

АНОТАЦІЯ

наукової роботи під шифром «Пожежний ризик»

Автозаправні станції (АЗС) є об'єктами підвищеної пожежовибухонебезпеки. У зв'язку з цим можливі аварії на АЗС становлять серйозну небезпеку для населення і навколишніх об'єктів. Отже, проблема оцінки пожежного ризику з урахуванням впливу навколишнього середовища є актуальною і потребує вирішення, в тому числі через розробку програмного продукту.

Метою роботи є розробка концептуальних положень щодо оцінювання пожежного ризику промислових об'єктів з метою своєчасної реалізації заходів щодо зниження його рівня через створення відповідного програмного забезпечення.

Відповідно до мети поставлено такі завдання:

- дослідити методи та моделі оцінки пожежного ризику та зниження його величини;
- сформулювати концептуальні підходи до оцінювання пожежного ризику;
- спроектувати та розробити програмне забезпечення для оцінювання пожежного ризику на прикладі автозаправної станції.

Об'єкт дослідження – процес управління пожежним ризиком техногенного об'єкту. Предмет дослідження – математичні методи та моделі оцінювання пожежного ризику.

Методи дослідження. Теоретичну та методологічну основу дослідження становили праці вітчизняних та зарубіжних вчених у сфері оцінювання пожежної безпеки промислових об'єктів.

У роботі використовується комплекс загальнонаукових методів: аналізу – для аналізу законодавчої, нормативно-технічної та спеціальної літератури в галузі пожежної безпеки, оцінки ризиків на станціях технічних об'єктах; теоретичне узагальнення – при огляді методів розрахунку пожежного ризику (; системного аналізу – при розробці концепції оцінки індивідуального

пожежного ризику для промислових об'єктів; економіко-статистичний метод – при визначенні сценаріїв розвитку пожежі, при яких реалізуються найгірші умови забезпечення безпеки людей; теорії прийняття рішення – для визначення протипожежних заходів, спрямованих на зниження величини пожежного ризику; візуального моделювання – при проектуванні інформаційної системи для оцінки пожежного ризику; економіко-математичний метод – при визначенні економічної ефективності впровадження програмного забезпечення для оцінювання пожежного ризику автозаправної станції; абстрактно-логічний метод – для теоретичного узагальнення та формулювання висновків.

Наукова новизна одержаних результатів полягає в тому, що:

удосконалено:

– метод визначення оцінки індивідуального пожежного ризику на основі генерації сценаріїв розвитку пожежі в залежності від умов зовнішнього та внутрішнього середовища та часових характеристик розвитку пожежі;

набуло подальшого розвитку:

– підхід до оцінки індивідуального пожежного ризику через зіставлення розрахункової величини пожежного ризику і його нормативної величини з урахування величини часу евакуації, визначеної методом імітаційного моделювання;

– метод вибору додаткових протипожежних заходів щодо зниження пожежного ризику на автозаправній станції через використання динамічного програмування;

– метод прогнозу можливих наслідків пожежі на автозаправній станції за допомогою розрахунку кількості рідино-моторного палива (РМП), розмірів зон вибухонебезпечних концентрацій, розмірів зон ураження при пожежі-спалаху, інтенсивності та дози теплового випромінювання в залежності від сценарію розвитку пожежі.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ОЦІНЮВАННЯ ПОЖЕЖНОГО РИЗИКУ ТЕХНОГЕННИХ ОБ'ЄКТІВ	6
1.1 Поняття пожежного ризику	6
1.2 Аналіз методів та моделей оцінювання пожежного ризику	9
2 РОЗРОБКА КОНЦЕПТУАЛЬНИХ ПІДХОДІВ ДО ОЦІНЮВАННЯ ПОЖЕЖНОГО РИЗИКУ	12
2.1 Концептуальні підходи до оцінювання пожежного ризику	12
2.2 Розробка об'єктно-орієнтованої моделі програмного забезпечення для оцінювання пожежного ризику	17
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ПОЖЕЖНОГО РИЗИКУ	22
ВИСНОВКИ	30
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	31
ДОДАТОК А. Математична модель визначення рівня пожежного ризику	34
ДОДАТОК Б. Об'єктно-орієнтована модель програмного забезпечення	46
ДОДАТОК В. Лістинг програми	51
ДОДАТОК Г. Перелік наукових праць автора	78

ВСТУП

Розвиток і інтенсифікація промислових виробництв в сучасних умовах неминує веде до зростання числа аварій і масштабів наслідків, пов'язаних з неконтрольованим викидом токсичних або вибухонебезпечних речовин в атмосферу. Аварії на таких підприємствах є небезпечними, що характеризуються великим радіусом ураження і тяжкістю своїх наслідків. За статистикою аварії на промисловому підприємстві найчастіше відбуваються через помилкові дії персоналу (38%), розгерметизацію (розрив) сховища (37%), відмови обладнання (21%), зовнішні впливи природного і техногенного характеру (4%).

Запобігання пожеж і вибухів є найважливішою складовою частиною загальної проблеми забезпечення пожежовибухобезпеки різних об'єктів, і тому їй приділяється першочергова увага при вирішенні питань захисту об'єктів при аварії. Рішення яких в свою чергу передбачає оцінку пожежовибухонебезпечних виробництв і призначення різних заходів організаційного та технічного характеру, які регламентуються різними нормативними документами

Науковими розробками в сфері оцінювання пожежної безпеки промислових підприємств займалися Абаєв О., Вяльцев О., Коряковцев Ю., Артамонов В., Брушлинский Н., Євграфов В., Копилова Н., Малигін В., Таранцев А., Чуприян А., Шаровар Ф., Гуліда Е., Коваль О., Паснак І., Шарий В. Оцінка і обґрунтування безпечної експлуатації автозаправної станції відображується в роботах: Ахмерова В., Івасенка В., Саркісова В., Ванчухіна П., Соколової К. та інш.

Однак залишаються невирішеними питання врахування впливу зовнішнього середовища на розвиток пожежі, визначення законів розподілу випадкової величини часу, що враховується при оцінюванні пожежного ризику.

Метою роботи є розробка концептуальних положень щодо оцінювання пожежного ризику промислових об'єктів з метою своєчасної реалізації заходів

щодо зниження його рівня через стровення відповідного програмного забезпечення.

Відповідно до мети поставлено такі завдання:

- дослідити методи та моделі оцінки пожежного ризику та зниження його величини;
- сформулювати концептуальні підходи до оцінювання пожежного ризику;
- спроектувати та розробити програмне забезпечення для оцінювання пожежного ризику на прикладі автозаправної станції.

Об'єкт дослідження – процес управління пожежним ризиком техногенного об'єкту.

Предмет дослідження – математичні методи та моделі оцінювання пожежного ризику.

1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ОЦІНЮВАННЯ ПОЖЕЖНОГО РИЗИКУ ТЕХНОГЕННИХ ОБ'ЄКТІВ

1.1 Поняття пожежного ризику

Аналіз існуючих математичних визначень ризику показав, що, незважаючи на інтенсивний розвиток в останнє десятиліття методів аналізу і оцінки ризику, в Україні досі існує значна невизначеність щодо терміну «ризик». Відсутня загальновизнана система термінів в області теорії ризику, що знайшло відображення як у наукових працях, так і в деяких нормативних актах.

У різних джерелах поняття «ризик» пояснюється як небезпека і як ймовірність несприятливої події, і як очікуваний збиток від несприятливої події, і як векторна величина, складовими якої є ймовірність несприятливої події і очікуваний збиток від неї. Головною проблемою наукового визначення поняття «ризик» є його комплексний характер, що вимагає якнайповнішого врахування всіх факторів небезпеки, що призводять до виникнення надзвичайних ситуацій, а також врахування їх наслідків [1].

Серед чинних наразі нормативно-правових документів можна виділити постанову Кабінету Міністрів України від 29 лютого 2012 р. №306 «Про затвердження критеріїв, за якими оцінюється ступінь ризику від провадження господарської діяльності та визначається періодичність здійснення планових заходів державного нагляду (контролю) у сфері техногенної та пожежної безпеки» (зі змінами) [2].

Відповідно до п. 6 цього документа, «планові заходи державного нагляду (контролю) за додержанням суб'єктами господарювання законодавства у сфері техногенної та пожежної безпеки» мають здійснюватись з періодичністю не частіше ніж один раз на рік, один раз на три роки і один раз на п'ять років залежно від того, до якого з названих вище ступенів ризику органи ДСНС відносять конкретний об'єкт.

