швидкодією та виглядає дуже зручно для користувача.

Забезпечує легкий доступ до всіх інших модулів та вікон, що робить взаємодію з різними функціональними частинами додатку інтуїтивно зрозумілою (див. рис 2.5).

Вікно Air_quantity

Процес роботи:

Вікно взаємодіє із користувачем швидко та ефективно, дозволяючи вводити дані з легкістю.

Після введення значень автоматично виконує розрахунки, і виводить результати відформатовано та зрозуміло для користувача.

Вікно Concent_limits

Процес роботи:

Забезпечує точний та швидкий розрахунок концентраційних меж поширення полум'я за різними формулами.

Під час взаємодії з користувачем, відображає візуальні ефекти та забезпечує зручність у введенні та збереженні результатів.

Вікно Explosion

Процес роботи:

Дозволяє користувачеві ефективно вводити та змінювати параметри для розрахунків теоретичної температури горіння та тиску вибуху.

Графічні елементи відображають динаміку зміни параметрів, що полегшує визначення критичних значень.

Вікно Thermal_intervals

Процес роботи:

Дозволяє вводити дані та отримувати точні температурні інтервали для різних речовин.

Простий та інтуїтивно зрозумілий інтерфейс полегшує взаємодію з користувачем.

Вікно Dynamic_fire

Процес роботи:

Відображає динаміку розвитку пожежі на графіку з використанням високоякісних графічних ефектів.

Дозволяє користувачеві швидко та точно аналізувати дані та реагувати на зміни параметрів.

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи стало успішне досягнення поставленої мети – розробки інструментів для прогнозування пожежонебезпеки. Для цього було виконано комплекс завдань, який включав збір та аналіз початкових даних, створення та розрахунок математичних моделей процесів горіння, розробку та застосування методів прогнозування динаміки розвитку пожежі.

Розроблена інформаційна технологія, яка базується на отриманих математичних моделях, є потужним інструментом для попередження можливих загроз та мінімізації ризиків пожежі. Застосування цього інструменту дозволяє контролювати пожежну небезпеку та вчасно реагувати на потенційні загрози.

Під час аналізу отриманих результатів виявлено, що розроблений інструмент візуалізації надає можливість спостерігати динаміку розвитку пожежі, що полегшує подальший аналіз та прийняття ефективних рішень.

Отже, кваліфікаційна робота успішно реалізує поставлені завдання та створює засіб, який відкриває нові можливості для прогнозування та контролю пожежної небезпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кусковець С.Л., Шаталов О.С., Турченко В.О. Основи теорії горіння та вибуху: Навчальний посібник, – Рівне: НУВГП, 2012. – 374 с.
2. Ємельяненко С.О., Лавренюк О.І., Михалічко Б.М. Теорія горіння та вибуху. Пожежі та їх ліквідація: Навчальний посібник, - Львів: ЛДУ БЖД, 2019. – 158 с.
3. Тарахно О.В., Жернокльов К.В., Трегубов Д.Г. Теорія розвитку та припинення горіння. Методичні вказівки до вивчення курсу, контрольні завдання для курсантів, слухачів денної та заочної форм навчання – Харків: Університет цивільного захисту України, 2011 – 207 с.
4. Рачинська А.Л. Конспект лекцій з дисципліни «Комп'ютерна графіка» для здобувачів вищої освіти першого (бакалаврського) рівня. Електронне видання. – Одеса: ОНУ, 2021.
5. Пічугін М. Ф. Комп'ютерна графіка: навч. посібник / М. Ф. Пічугін, І. О. Канкін, В. В. Воротніков. Київ: Центр учбової літ., 2013. – 346 с.
6. Маценко В. Г. Комп'ютерна графіка: Навчальний посібник. Чернівці: Рута, 2009. - 343 с.
7. Інформаційно-довідковий збірник з пожежної безпеки: Довідник. Під ред. С.І.Білика – Львів: Львівський державний університет безпеки життєдіяльності МНС України, 2013. – 79 с.
8. Гази горючі природні родовищ України для промислового та комунальнопобутового призначення : ТУ У 320.001.58764-033-2000. – К. : Вид-во "Київ", 2000. – 37 с.
9. Довідник пожежного-рятувальника, НДР №0116U002021, Харків: Національний університет цивільного захисту України, 2017 – 114 с.
10. Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming: ISBN-13 (pbk): 978-1-4842-6938-1. – Copyright © 2021 by Andrew Troelsen, Phillip Japikse, - 1383 p.
11. Fundamentals of computer programming with C# (The Bulgarian C# Programming Book) © Svetlin Nakov & Co., 2013 – 1122 p.