Відповідно до постанови КМУ №306 від 29.02.2012 «Про затвердження критеріїв, за якими оцінюється ступінь ризику від впровадження господарської

діяльності та визначається періодичність здійснення планових заходів державного нагляду (контролю) у сфері техногенної та пожежної безпеки» [2]: «Техногенний ризик – це узагальнена характеристика можливості реалізації небезпеки в техногенній сфері, яка визначається через ймовірність виникнення техногенної аварії або катастрофи і математичне очікування негативних наслідків від них.

При визначенні показників техногенного ризику використовують критерії міцності, ресурсу, надійності, живучості, а також дані про збитки – люди, об'єкти техносфери і навколишнє середовище.

Джерелами техногенних ризиків прийнято називати різні небезпеки, що призводять до нештатного функціонування технічних систем або до помилок операторів. Розрізняють внутрішні і зовнішні джерела для кожного технічного пристрою і кожної технічної системи [3].

Зважаючи на причини виникнення техногенного ризику можна вважати, що різновидом техногенного ризику є пожежний ризик.

Основним нормативно-правовим актом в Україні, який регулює відносини у сфері пожежної безпеки, є НАПБ А.01.001-2014 Правила пожежної безпеки в Україні [4].

Відповідно до «ДСТУ 2272:2006 ССБТ. Пожежна безпека. Терміни та визначення» дано таке визначення: «Пожежний ризик – міра можливості реалізації пожежної небезпеки об'єкта захисту та її наслідків для людей і матеріальних цінностей.

Допустимий пожежний ризик – пожежний ризик, рівень якого допустимий і обґрунтований виходячи з соціально-економічних умов.

Індивідуальний пожежний ризик – пожежний ризик, який може призвести до загибелі людини в результаті впливу небезпечних факторів пожежі» [5].

Згідно з НАПБ Б.03.002-2007 [6] приміщення та будівлі в цілому діляться за ступенем пожежо- або вибухонебезпечності на п'ять категорій.

Пожежонебезпечна зона – це простір у приміщенні або за його межами, в якому постійно або періодично знаходяться (зберігаються, використовуються

або виділяються під час технологічного процесу) горючі речовини як при нормальному технологічному процесі, так і при його порушенні в такій кількості, яка вимагає спеціалізованих заходів у конструкції електрообладнання під час його монтажу та експлуатації.

Одним з пожежонебезпечних об'єктів є АЗС.

Автозаправні станції – це комплекс будівель і споруд з обладнанням, призначеним для прийому, зберігання та видачі нафтопродуктів транспортним засобам.

На АЗС також організовують продаж масел, консистентних мастил, запасних частин, аксесуарів до автомобілів та інших транспортних засобів, приймання від власників індивідуального транспорту відпрацьованих масел і дрібної тари з-під нафтопродуктів, технічне обслуговування. Крім того, на АЗС надають сервісні послуги з обслуговування автотранспорту, його власників і пасажирів.

Технологічна схема АЗС складається з трьох стадій:

- прийому нафтопродуктів з бензовозів в підземні резервуари;
- зберігання нафтопродуктів в резервуарах до моменту їх перекачування через паливороздавальні колонки для заправки автотранспортної техніки;
- заправки нафтопродуктами з підземних резервуарів автотранспортної техніки через паливороздавальні колонки.

Специфічною особливістю АЗС є розміщення технологічного обладнання на відкритих майданчиках. При подібному розміщенні виділяються горючі і токсичні пари розсіюються природними повітряними потоками, причому їх концентрація в подальшому знижується до безпечного рівня. Вибухи і пожежі на зовнішніх установках АЗС можливі тільки при аварійних ситуаціях, пов'язаних з освітою вибухонебезпечних концентрацій парів нафтопродуктів в повітряному середовищі [7].

Пожежний ризик, по-перше, характеризують можливістю реалізації пожежної небезпеки у вигляді пожежі та, по-друге, у вигляді оцінки його можливих наслідків (а також обставин, що сприяють розвитку пожежі).

Отже, при їх визначенні необхідно знати частотні характеристики виникнення пожежі на об'єкті, а також передбачувані обсяги його соціальних, економічних і екологічних наслідків, обумовлених тими чи іншими обставинами.

1.2 Аналіз методів та моделей оцінювання пожежного ризику

Кількісним виразом індивідуального пожежного ризику є частота дії небезпечних чинників пожежі (НЧП) на людину, що знаходиться в будівлі. Перелік НЧП і нормативні (граничні) значення пожежного ризику встановлені «Технічним регламентом про вимоги пожежної безпеки». В основі розрахунку пожежних ризиків лежить принцип зіставлення нормативної величини пожежного ризику і розрахункової величини пожежного ризику [8].

Опираючись на джерело [9] при виборі методу оцінки пожежного ризику для конкретного застосування необхідно враховувати наступні фактори: цілі зацікавлених осіб і критерії допустимості; зміст оцінки пожежного ризику; цільову аудиторію і осіб, які приймають рішення; нормативні та / або судові питання; прецеденти подібного застосування; наявні ресурси і дані; обмеження за часом і витратами; кваліфікацію персоналу і потенційну необхідність врахування невизначеностей.

На сьогоднішній день відповідно до закону України «Про основні засади державного нагляду (контролю) за господарською діяльністю» від 2007 року [10] державний пожежний нагляд у сфері господарської діяльності та терміни здійснення планових заходів мають бути диференційовані залежно від ступеня ризику, оцінка якого має ґрунтуватися на оцінці пожежних ризиків. Нормування значень пожежних ризиків передбачає наявність надійної методики їх кількісного оцінювання.

Категорії методів оцінки пожежного ризику наведені в таблиці 1.1.

Таблиця 1.1 – Категорії методів оцінки пожежного ризику

Категорія	Визначення	Тип вихідних даних	Приклад
1. Якісний метод	Оцінює як можливість, так і наслідки в якісних показниках.	Табличні дані про результати і відносну можливість різних сценаріїв пожежі, а також вплив на них різних варіантів протипожежного захисту.	Аналіз «що якщо ...?» Матриці ризиків. Бальна оцінка ризиків. Дерево концепцій пожежної безпеки.
2. Напівкількісний метод оцінки можливості	Оцінює можливість в кількісних показниках, а наслідки – в якісних.	Визначення частоти виникнення різних типів пожеж та / або в залежності від різних типів захисту.	Страховий аналіз /статистичний аналіз збитків. Аналіз дерева окремого події.
3. Напівкількісний метод оцінки наслідків	Оцінює наслідки в кількісних показниках, а можливість - в якісних.	Вихідні дані детермінованої моделі пожежі з обробкою можливості в якісних показниках.	Моделі пожежі в приміщеннях для обраних складних сценаріїв.
4. Кількісний метод	Об'єднує кількісну оцінку ймовірності та наслідків.	(1) визначення очікуваного збитку; або (2) визначення ймовірності спалаху; або (3) визначення ймовірності смертей в інших приміщеннях або поверхах будівлі; або (4) діаграма частоти щодо кількості смертельних випадків; або (5) діаграма частоти щодо величини збитку. Або (6) визначення можливості травм, смертельних випадків збитку майна і переривання комерційної діяльності. Або (7) визначення індивідуального ризику (користувачам будівлі) та соціального ризику.	Методи оцінки пожежного ризику для визначення можливості плавлення активної зони ядерного реактора. Аналіз дерева подій в поєднанні з моделями пожежі.
5. Техніко-економічні методи оцінки ризику	Включають визначення витрат на альтернативні підходи по обмеження наслідків і / або можливостей	(1) визначення витрат, необхідних для досягнення різних рівнів зниження ризику; або (2) визначення «оптимального» рівня протипожежного захисту, заснованого на зведенні «загального ризику» до мінімуму або на інший показник ризику	Обчислювальні моделі, що включають в себе комплексні дані про ймовірності, наслідки і витрати

Вище наведені методи оцінки пожежного ризику реалізовані в декількох програмах, які дозволяють проектувати можливий розвиток аварій при різноманітних ситуаціях:

1. Fire Dynamics Simulator (FDS). Польова модель FDS може передбачати поширення диму, температури, чадного газу та інших небезпечних факторів під час пожежі. Результати моделювання використовуються для забезпечення безпеки будівель при проектуванні, визначення безпеки існуючих будівель, реконструкції пожеж при розслідуваннях і допомоги в тренуваннях пожежників [11].