ДОДАТОК А. КОД КЛАСУ frmContainer

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Microsoft.DirectX;
using Microsoft.DirectX.Direct3D;

namespace Explosion_hazard_App
{
    public partial class frmContainer : Form
    {
        Device d3d;

        public frmContainer()
        {
            InitializeComponent();
            d3d = null;
            this.SetStyle(ControlStyles.AllPaintingInWmPaint |
ControlStyles.Opaque, true);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Air_quantity formAir = new Air_quantity(this);
            formAir.MdiParent = this;
            formAir.WindowState = FormWindowState.Normal;
            formAir.Show();
        }

        private void DateToRichTextBox()
        {
            string dateTime = DateTime.Now.ToString("dd-MM-yyyy
HH:mm:ss");
            //richTextBox1.AppendText($" {dateTime} Протокол
#" + Environment.NewLine + Environment.NewLine);
        }

        private void pictureBox_close_Click(object sender,
EventArgs e)
        {
            this.Close();
        }
    }
}

```

```

        private void pictureBox_Clear_Click(object sender,
EventArgs e)
        {

            DateToRichTextBox();

        }

        private void panel_grafics_Paint(object sender,
PaintEventArgs e)
        {

            d3d.Clear(ClearFlags.Target | ClearFlags.ZBuffer,
Color.Green, 1.0f, 0);

            //окружность
            float y = 0;
            CustomVertex.PositionColored[] verts = new
CustomVertex.PositionColored[400];
            for (int i = 0; i < 400; i++)
            {

                float x = -Convert.ToSingle(Math.Cos(Math.PI /
200 * i));
                float z = Convert.ToSingle(Math.Sin(Math.PI /
200 * i));

                verts[i].Position = new Vector3(x, y, z);
                verts[i].Color = Color.DarkOrange.ToArgb();

            }
            d3d.BeginScene();
            d3d.VertexFormat =
CustomVertex.PositionColored.Format;
            d3d.DrawUserPrimitives(PrimitiveType.LineStrip, 399,
verts);

            d3d.EndScene();
            d3d.Present();

        }

        private void button_dynamics_Click(object sender,
EventArgs e)
        {

            Dynamic_fire formFire = new Dynamic_fire();
            formFire.MdiParent = this;
            formFire.WindowState = FormWindowState.Normal;
            formFire.Show();

        }

        private void button2_Click(object sender, EventArgs e)
        {

            Concent_limits formFire = new Concent_limits();
            formFire.MdiParent = this;
            formFire.WindowState = FormWindowState.Normal;
            formFire.Show();

        }

```



```
private void button3_Click(object sender, EventArgs e)
{
    Thermal_intervals formFire = new
Thermal_intervals();
    formFire.MdiParent = this;
    formFire.WindowState = FormWindowState.Normal;
    formFire.Show();
}