2. Pathfinder – програма, яка реалізує індивідуально-потоківу модель руху людей при евакуації. Вона має графічний інтерфейс для завдання вихідних даних, а також інструменти для 2D і 3D-візуалізації результатів. Крім візуалізації Pathfinder також створює вихідні дані у вигляді графіків в форматі CSV (значення, розділені комами) і текстового звіту, в якому міститься час виходу з приміщень і швидкості потоків через двері [12].

3. Програми «СІТІС: Флоутек», «СІТІС: Блок», «СІТІС: ВІМ» і «СІТІС: Фламмер» реалізують моделі розрахунку руху при евакуації і динаміки небезпечних факторів пожежі і дозволяють виконати розрахунок часу блокування шляхів евакуації небезпечними факторами пожежі і розрахунковий час евакуації відповідно. Програма «СІТІС: Спринт» дозволяє визначити розрахункову величину індивідуального пожежного ризику на підставі даних моделювання [13].

4. Програма «Fenix +» призначена для визначення величини індивідуального пожежного ризику в будівлях, спорудах і будівлях різних класів функціональної пожежної безпеки [14].

У наведених програмах відсутні можливості самостійного генерування сценаріїв розвитку пожежі, відстеження динаміки раніше отриманих значень пожежного ризику, не передбачений ввід користувачем часових характеристик, в разі отримання недопустимого значення пожежного ризику не передбачені заходи щодо його зниження.

2 ПРОЕКТУВАННЯ ПРОГРАММНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ПОЖЕЖНОГО РИЗИКУ

2.1 Концептуальні підходи до оцінювання пожежного ризику

Своєчасність і безперешкодність евакуації – два основних критеріїв безпеки людей під час пожежі. Проектування будівель і споруд має бути направлено на створення максимально можливих сприятливих умов для забезпечення безпеки людини при евакуації. Такі вимоги містять НАПБ А.01.001-2014 [4] та ДБН В.1.2-7-2008 [15], ГОСТ 12.1.004-91 [16]. Їх практична реалізація вимагає обґрунтованих даних, отриманих в результаті натурних спостережень і теоретичних досліджень. Однак така найважливіша складова загального часу евакуації t_e , як час початку евакуації t_{ne} , значення якого може перевищувати розрахунковий час евакуації t_p , не має достатньої статистичної бази результатів натурних спостережень та теоретичних досліджень його формування.

Згідно з наведеними нормативними актами про безпеку людей можна говорити тільки в тому випадку, коли ймовірність евакуації P_E , що розраховується за формулою (2.1), становить не менше 0,999.

$$P_{Ei} = \begin{cases} \frac{0,8 \cdot t_{\text{бл}}}{t_p}, & \text{якщо } t_p < 0,8 \cdot t_{\text{бл}} < t_p + t_{ne} \text{ та } t_{ck} \leq 6 \text{ хв} \\ 0,999, & \text{якщо } t_p + t_{ne} \leq 0,8 \cdot t_{\text{бл}} \text{ та } t_{ck} \leq 6 \text{ хв} \\ 0,000, & \text{якщо } t_p \geq 0,8 \cdot t_{\text{бл}} \text{ або } t_{ck} > 6 \text{ хв} \end{cases}, \quad (2.1)$$

де t_{ne} – час початку евакуації, хв.; t_p – розрахунковий час евакуації людей, хв.; $t_{\text{бл}}$ – час блокування шляхів евакуації, хв.; t_{ck} – час існування скупчень людей на ділянках шляху (щільність людського потоку на шляхах евакуації перевищує значення 0,5).

Як відомо, розрахунковий час евакуації людей t_p з приміщень і будівель встановлюється за часом руху одного або декількох людських потоків через евакуаційні виходи від найбільш віддалених місць розміщення людей.

Закономірності руху людських потоків, що лежать в основі нормування t_p , встановлені на підставі численних натурних спостережень і багатоаспектних теоретичних досліджень [17-18]. Час же початку евакуації (інтервал часу від виникнення пожежі до початку евакуації людей), що визначається діючими нормативними актами, не має настільки глибоких обґрунтувань. Значення часу початку евакуації t_{ne} для будівель (споруд) без систем оповіщення обчислюється за результатами дослідження поведінки людей при пожежах в будівлях конкретного призначення.

Вплив на величину часу початку евакуації численних факторів (видів діяльності в будівлях різного призначення, вікового складу знаходяться в будівлі, їх фізичного та фізіологічного стану, інерційності систем виявлення та оповіщення) показує, що час початку евакуації повинно нормуватись як випадкова величина, оскільки тільки так стає можливим врахування впливу перерахованих факторів на розкид його значень.

Враховуючи наведене, концептуальна модель оцінювання пожежного ризику на АЗС з урахування впливу зовнішніх факторів матиме вигляд.

Етап 1. Визначення факторів, що впливають на розвиток пожежі.

Для формування сценаріїв розвитку пожежі важливо визначити фактори внутрішнього та зовнішнього середовища, що безпосередньо впливають на тенденцію (масштаб) розвитку пожежі. Серед таких факторів визначимо наступні: місце виникнення пожежі (табл. 2.1); закономірності розповсюдження пожежі (табл. 2.2); початкова область пожежі (табл. 2.3); параметри навколишнього середовища (табл. 2.4); початкові параметри приміщення (табл. 2.5). Для оцінки факторів доцільно використати експертні методи з причини відсутності відповідних статистичних даних.

Сутність методів експертних оцінок полягає в тому, що в основу прийнятого рішення, прогнозу, виведення закладається думка фахівця чи колективу фахівців, засноване на їх знаннях і практичному професійному досвіді. Вони допомагають встановити ступінь складності та актуальності

проблеми, визначити основні цілі і критерії, виявити важливі чинники і взаємозв'язки між ними.

Таблиця 2.1 – Можливі місця виникнення пожежі

№	Можливий варіант	Експертна оцінка, бал
1	Поруч з паливо роздавальними колонками	3
2	Поблизу з приміщенням для персоналу АЗС	1
3	На майданчику для автотранспорту	1
4	Поруч з резервуарами для зберігання палива	5

Таблиця 2.2 – Закономірності розповсюдження пожежі

№	Можливий варіант	Експертна оцінка, бал
1	Низька ймовірність горизонтального розповсюдження (охоплення пожежею нових областей)	1
2	Середня ймовірність розповсюдження (охоплення пожежею нових областей)	3
3	Висока ймовірність розповсюдження (охоплення пожежею нових областей)	5

Таблиця 2.3 – Початкова область пожежі

№	Можливий варіант	Експертна оцінка, бал
1	Від 0 до 0,5 м ²	1
2	Від 0,5 до 1 м ²	3
3	Більше 1 м ²	5

Таблиця 2.4 – Параметри навколишнього середовища

№	Можливий варіант	Експертна оцінка, бал
1	Агресивне середовище	5
2	Нейтральне середовище	3
3	Сприятливе середовище	1

Таблиця 2.5 – Початкові параметри приміщення

№	Можливий варіант	Експертна оцінка, бал
1	Велика кількість сходинок	1
2	Недостатня ширина дверного отвору	1
3	Недостатня кількість виходів	3
4	Наявність предметів, перешкоджаючих вільний вихід	3

Етап 2. Моделювання часу евакуації як випадкової величини.

Припустимо, що показники $t_{ne}, t_p, t_{ол}, t_{ск}$ є випадковими величинами, які формують можливі сценарії розвитку пожежі через розрахунок ймовірності евакуації людей під час пожежі. Для прогнозування найбільш песимістичних

сценаріїв евакуації логічно використовувати різні закони розподілу для будівель різних класів функціональної пожежної безпеки.

Етап 3. Формування сценаріїв розвитку пожежі.

Оскільки розрахунок оцінки пожежного ризику на промисловому об'єкті визначається як максимальне значення пожежного ризику з розглянутих сценаріїв пожежі, то на даному етапі відбувається визначення загальної кількості сценаріїв та моделювання умов розвитку пожежі, враховуючи фактори навколишнього середовища й місце пожежі, закономірності розповсюдження пожежі та параметри об'єкту.

Розрахунок оцінки індивідуального пожежного ризику Q_B для i -того сценарію пожежі визначається за формулою:

$$Q_B = Q_{\Pi i} \cdot (1 - K_{A\Pi i}) \cdot P_{\Pi Pi} \cdot (1 - P_{Ei}) \cdot (1 - K_{\Pi.3.i}) \cdot K_p, \quad (2.2)$$

де $Q_{\Pi i}$ – частота виникнення пожежі в будівлі протягом року. При відсутності статистичної інформації допускається приймати $Q_{\Pi} = 4 \cdot 10^{-2}$; K_p – коефіцієнт, що характеризує інтенсивність розвитку пожежі; $K_{A\Pi i}$ – коефіцієнт, що враховує відповідність установок автоматичного пожежогасіння (далі – АУП) вимогам нормативних документів з пожежної безпеки; P_{Ei} – ймовірність евакуації людей; $P_{\Pi Pi}$ – ймовірність присутності людей в будівлі; $K_{\Pi.3.i}$ – коефіцієнт, що враховує відповідність системи протипожежного захисту вимогам нормативних документів з пожежної безпеки.