private void button4_Click(object sender, EventArgs e)
{
    Explosion formFire = new Explosion();
    formFire.MdiParent = this;
    formFire.WindowState = FormWindowState.Normal;
    formFire.Show();
}
}
```

ДОДАТОК Б. КОД КЛАСУ Air_quantity

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using Exp_Hazard_Library;
using Air_q = Exp_Hazard_Library.Air_quantity;
using HC = Exp_Hazard_Library.Hydrocarbons;
using Org = Exp_Hazard_Library.Organic;
using Read = Exp_Hazard_Library.Reader;

namespace Explosion_hazard_App
{
    public partial class Air_quantity : Form
    {
        private frmContainer form1;

        List<HC> list_fuel = new List<HC>();
        private GroupBox dynamicGroupBox;
        private Label dynamicLabel, n_O2, n_N2, n_fuel,
info_label, result_label;
        private TextBox result_textBox;
        private ComboBox comboBox_fuel;
        public Air_quantity(frmContainer form1)
        {
            InitializeComponent();
            this.form1 = form1;
        }

        private void comboBox1_SelectedIndexChanged(object
sender, EventArgs e)
        {
            ClearDynamicControls();
            string selectedItem =
comboBox1.SelectedItem.ToString();
            CreateDynamicControls(selectedItem);
        }

        private void CreateDynamicControls(string selectedItem)
        {
            dynamicGroupBox = new GroupBox();
            dynamicGroupBox.Text = selectedItem;
            dynamicGroupBox.Location = new Point(50, 70);
            dynamicGroupBox.Size = new Size(450, 400);
            switch (selectedItem)
            {
                case "Індивідуальна речовина": Ind_Air(); break;
                case "Речовина складного складу": Dif_Air();
break;

                case "Суміш газів": Mix_Air(); break;
                default: break;
            }
        }
    }
}

```

```

        panell1.Controls.Add(dynamicGroupBox);
    }

private void Ind_Air()
{
    dynamicLabel = new Label();
    dynamicLabel.Text = "Горюча речовина";
    dynamicLabel.Size = new Size(120, 20);
    dynamicLabel.Location = new Point(10, 20);

    comboBox_fuel = new ComboBox();
    Read read = new Read();
    list_fuel =
read.LoadItemsFromFile("Hydrocarbones.txt");
    comboBox_fuel.DisplayMember = "Name";
    comboBox_fuel.Items.AddRange(list_fuel.ToArray());
    comboBox_fuel.Location = new Point(10, 40);
    comboBox_fuel.SelectedIndexChanged +=
ComboBoxFuel_SelectedIndexChanged;

    n_O2 = new Label();
    n_O2.Location = new Point(150, 40);
    n_O2.Size = new Size(240, 20);

    n_N2 = new Label();
    n_N2.Location = new Point(150, 65);
    n_N2.Size = new Size(240, 20);

    n_fuel = new Label();
    n_fuel.Location = new Point(150, 90);
    n_fuel.Size = new Size(240, 20);

    info_label = new Label();
    info_label.Location = new Point(10, 120);
    info_label.Size = new Size(400, 100);

    result_label = new Label();
    result_label.Location = new Point(10, 160);
    result_label.Size = new Size(350, 100);

    dynamicGroupBox.Controls.Add(dynamicLabel);
    dynamicGroupBox.Controls.Add(comboBox_fuel);
    dynamicGroupBox.Controls.Add(n_O2);
    dynamicGroupBox.Controls.Add(n_N2);
    dynamicGroupBox.Controls.Add(n_fuel);
    dynamicGroupBox.Controls.Add(info_label);
    dynamicGroupBox.Controls.Add(result_label);
}

private void Dif_Air()
{
    dynamicLabel = new Label();
    dynamicLabel.Text = "Горюча речовина";
    dynamicLabel.Size = new Size(120, 20);

```

```

dynamicLabel.Location = new Point(10, 20);

string[] controlNames = { "C", "H", "O", "N", "W",
"S", "alfa", "T", "P", "rezult_V" };

foreach (string controlName in controlNames)
{
    Label label = new Label();
    label.Name = "l_" + controlName;
    label.Text = controlName;
    label.Location = new Point(20, 20 + (30 *
Array.IndexOf(controlNames, controlName)));

    TextBox textBox = new TextBox();
    textBox.Name = "text_" + controlName;
    textBox.Location = new Point(120, 20 + (30 *
Array.IndexOf(controlNames, controlName)));
    textBox.KeyPress += textBox_KeyPress;

    dynamicGroupBox.Controls.Add(label);
    dynamicGroupBox.Controls.Add(textBox);
}
Button result_organic = new Button();
result_organic.Text = "Розрахувати";
result_organic.Location = new Point(270, 50);
result_organic.Size = new Size(120, 30);
result_organic.Click += Result_organic_Click;

Button save_organic = new Button();
save_organic.Text = "Зберегти";
save_organic.Location = new Point(270, 100);
save_organic.Size = new Size(120, 30);
save_organic.Click += Save_organic_Click;

result_label = new Label();
result_label.Location = new Point(270, 150);
result_label.Size = new Size(100, 40);
result_label.Text = "ghhjhgj";

result_textBox = new TextBox();
result_textBox.Location = new Point(270, 200);
result_textBox.Size = new Size(120, 40);
result_textBox.Text = "0";
result_textBox.ReadOnly = true;

dynamicGroupBox.Controls.Add(save_organic);
dynamicGroupBox.Controls.Add(result_organic);
dynamicGroupBox.Controls.Add(result_label);
dynamicGroupBox.Controls.Add(result_textBox);
}
private void Mix_Air()