В свою чергу, сценарій пожежі – це варіант розвитку пожежі з урахуванням прийнятого місця виникнення і характеру його розвитку, що визначається на основі даних про об'ємно-планувальні рішення, про розміщення горючого навантаження і людей на об'єкті. Розглядаються сценарії пожежі, при яких реалізуються найгірші умови для забезпечення безпеки людей.

Етап 4. Розрахунок рівня пожежного ризику з урахуванням зовнішніх факторів.

На основі реалізованих сценаріїв визначається рівень пожежного ризику:

$$Q_{\max} = \max\{Q_{B,1}, \dots, Q_{B,i}, \dots, Q_{B,N}\}, \quad (2.3)$$

де $Q_{B,i}$ – це розрахункова величина пожежного ризику для i -го сценарію пожежі; N – кількість розглянутих сценаріїв пожежі.

Після розрахунків отриману величину порівнюють з нормативним значенням пожежного ризику, що становить $Q_B^H = 10^{-6}$ рік⁻¹.

Етап 5. Розробка заходів зниження пожежного ризику. У випадку, якщо розраховане значення пожежного ризику перевищує нормативне число, розробляються додаткові протипожежні заходи щодо зниження пожежного ризику та розробляється коефіцієнт зниження ризику в залежності від обраного заходу в умовах обмеженого обсягу фінансових ресурсів організації.

До протипожежних заходів, спрямованих на зниження величини пожежного ризику, відносяться: побудова додаткових евакуаційних шляхів і виходів; встановлення систем автоматичного пожежогасіння; вдосконалення систем оповіщення людей про пожежу та управління евакуацією людей підвищеного типу; скорочення часу перебування людей на території; захист обладнання від пожежі і механічного пошкодження (підземне розташування, теплоізоляція, водяне зрошення і т.п.). Виходячи з того, що кожен захід потребує фінансових вкладень, для розподілу наявної у організації суми між наведеними діями пропонується використати метод динамічного програмування, в основі якого лежить розв'язок рівняння Беллмана.

Етап 6. Прогнозування наслідків пожежі. На даному етапі визначаються кількість легкозаймисті речовини (ЛМР), що випарувався з протоки, розміри зон вибухонебезпечних концентрацій та розміри зон ураження при пожежі-спалаху, доза теплового випромінювання, що отримує людина від «вогняного шару» й умовна імовірність ураження людини тепловим випромінюванням пожежі протоки.

Математична модель визначення рівня пожежного ризику наведена у Додатку А.

2.2 Об'єктно-орієнтована модель програмного забезпечення для оцінювання пожежного ризику

Проектування програмного забезпечення для оцінки пожежного ризику здійснено за допомогою стандарту UML у вигляді наступних діаграм:

- діаграми варіантів використання (use case diagrams) для моделювання бізнес-процесів організації і вимог до створюваної системи;
- діаграми класів (class diagrams) для моделювання статичної структури класів системи і зв'язків між ними;
- діаграми поведінки системи (behavior diagrams):
 - 1) діаграми послідовності (sequence diagrams);
 - 2) кооперативні діаграми (collaboration diagrams) для моделювання процесу обміну повідомленнями між об'єктами;
- діаграми станів (statechart diagrams) для моделювання поведінки об'єктів системи при переході з одного стану в інший;
- діаграми діяльності (activity diagrams) для моделювання поведінки системи в рамках різних варіантів використання, або моделювання діяльності;
- діаграми компонентів (component diagrams) для моделювання ієрархії компонентів (підсистем) системи.

Діаграма варіантів використання наведена на рисунку Б.1.

Актором виступає інженер з охорони праці АЗС, який проводить оцінку індивідуального пожежного ризику на АЗС та має доступ до бази даних.

Оцінка індивідуального пожежного ризику включає генерацію сценаріїв розвитку пожежі в залежності від умов внутрішнього та зовнішнього середовища та прогнозування наслідків пожежі.

Доступ до бази даних передбачає перегляд вхідних даних у вигляді довідників, а також збереження розрахованих значень оцінки пожежного ризику.

Передбачені наступні варіанти використання. Варіант «Робота з базою даних» включає перегляд даних, додатковою можливістю є редагування даних.

Варіант «Оцінка пожежного ризику» ініціюється варіантом «Визначення кількості сценаріїв» та має додаткову дію: «Прогнозування наслідків пожежі».

Варіант «Прогнозування наслідків пожежі» включає «Введення даних», «Розрахунок кількості РМП», «Розрахунок розмірів зон вибухонебезпечних концентрацій», «Розрахунок розмірів зон ураження при пожежі-спалаху» та «Розрахунок інтенсивності та дози теплового випромінювання».

Варіант «Визначення кількості сценаріїв» ініціюється варіантом «Вибір умов розвитку пожежі», який включає варіанти «Введення умов розвитку» та «Моделювання умов розвитку».

Наступною дією є «Моделювання параметрів часу», що передбачає вибір одного з наступних варіантів: «Введення параметрів часу», «Оцінка пожежного ризику (нормальний закон)», «Оцінка пожежного ризику (рівномірний закон)», «Візуалізація параметрів часу».

«Моделювання параметрів часу» ініціюється варіантом «Розрахунок оцінки пожежного ризику», що має додаткові дії «Відстеження динаміки зміни значення пожежного ризику», який включає «Вибір періоду» та «Візуалізація зміни значення пожежного ризику» та «Вибір заходів щодо зниження пожежного ризику», який складається з варіантів «Розрахунок коефіцієнту зниження ризику» та «Розподіл коштів між заходами».

Діаграма класів представляє статичну структуру моделі системи в термінології класів об'єктно-орієнтованого програмування. Діаграма класу представлена на рисунку Б.2.

Клас «Інженер з охорони праці» – головний активний об'єкт, який взаємодіє з програмою ззовні. Клас «АЗС» – керуючий клас, який забезпечує зв'язок користувача з іншими класами для виконання основних функцій програми. Клас «База даних» виконує операцію підключення до бази даних, перегляд даних і збереження розрахункових значень для подальшого їх використання. Клас «Заходи» виконує операцію перерахунку оцінки пожежного ризику з урахуванням заходів щодо зниження пожежного ризику. Клас «Часові характеристики» – за допомогою закону розподілення випадкової

величини моделює різні значення параметрів часу для сценаріїв та візуальне зображення часових параметрів. Клас «Оцінка пожежного ризику» виконує операції безпосереднього розрахунку пожежного ризику, порівняння його з нормативним, а також є можливість відстеження зміни оцінки пожежного ризику за різні періоди. Клас «Пожежа» – формує сценарії розвитку пожежі з урахуванням обраних умов розвитку, місця, закономірності розвитку пожежі та умов навколишнього середовища. Клас «Наслідки пожежі» виконує операцію прогнозування наслідків пожежі після введення початкових даних: кількість РМП, розмір зон вибухонебезпечних концентрацій, розмір зон ураження при пожежі-спалаху, інтенсивність та дозу теплового випромінювання.

Діаграма послідовності відображує послідовні дії користувача в програмі (рис. Б.3). Послідовність дій наступна: інженер з охорони праці, який є користувачем програми, підключається до бази даних, де має можливість редагування даних, які там зберігаються. Далі інженер вводить кількість сценаріїв, обирає умови розвитку, місце, закономірності розвитку пожежі та умови навколишнього середовища, моделює параметри часу за допомогою закону розподілу випадкової величини, вводить параметри закону розподілу; потім розраховуються параметри часу, здійснюється вивід результатів на екран та їх збереження. Далі здійснюється виклик форми з графіком та візуалізація параметрів часу. Після цього користувач обирає розрахунок оцінки пожежного ризику, де відбувається порівняння розрахункової величини з нормативним значенням та відображення результатів на формі. Також інженер може переглянути динаміку зміни значення оцінки пожежного ризику за різними періодами часу.

В разі необхідності має можливість викликати форму для перерахунку пожежного ризику, що передбачає вибір заходів щодо його зниження, розрахунку коефіцієнта зниження ризику, розподілу коштів між заходами. Після цього відбувається перерахунок рівня ризику та відображення на екрані кінцевих результатів. Також за потреби інженер може спрогнозувати значення можливих наслідків, а саме: розрахувати кількість РМП, обчислити розмір зон

вибухонебезпечних концентрацій та зон ураження при пожежі-спалаху, знайти інтенсивність та дозу теплового випромінювання.

За допомогою діаграми кооперації (рис. Б.4) відображаються відносини між об'єктами. Послідовність взаємодій і паралельних потоків визначається за допомогою порядкових номерів. Послідовність аналогічна як на діаграмі послідовності.

На рисунку Б.5 зображена діаграма станів, яка описує усі можливі стани об'єкта в системі. Після відкриття програми відбувається підключення до бази даних, інженер з охорони праці може спочатку або зробити частковий прогноз величин можливих наслідків пожежі, або виконати формування сценаріїв розвитку пожежі. Для цього він водить необхідну кількість сценаріїв, обирає умови для формування цих сценаріїв. Далі за допомогою рівномірного або нормального закону розподілу випадкової величини моделюються значення параметрів часу, після чого відбувається обчислення величини індивідуального пожежного ризику. Якщо є необхідність графічного відображення, то інженер обирає візуалізацію значень пожежного ризику, в іншому випадку переходить до порівняння розрахованої величини з нормативним значенням пожежного ризику.

У випадку, якщо розраховане значення оцінки перевищує нормативне число, здійснюється перехід на форму «Вибір заходів щодо зниження значення», де відбувається розрахунок коефіцієнту зниження рівня ризику, розподіл коштів між заходами та безпосередньо перерахунок індивідуального пожежного ризику. Якщо не перевищує, тоді відбувається збереження результатів та вихід з програми.

Інженер має вибір після розрахунку та перерахунку величини пожежного ризику перейти до прогнозування наслідків пожежі: розрахувати кількість рідкого моторного палива (РМП), обчислити розмір зон вибухонебезпечних концентрацій та зон ураження при пожежі-спалаху, знайти інтенсивність та дозу теплового випромінювання.


Елементами діаграми компонентів є фізичні модулі програмного коду (рис. Б.6). Система складається з 17 модулів, кожен з яких виконує певний набір функцій: «Main.pas» – перший модуль системи, де розташована основна інформація щодо АЗС; «Menu.pas» є головним модулем системи. За його допомогою здійснюється розрахунок індивідуального пожежного ризику та доступ до всіх основних модулів; «Scens.pas» – модуль, де відбувається введення кількості сценаріїв; «Condition.pas» – модуль вибору умов розвитку пожежі; «TimeH.pas» – модуль моделювання параметрів часу (випадково або користувачем), в залежності від вибору користувача відкриється одна з двох форм: «GenScen.pas» – модуль, де необхідно обрати закон розподілу випадкової величини для генерації параметрів часу; «ZRavn.pas» – модуль, де параметри часу моделюються за допомогою рівномірного закону розподілу; «ZNorm.pas» – модуль, де параметри часу моделюються за допомогою нормального закону розподілу; «TimeP.pas» – модуль, де користувач вводить параметри часу і де відображаються результати моделювання параметрів за допомогою одного з законів розподілу. Також є доступ до форми завантаження параметрів часу за обраний період; «SelDt.pas» – модуль завантаження параметрів часу з бази даних за обраний період часу; «IPRCal.pas» – модуль обчислення пожежного ризику. Також є можливість відстеження динаміки зміни значення пожежного ризику за різні періоди; «SelChart.pas» – модуль графічного відображення зміни значення пожежного ризику за різні періоди; «IPR_ReCal.pas» – модуль для повторного розрахунку пожежного ризику; «FireForecast» – модуль для прогнозування можливих наслідків пожежі; «ChartUnit.pas» – модуль графічного відображення отриманих значень пожежного ризику для різних сценаріїв; «Spr.pas» – модуль для відображення інформації з довідників; «BD.dmb» – база даних, де зберігаються вхідні дані та результати розрахунку пожежного ризику.

Отже, представлені діаграми відображають структуру і логіку майбутнього програмного забезпечення для оцінювання пожежного ризику.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІННЮВАННЯ ПОЖЕЖНОГО РИЗИКУ

Для розробки програмного забезпечення було обрано середовище програмування Delphi. Delphi – це RAD, або Rapid Application Development – це середовище швидкої розробки додатків. Програма складається з 17 форм, 7 з яких призначені для роботи з таблицями бази даних, одна головна.

Програмний продукт працює з базою даних Microsoft Access 2010, використовуючи технологію ADO. ADO (Active Data Objects) – це компонент технології доступу до даних від Microsoft (MDAC – Microsoft Data Access Components). Для здійснення підключення до бази даних Microsoft Access 2010 та роботи з нею були використані наступні компоненти: TADODConnection (вкладка ADO); TADOQuery (вкладка ADO); TADODTable (вкладка ADO); TDataSource (вкладка Data Access); TDBGrid (вкладка Data Controls); TDBNavigator (вкладка Data Controls).

Розглянемо роботу програми на прикладі конкретної автозаправної станії ТОВ «АЗТ Славутич». Головна форма програмного забезпечення містить в правому верхньому кутку логотип  компанії, в центрі – корисна інформація про АЗТ «Славутич», також на формі розташована кнопка «Оцінка пожежного ризику», яка призначена для розрахунку індивідуального пожежного ризику.

При натисканні на кнопку «Оцінка пожежного ризику» відкривається нове вікно, зовнішній вигляд якого представлений на рисунку 3.1. Форма містить інформацію щодо визначення індивідуального пожежного ризику та як здійснюється розрахунок оцінки. На вкладці меню відображаються 4 основні вкладки для розрахунку величини індивідуального пожежного ризику: введення даних; обчислення; наслідки пожежі; довідники.

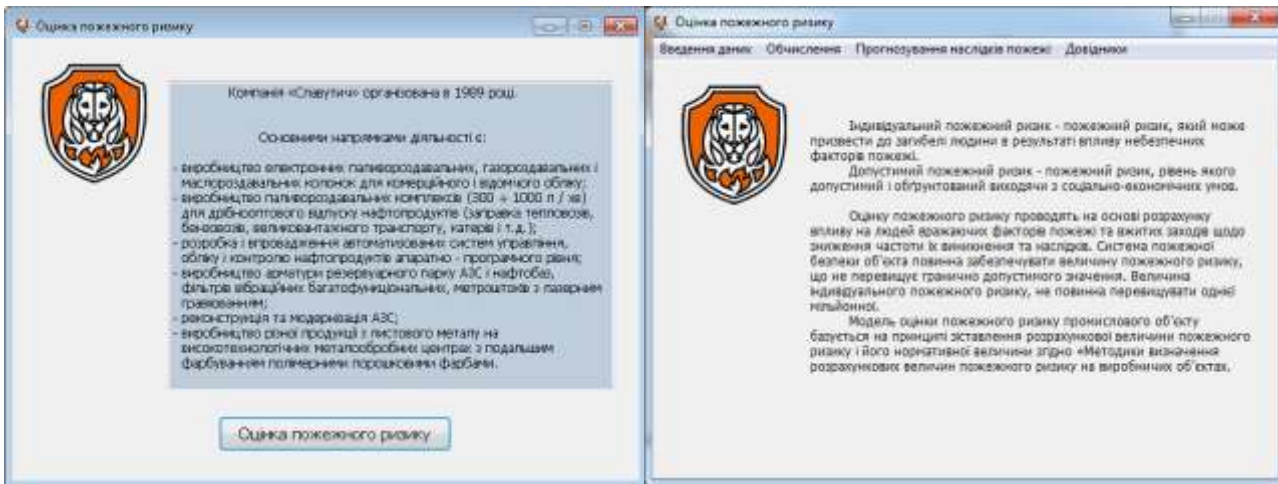


Рисунок 3.1 – Форма «Оцінка пожежного ризику»

При натисненні на вкладку «Введення даних» з'являється перелік можливих дій: вибір кількості сценаріїв; вибір умов розвитку пожежі; вибір часових характеристик. Активним полем є тільки «Вибір кількості сценаріїв», так як усі дії в програмі є послідовними, тільки в випадку, коли потрібно виправити введені дані, всі поля вкладки будуть активними. Тому вкладка «Обчислення» неактивна (рис. 3.2).