```

```

    {
        string[] controlNames = { "1", "2", "3", "4", "5",
"6", "7", "8", "9", "10" };

        foreach (string controlName in controlNames)
        {
            CheckBox checkBox = new CheckBox();
            checkBox.Name = "check_" + controlName;
            checkBox.Text = controlName;
            checkBox.Location = new Point(20, 20 + (30 *
Array.IndexOf(controlNames, controlName)));
            dynamicGroupBox.Controls.Add(checkBox);
        }
    }

private void button1_Click(object sender, EventArgs e)
{
    //string text = n_N2.Text + " " + n_O2 + " " +
n_fuel;
    //form1.RichTextBox_form(text);
}

private void Save_organic_Click(object sender, EventArgs
e)
{
    throw new NotImplementedException();
}

private void Rezult_organic_Click(object sender,
EventArgs e)
{
    double valueC, valueH, valueO, valueS, valueT,
valueW, valueN, valuealfa, valueP;
    valueC =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_C"]).T
ext);
    valueH =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_H"]).T
ext);
    valueO =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_O"]).T
ext);
    valueN =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_N"]).T
ext);
    valueS =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_S"]).T
ext);

```

```

        valueW =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_W"]).Text);
        valueT =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_T"]).Text);
        valueP =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_P"]).Text);
        valuealfa =
Convert.ToDouble(((TextBox)dynamicGroupBox.Controls["text_alfa"]).Text);
        Org organic = new Org(valueC, valueH, valueO,
valueN, valueW, valuealfa, valueT, valueP, valueS);

        result_textBox.Text =
organic.V_air_diff().ToString();
    }
    private void ClearDynamicControls()
    {
        foreach (Control control in panell.Controls)
        {
            if (control is GroupBox)
            {
                panell.Controls.Remove(control);
                control.Dispose();
            }
        }
    }
    private void textBox_KeyPress(object sender,
KeyPressEventArgs e)
    {
        if (!char.IsDigit(e.KeyChar) && e.KeyChar != 8)
        {
            e.Handled = true;
        }
    }
    private void ComboBoxFuel_SelectedIndexChanged(object
sender, EventArgs e)
    {
        string selectedItem =
comboBox1.SelectedItem.ToString();
        if (comboBox_fuel.SelectedItem != null)
        {
            switch (selectedItem)
            {
                case "Індивідуальна речовина":
                    HC selectedFuel =
(HC)comboBox_fuel.SelectedItem;
                    n_O2.Text = $"Кількість кислороду:
{selectedFuel.n_O2} кмоль";

```

```
        n_N2.Text = $"Кількість азоту:
{selectedFuel.n_N2} кмоль";
        n_fuel.Text = $"Кількість горючої
рідини: {selectedFuel.n_fuel} кмоль";
        info_label.Text = $"Дані отримані при
розрахуванні матеріального \r\nбалансу горіння в повітрі
речовини: {selectedFuel.Name}";
        result_label.Text = $"Теоретична
кількість повітря, \r\n необхідна для горіння 1 м3 горючого за
нормальних умов: {selectedFuel.V_air_ind()} "; break;
        case "Речовина складного складу":
            Dif_Air(); break;
        case "Суміш газів":
            Mix_Air(); break;
        default: break;
    }
}
}
}
```

ДОДАТОК В. КОД КЛАСУ Concent_limits.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Explosion_hazard_App
{
    public partial class Concent_limits : Form
    {
        public Concent_limits()
        {
            InitializeComponent();
        }

        public enum TypeFuel { Individual, Difficult, MixtureGas }
        public TypeFuel typeFuel { get; set; } = TypeFuel.Individual;
        public static double n_O2 { get; set; }
        public static double n_N2 { get; set; }
        public static double n_fuel { get; set; }
        double C, H, S, O, fi_O2;
        public static int n { get; set; }
        double [] fi_fuel, N_O2;

        public double V_air_ind()
        {
            return (n_O2+n_N2)/n_fuel;
        }
        public double V_air_diff()
        {
            return 0.269*(C/3.0 + H + (S-O)/8.0);
        }
        public double V_air_mixgas()
        {
            N_O2 = new double[n];
            fi_fuel = new double[n];

            double sum = 0;
            for (int i = 0; i<n;i++) { sum += fi_fuel[i] * N_O2[i]; }
            return (sum - fi_O2)/21;
        }
    }
}

```



```
public double Calc_V()
{
    switch (typeFuel)
    {
        case TypeFuel.Individual:
            return V_air_ind();
        case TypeFuel.Difficult:
            return V_air_diff();
        case TypeFuel.MixtureGas:
            return V_air_mixgas();

        default: return double.NaN;
    }
}

private void radioButton3_CheckedChanged(object sender,
EventArgs e)
{
    radioButton3.Checked = false;
}

private void Concent_limits_Load(object sender,
EventArgs e)
{
    foreach (Control control in panell1.Controls)
    {
        if (control is RadioButton radioButton)
        {
            radioButton.Checked = false;
        }
    }
}
}
```

ДОДАТОК Г. КОД КЛАСУ Explosion

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace Explosion_hazard_App
{
    public partial class Dynamic_fire : Form
    {
        public Dynamic_fire()
        {
            InitializeComponent();
        }

        public void ChartDynamicsFire(double[]x,
double[]y,string title)
        {
            chart_dynamics.Series[0].Points.Clear();
            chart_dynamics.Titles[0].Text = title;
            chart_dynamics.ChartAreas[0].AxisX.Interval = 1.0;
            chart_dynamics.ChartAreas[0].AxisY.Interval = 1.0;
            Series S1 = new Series();
            int n = x.Length;
            double f_min = double.MaxValue;
            double f_max = double.MinValue;
            for (int i = 0; i < n; i++)
            {
                S1.Points.AddXY(x[i], y[i]);
                f_min = Math.Min(f_min, y[i]);
                f_max = Math.Max(f_max, y[i]);
            }
            S1.ChartType = SeriesChartType.Line;
            S1.MarkerStyle = MarkerStyle.None;
            S1.BorderWidth = 3;
            S1.Color = Color.DarkOrange;
            chart_dynamics.Series[0] = S1;

            chart_dynamics.ChartAreas[0].AxisX.Minimum =
Math.Floor(x[0]);
            chart_dynamics.ChartAreas[0].AxisX.Maximum =
Math.Ceiling(x[n-1]);
            chart_dynamics.ChartAreas[0].AxisY.Minimum =
Math.Floor(f_min);

```

```

        chart_dynamics.ChartAreas[0].AxisY.Maximum =
Math.Ceiling(f_max);
        chart_dynamics.Invalidate();
    }

e)    private void Dynamic_fire_Load(object sender, EventArgs
    {
        double[] x = new double[] { 1, 5, 9, 12 };
        double[] y = new double[] { 10, 15, 9, 22 };
        ChartDynamicsFire(x, y, "Grafik");
    }

e)    private void fire_2D_Paint(object sender, PaintEventArgs
    {
        Graphics g = e.Graphics;
        Point p1 = new Point();
        p1.X = 0;
        p1.Y = 0;
        Point p2 = new Point();
        p2.X = 210;
        p2.Y = 150;

        LinearGradientBrush MyB = new
LinearGradientBrush(p1, p2, color1: Color.Red, color2:
Color.DarkBlue);
        g.FillRectangle(MyB, 10, 50, 200, 100);
        MyB.Dispose();
        g.Dispose();
    }
}