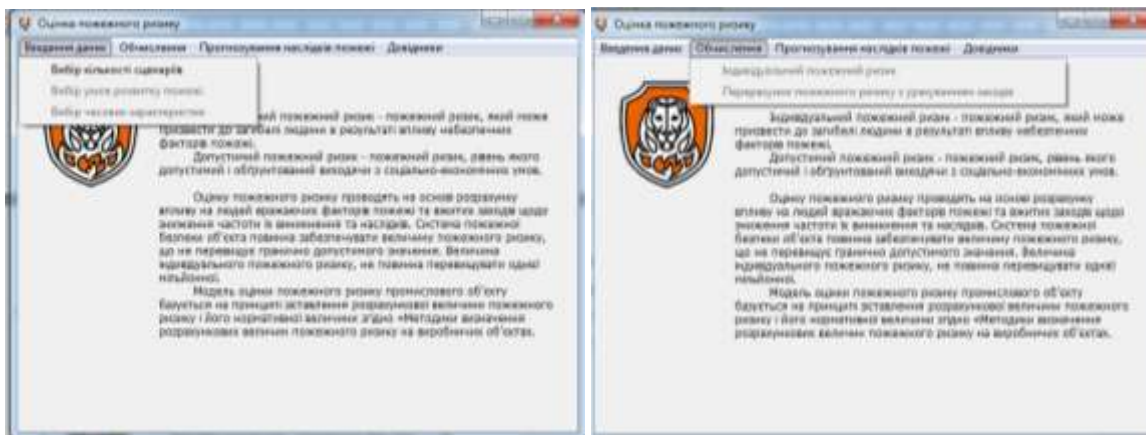


Рисунок 3.2 – Вкладка «Введення даних» та вкладка «Обчислення»

При переході на меню «Вибір кількості сценаріїв» з'являється інше вікно, де потрібно ввести необхідну кількість сценаріїв і натиснути «Далі». Натисканням на кнопку виконується збереження введеного значення з подальшим відображенням у таблиці на формі з кінцевим розрахунками.

Далі (рис. 3.3) відкривається вікно «Вибір умов розвитку пожежі» (рис. 4.6), де або користувач сам обирає умови для генерації сценаріїв, або генерація здійснюється в автоматичному режимі.

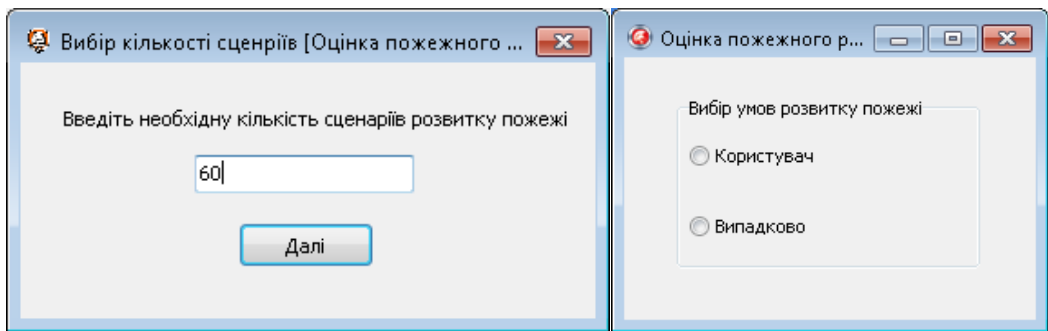


Рисунок 3.3 – Вибір кількості сценаріїв та умов розвитку пожежі

При виборі пункту «Користувач» на формі з'являється таблиця, де користувач для кожного сценарію обирає за кожним параметром підпункти із переліку. Крім цього на формі розміщені кнопки «Зберегти та продовжити» (виконує функцію збереження обраних параметрів) та «Назад» (повернення на попередню форму) (рис. 3.4). При виборі пункту «Випадково» програма генерує для кожного сценарію списку параметри самостійно. Аналогічно на формі розміщені кнопки «Зберегти та продовжити» та «Назад».

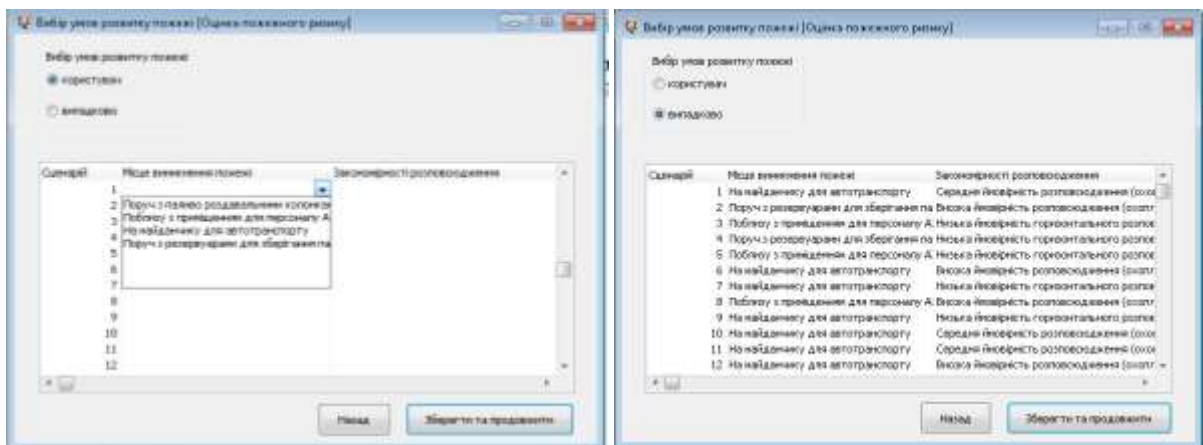


Рисунок 3.4 – Вибір умов розвитку пожежі

Незалежно від того, як саме були обрані «Умови розвитку пожежі», після натискання на кнопку «Зберегти та продовжити» відкривається форма «Вибір часових характеристик», де або користувач сам визначає часові характеристики для кожного сценарію, або це програма здійснюється автоматично.

У випадку, коли користувач вирішує сам внести значення параметрів часових характеристик, відкривається форма, як на рисунку 3.5. Справа розміщено пояснення умовних позначень, в таблицю користувач заносить дані, враховуючи обмеження – значення від 0,01 до 60. На формі розміщені такі кнопки: «Завантажити з бази даних» – завантаження з бази даних раніше

введених значень користувачем. Є можливість вибору завантаження даних за період. «Перейти до розрахунків» – перехід на форму, де починається робота з вкладкою «Обчислити».

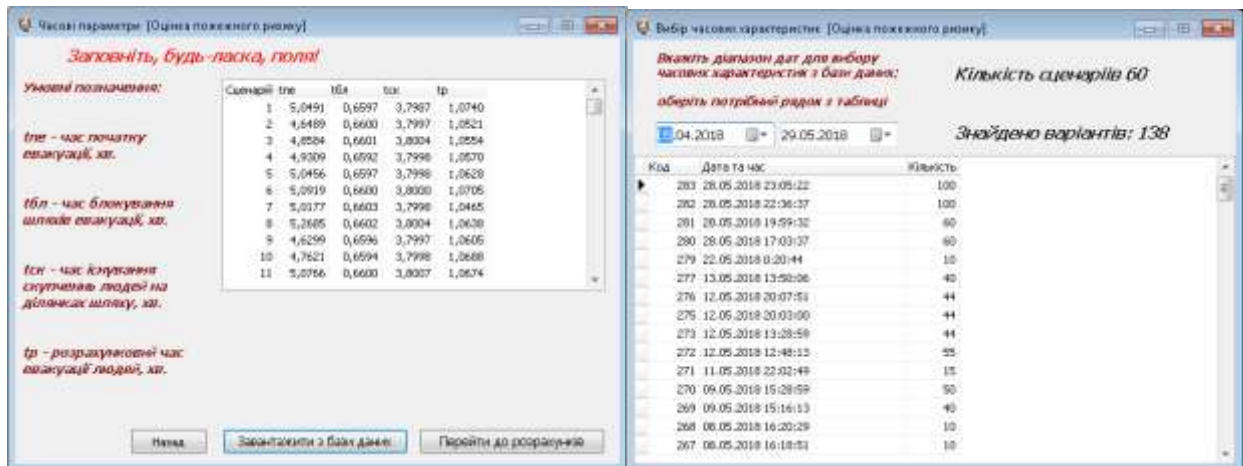


Рисунок 3.5 – Введення параметри часу

При виборі пункту «Випадково» значення часових величин генеруються випадковим чином за допомогою нормального чи рівномірного закону розподілу (рис. 3.6).