```

ДОДАТОК Д. КОД КЛАСУ `Thermal_intervals`

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Explosion_hazard_App
{
    public partial class Thermal_intervals : Form
    {
        public Thermal_intervals()
        {
            InitializeComponent();
        }
        public double V_air_ind()
        {
            return (n_O2+n_N2)/n_fuel;
        }
        public double V_air_diff()
        {
            return 0.269*(C/3.0 + H + (S-O)/8.0);
        }
        public double V_air_mixgas()
        {
            N_O2 = new double[n];
            fi_fuel = new double[n];

            double sum = 0;
            for (int i = 0; i<n;i++) { sum += fi_fuel[i] * N_O2[i]; }
            return (sum - fi_O2)/21;
        }

        public double Calc_V()
        {
            switch (typeFuel)
            {
                case TypeFuel.Individual:
                    return V_air_ind();
                case TypeFuel.Difficult:
                    return V_air_diff();
                case TypeFuel.MixtureGas:
                    return V_air_mixgas();

                default: return double.NaN;
            }
        }
    }
}

```

ДОДАТОК Е. КОД КЛАСУ Dynamic_fire

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace Explosion_hazard_App
{
    public partial class Dynamic_fire : Form
    {
        Graphics g = null;
        private float rectangleWidth;
        Point pColor1, pColor2;
        private float radius = 100.0f;

        public Dynamic_fire()
        {
            InitializeComponent();
            rectangleWidth = fire_2D.Width / 2.0f;
            rb_circle.Checked = false;
            rb_circle12.Checked = false;
            rb_angle.Checked = false;
        }

        public void ChartDynamicsFire(double[]x, double[]y,
double[] x2, double[] y2, string title)
        {
            chart_dynamics.Series[0].Points.Clear();
            chart_dynamics.Titles[0].Text = title;
            chart_dynamics.ChartAreas[0].AxisX.Interval = 1.0;
            chart_dynamics.ChartAreas[0].AxisY.Interval = 1.0;
            Series S1 = new Series();
            int n = x.Length;
            double f_min = double.MaxValue;
            double f_max = double.MinValue;
            for (int i = 0; i < n; i++)
            {
                S1.Points.AddXY(x[i], y[i]);
                f_min = Math.Min(f_min, y[i]);
                f_max = Math.Max(f_max, y[i]);
            }
            S1.ChartType = SeriesChartType.Line;
            S1.MarkerStyle = MarkerStyle.None;
        }
    }
}

```

```

S1.BorderWidth = 3;
S1.Color = Color.DarkOrange;
S1.LegendText = "Площа";
chart_dynamics.Series[0] = S1;

Series series2 = new Series();
int n2 = x2.Length;
double f_min2 = double.MaxValue;
double f_max2 = double.MinValue;
for (int i = 0; i < n2; i++)
{
    series2.Points.AddXY(x2[i], y2[i]);
    f_min2 = Math.Min(f_min2, y2[i]);
    f_max2 = Math.Max(f_max2, y2[i]);
}
series2.ChartType = SeriesChartType.Line;
series2.MarkerStyle = MarkerStyle.None;
series2.BorderWidth = 3;
series2.Color = Color.Blue;
series2.LegendText = "Фронт";
chart_dynamics.Series.Add(series2);

chart_dynamics.ChartAreas[0].AxisX.Minimum =
Math.Floor(x[0]);
chart_dynamics.ChartAreas[0].AxisX.Maximum =
Math.Ceiling(x[n-1]);
chart_dynamics.ChartAreas[0].AxisY.Minimum =
Math.Floor(f_min);
chart_dynamics.ChartAreas[0].AxisY.Maximum =
Math.Ceiling(f_max);
chart_dynamics.Invalidate();
}

private void Dynamic_fire_Load(object sender, EventArgs
e)
{
}

private void fire_2D_Paint(object sender, PaintEventArgs
e)
{
    Picture_fire(e.Graphics);
}

public void Picture_fire(Graphics graphics)
{
    using (Graphics g = graphics)
    {
        Point center = new Point(fire_2D.Width / 2,
fire_2D.Height / 2);

        Color color1 = Color.Red;