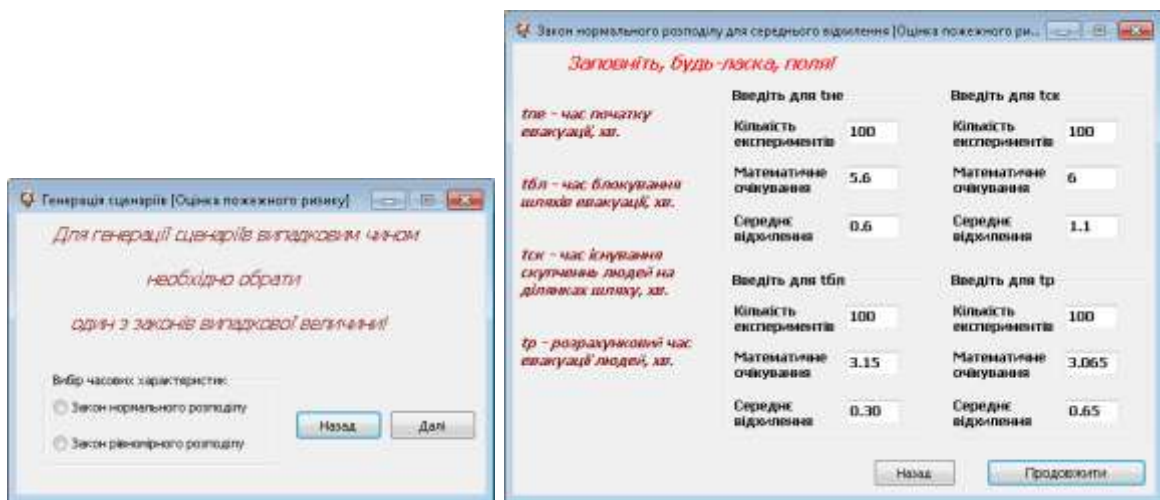


Рисунок 3.6 – Вікно генерації випадковим чином часових характеристик

При виборі пункту «Закон нормального розподілу» відкриється відповідна форма, яка містить пояснення для кожної невідомої величини та кнопки «Продовжити», яка буде активна в разі заповнення всіх полів, та «Назад» – повернення на попередню форму.

У програмі є обмеження на формі «Закон нормального розподілу» на введення значення параметрів «Математичне очікування» й «Середнє

відхилення» в проміжку від 0,001 до 50, та «Кількість експериментів» – до 1000 експериментів. Також при наведенні даних в поле «Середнє відхилення» з'являється підказка, що «Величина середнього відхилення завжди є меншою величиною, ніж математичне очікування». В разі введення неправильних значень параметрів поле виділяється червоним кольором та кнопка «Продовжити» буде стає неактивною. При виборі пункту «Закон рівномірного розподілу» все здійснюється аналогічно.

Після виконання вище описаних дій користувач може перейти до обчислення параметру «Індивідуальний пожежний ризик». Форма має вигляд як на рисунку 3.7. Візуально вікно поділено на дві частини: праворуч – таблиця з розрахованими значеннями оцінки пожежного ризику Q_v ; зліва – графічне відображення розрахованих значень у вигляді гістограми. Столпчик з максимальним значенням Q_v виділений іншим кольором. На формі розташована кнопка «Відстежити динаміку», яка дозволяє порівняти результати, отримані за різні періоди часу. При цьому порівняння наводиться графічно.

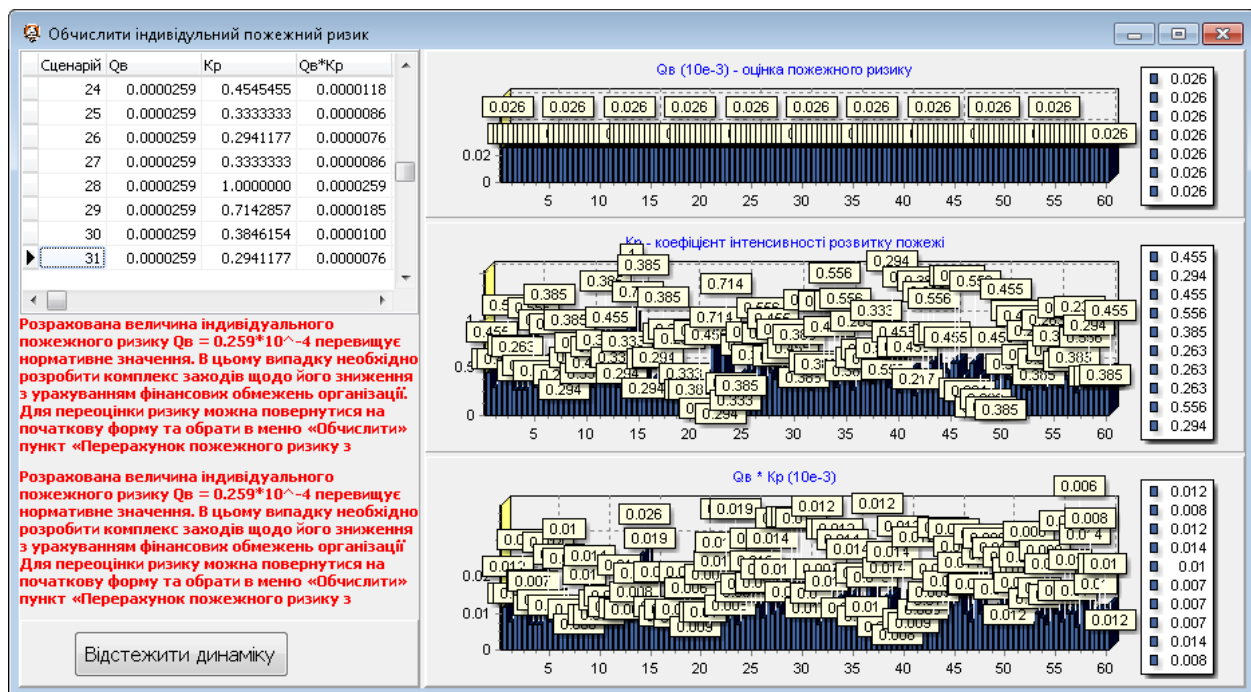


Рисунок 3.7 – Вкладка «Обчислити Індивідуальний пожежний ризик»

Використання пункту «Перерахунок пожежного ризику з урахуванням заходів» необхідне, якщо розрахована величина індивідуального пожежного

ризик перевищує допустиме значення і необхідні додаткові протипожежні заходи щодо зниження пожежного ризику. На формі розміщена кнопка «Завантажити дані», в базі даних є заздалегідь розраховані коефіцієнти ефективності кожного заходу в залежності від величини фінансових вкладень. Після того як дані завантажені з бази даних, на формі стане активною кнопка «Перерахувати пожежний ризик» (рис. 3.8). Користувач може редагувати дані на формі або відкрити базу даних та змінити значення.

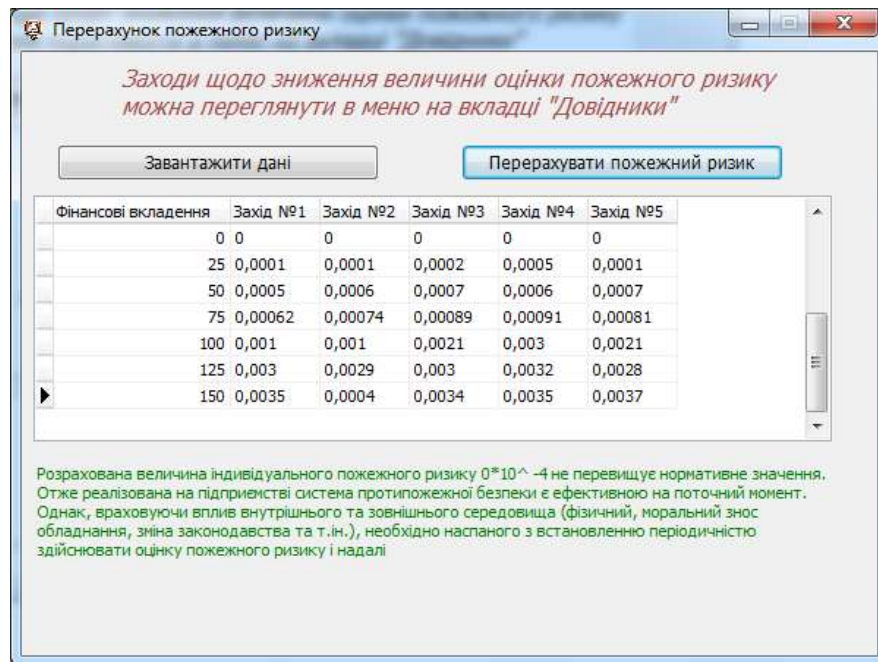


Рисунок 3.8 – Перерахунок оцінки пожежного ризику

Вкладка «Прогнозування наслідків пожежі» при натисканні містить наступні можливі варіанти дій: кількість рідкого моторного палива (РМП), що випарувався з протоки; розміри зон вибухонебезпечних концентрацій; розміри зон ураження при пожежі-спалаху; доза теплового випромінювання, яку отримає людина.