```

```

Color color2 = Color.Yellow;
Point absolutePColor1;
Point absolutePColor2;

if (rb_rect.Checked)
{
    pColor1 = new Point(fire_2D.Width / 4, 0);
    pColor2 = new Point(3 * fire_2D.Width / 4,
0);
    absolutePColor1 =
fire_2D.PointToScreen(pColor1);
    absolutePColor2 =
fire_2D.PointToScreen(pColor2);
    using (LinearGradientBrush gradientBrush =
new LinearGradientBrush(pColor1, pColor2, color1, color2))
    {
        ColorBlend colorBlend = new
ColorBlend(3);
        colorBlend.Colors = new Color[] {
color2, color1, color2 };
        colorBlend.Positions = new float[] {
0.0f, 0.5f, 1.0f };
        gradientBrush.InterpolationColors =
colorBlend;
        g.FillRectangle(gradientBrush,
absolutePColor1.X, 0, rectangleWidth, fire_2D.Height);
    }
}
else if (rb_circle.Checked)
{
    int centerX = fire_2D.Width / 2;
    int centerY = fire_2D.Height / 2;
    GraphicsPath path = new GraphicsPath();
    path.AddEllipse(centerX - radius, centerY -
radius, 2*radius, 2*radius);
    using (PathGradientBrush gradientBrush = new
PathGradientBrush(path))
    {
        gradientBrush.CenterPoint = new
PointF(centerX, centerY);
        gradientBrush.CenterColor = Color.Red;
        gradientBrush.SurroundColors = new
Color[] { Color.Yellow };
        g.FillPath(gradientBrush, path);
    }
}
else if (rb_circle12.Checked)
{
    int centerX = 0;
    int centerY = fire_2D.Height / 2;
    GraphicsPath path = new GraphicsPath();

```

```

        path.AddEllipse(0 - radius, center.Y -
radius, 2 * radius, 2 * radius);
        using (PathGradientBrush gradientBrush = new
PathGradientBrush(path))
        {
            gradientBrush.CenterPoint = new
PointF(centerX, centerY);
            gradientBrush.CenterColor = Color.Red;
            gradientBrush.SurroundColors = new
Color[] { Color.Yellow };
            g.FillPath(gradientBrush, path);
        }
    }
else
{
    int centerX = 0;
    int centerY = fire_2D.Height;

    int startAngle = 270;
    int sweepAngle = 90;
    GraphicsPath path = new GraphicsPath();
    path.AddPie(0 - radius, fire_2D.Height -
radius, 2 * radius, 2 * radius, startAngle, sweepAngle);
    using (PathGradientBrush gradientBrush = new
PathGradientBrush(path))
    {
        gradientBrush.CenterPoint = new
PointF(centerX, centerY);
        gradientBrush.CenterColor = Color.Red;
        gradientBrush.SurroundColors = new
Color[] { Color.Yellow };
        g.FillPath(gradientBrush, path);
    }
}
}

private void rb_rect_CheckedChanged(object sender,
EventArgs e)
{
    fire_2D.Invalidate();
}

private void checkBox1_CheckedChanged(object sender,
EventArgs e)
{
    if (checkBox1.Checked)
        timer1.Enabled = true;
    else
        timer1.Enabled = false;
}

```



```

        private void rb_circle_CheckedChanged(object sender,
EventArgs e)
        {
            fire_2D.Invalidate();
        }

        private void rb_circle12_CheckedChanged(object sender,
EventArgs e)
        {
            fire_2D.Invalidate();
        }

        private void rb_angle_CheckedChanged(object sender,
EventArgs e)
        {
            fire_2D.Invalidate();
        }

        private void button_draw_Click(object sender, EventArgs
e)
        {
            textBox1.Text = 0.75f.ToString();
            double[] x = new double[] { 5, 15, 20};
            double[] y = new double[] { 5.7, 16.4, 27.7 };
            double[] x1 = new double[] { 5, 15, 20 };
            double[] y1 = new double[] { 6, 9.2, 13.2 };
            ChartDynamicsFire(x, y, x1, y1, "Площа та фронт
пожежі");

            g = fire_2D.CreateGraphics();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            rectangleWidth = fire_2D.Width / 2.0f;
            radius = 100.0f;
            fire_2D.Invalidate();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            if (rb_rect.Checked)
            {
                rectangleWidth += 2.0f;
                pColor1.X -= 1;
                pColor2.X += 1;
            }
            else
            {
                radius += 3;
            }
            fire_2D.Invalidate();
        } } }

```

ДОДАТОК Ж. БІБЛІОТЕКА КЛАСІВ Exp_Hazard_Library

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Exp_Hazard_Library
{
    public class Hydrocarbons
    {
        public string Name { get; set; }
        public double n_O2 { get; set; }
        public double n_N2 { get; set; }
        public double n_fuel { get; set; }
        public double M_fuel { get; set; }

        public Hydrocarbons(string Name, double n_O2, double
n_N2, double n_fuel, double M_fuel)
        {
            this.Name = Name;
            this.n_N2 = n_N2;
            this.n_O2 = n_O2;
            this.n_fuel = n_fuel;
            this.M_fuel = M_fuel;
        }
        public double V_air_ind()
        {
            return (n_O2 + n_N2)*22.4 / n_fuel/M_fuel;
        }
    }
    public class Organic
    {
        public double C { get; set; }
        public double H { get; set; }
        public double O { get; set; }
        public double N { get; set; }
        public double W { get; set; }
        public double S { get; set; }
        public double alfa { get; set; }
        public double T { get; set; }
        public double P { get; set; }

        public Organic()
        { }
        public Organic(double C, double H, double O, double N,
double W, double alfa, double T, double P,
double S)
        {
            this.C = C;
            this.H = H;
            this.O = O;

```

```

        this.N = N;
        this.W = W;
        this.S = S;

        this.alfa = alfa;
        this.T = T;
        this.P = P;
    }
    public double V_air_diff()
    {
        return 0.269 * (C / 3.0 + H + (S - O) / 8.0);
    }
}

public class MixGas
{
}
public class Reader
{
    public List<Hydrocarbons> LoadItemsFromFile(string filePath)
    {
        List<Hydrocarbons> items = new List<Hydrocarbons>();
        try
        {
            if (File.Exists(filePath))
            {
                string[] lines = File.ReadAllLines(filePath);

                foreach (string line in lines)
                {
                    char[] separator = new char[] { ' ', ',', '.' };
                    string[] parts = line.Split(separator,
StringSplitOptions.RemoveEmptyEntries);
                    double a, b, c, d;
                    string text = parts[0].Trim();
                    a= double.Parse(parts[1]);
                    b= double.Parse(parts[2]);
                    c= double.Parse(parts[3]);
                    d= double.Parse(parts[4]);
                    Hydrocarbons hc = new Hydrocarbons(text, a,
b, c, d);

                    items.Add(hc);
                }
            }
            else
            {
                Console.WriteLine($"Файл {filePath} не існує.");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Помилка при завантаженні з
файлу: {ex.Message}");
        }
    }
}

```

```

    }
    return items;
}
}
public class Writer
{
    public static StreamWriter ResultWriter(string name, params
string[] files)
    {
        string projectDir =
Directory.GetParent(Directory.GetCurrentDirectory()).Parent.Full
Name;

        string file_name = name + " Operator 1.txt";

        StreamWriter writer = new StreamWriter(file_name);
        foreach (string file in files)
        {
            writer.WriteLine(file + ":\r\n");
            writer.Flush();
            using (StreamReader reader = new
StreamReader(projectDir + '\\ ' + file))
                reader.BaseStream.CopyTo(writer.BaseStream);
            writer.WriteLine("\r\n// -----
----- //\r\n");
        }
        writer.Write(file_name);
        writer.WriteLine($" data :
{DateTime.Now.ToShortDateString()} " +
            $" time :
{DateTime.Now.ToShortTimeString()}\r\n");
        return writer;
    }
}
}
}

```

ДОДАТОК 3. ДОВІДКА ПРО ВПРОВАДЖЕННЯ