Розрахунок наслідків пожежі складається з трьох величин:

- 1) величина «...: середньостатистична величина», обчислення якої можливе до початку будь-яких розрахунків на попередніх вкладках;
- 2) величина «...: до реалізації заходів зниження пожежного ризику», розрахунок якої можливий після виконаних розрахунків на вкладці «Обчислення – Індивідуальний пожежний ризик»;

3) величина «... після реалізації заходів зниження пожежного ризику», обчислення якої доступно після відповідних розрахунків на вкладці «Обчислення – Перерахунок пожежного ризику з урахуванням заходів». В випадках, коли відповідні обчислення не були виконані, то поле вибору не активне, та спливає підказка про неможливість розрахунку.

Приклади розрахунків наведені на рисунках 3.9–3.10.

Рисунок 3.9 – Вікно «Кількість РМП»

Рисунок 3.10 – Вікно «Доза теплового випромінювання»

В меню вкладки «Довідники» представлені довідники: «Місце виникнення пожежі», «Закономірності розповсюдження пожежі», «Початкова

область пожежі», «Параметри навколишнього середовища», «Початкові параметри приміщення», «Список заходів щодо зниження пожежного ризику».

Реалізований програмний продукт при оцінюванні пожежного ризику дозволяє формувати песимістичні сценарії розвитку пожежі, моделювати параметри часу за допомогою одного з двох законів розподілу випадкової величини, дозволяє графічно інтерпретувати вхідні параметри, моделювати умови розвитку пожежі на основі експертних оцінок факторів внутрішнього та зовнішнього середовища, обирати заходи для зниження ризику в умовах обмеженого фінансового ліміту, робити прогноз можливих наслідків пожежі.

ВИСНОВКИ

Для оцінки пожежного ризику запропонована математична модель, яка базується на зіставленні розрахункової величини пожежного ризику і його нормативної величини. Для визначення очікуваного рівня ризику запропонована генерація сценаріїв розвитку пожежі в залежності від умов зовнішнього та внутрішнього середовища. Визначення розрахункової величини пожежного ризику здійснювалося з урахуванням величини часу евакуації, визначеної методом імітаційного моделювання.

Виконано проектування програмного забезпечення за допомогою UML-діаграм, які відображають його структуру і логіку. Побудовані діаграма варіантів використання, діаграма класів, діаграма послідовності та кооперації, діаграма станів та діаграма компонентів.

Розроблене програмне забезпечення надає можливість вибору умов та часових характеристик для формування різних сценаріїв розвитку пожеж на автозаправній станції. Також у випадку отримання значення індивідуального пожежного ризику, що перевищує нормативне, передбаченні заходи для його зниження. Реалізована можливість прогнозування можливих наслідків пожежі, а саме визначення кількості рідинно-моторного палива, що випарувалась при аварії, розмірів зон вибухонебезпечних концентрацій та зон ураження при пожежі-спалаху, ймовірної дози теплового випромінювання.

Результати роботи були апробовані на всеукраїнській науково-практичній Internet-конференції, м. Черкаси, 2019 р., міжнародній науково-технічній конференції молодих учених та студентів, м. Тернопіль, 28–29 листопада 2018 р., XXII міжнародній науково-технічній конференції, м. Северодонецьк, 26–27 квіт. 2019 р. та опубліковані в науковому журналі – «Вісник економічної науки України», 2018 р.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Математичне визначення ризику [Електронний ресурс]. – Режим доступу: http://pidruchniki.com/72391/ekologiya/matematichne_viznachennya_riziku
2. Постанова Кабінету Міністрів України від 5 вересня 2018 р. № 715 Про затвердження критеріїв, за якими оцінюється ступінь ризику від провадження господарської діяльності та визначається періодичність здійснення планових заходів державного нагляду (контролю) у сфері техногенної та пожежної безпеки Державною службою з надзвичайних ситуацій – Режим доступу: http://search.ligazakon.ua/l_doc2.nsf/link1/KP180715.html
3. Причини виникнення ризику [Електронний ресурс]. – Режим доступу: <http://studepedia.org/index.php?vol=1&post=62935>
4. НАПБ А.01.001-2014 Правила пожежної безпеки в Україні [Електронний ресурс]. – Режим доступу: <http://zakon5.rada.gov.ua/laws/show/z0252-15>
5. ДСТУ 2272:2006 ССБТ. Пожежна безпека. Терміни та визначення . – Київ: Держстандарт України, 2006. – 38 с.
6. ДСТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою [Електронний ресурс]. – Режим доступу: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65419
7. НПБ П1-98*. Автозаправні станції. Вимоги пожежної безпеки [Електронний ресурс]. – Режим доступу: <http://zakon2.rada.gov.ua/laws/show/z0252-15>
8. Розрахунок пожежного ризику [Електронний ресурс]. – Режим доступу: http://expert112.com.ua/raschet_pojarnogo_riska/index_ua.html

9. Руководство по анализу оценки пожарного риска. – Национальная организация по противопожарной защите (NFPA), г. Куинси, штат Массачусетс, США, 2010. – 33 с.

10. Закон України «Про основні засади державного нагляду (контролю) за господарською діяльністю» від 2007 року [Електронний ресурс]. – Режим доступу: <http://zakon3.rada.gov.ua/laws/show/877-16>

11. Офіційний сайт програмного комплексу «Fire Dynamics» [Електронний ресурс]. – Режим доступу: <http://www.sitis.ru>

12. Pathfinder – расчет времени эвакуации людей [Електронний ресурс]. – Режим доступу: <https://pyrosim.ru/programmy-dlya-rascheta-pozharnogo-riska/raschet-vremeni-ehvakuacii-lyudej>

13. Офіційний сайт програмного комплексу «СІТІС: Спринт» [Електронний ресурс]. – Режим доступу: <http://www.sitis.ru/>

14. Офіційний сайт програмного комплексу «Fenix +» [Електронний ресурс]. – Режим доступу: <http://mst.su/fenix/>

15. ДБН В.1.2-7-2008 «Система забезпечення надійності та безпеки будівельних об'єктів. Основні вимоги до будівель і споруд. Пожежна безпека» [Електронний ресурс]. – Режим доступу: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=45884

16. ГОСТ 12.1.004-91* «Система стандартов безопасности труда. Пожарная безопасность. Общие требования» [Електронний ресурс]. – Режим доступу: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=48679

17. Холщевников В.В., Самошин Д.А. Эвакуация и поведение людей при пожарах: Учеб. пособие. М.: Академия ГПС МЧС России, 2009. – 212 с.

18. Цвіркун С. В. Оцінка індивідуального пожежного ризику громадських будівель різними методами / С. В. Цвіркун, В.С. Щербина, П.І. Заїка // Пожежна безпека: теорія і практика : збірник наукових праць. – Черкаси : АПБ ім. Героїв Чорнобиля, 2014. – №16. – 121–129 с

19. Методика определения расчетных величин пожарного риска на производственных объектах: утв. приказом МЧС России от 10.07.2009 г..

№ 404. Официальный сайт МЧС России [Электронный ресурс]. – Режим доступа: http://www.mchs.gov.ru/law/index.phpID=8431&sphrose_id=1369844

20. Самошин Д.А. Законы распределения случайной величины времени начала эвакуации людей при пожарах / Д.А. Самошин // Академия ДПС МЧС России. Интернет-журнал. – 2016. [Электронный ресурс]. – Режим доступа: <http://www.fireevacuation.ru/files/papers/distribution%20law.pdf>

21. Пожаровзрывоопасность веществ и материалов и средства их тушения / Под ред. А.Н. Баратова, А.Я. Корольченко. – М.: Химия, 1990. – 460 с.

22. ДСТУ-Н Б В.1.1-27:2010 Будівельна кліматологія. Мінрегіонбуд України, 2011 – 130 с.

ДОДАТОК А. Математична модель визначення рівня пожежного ризику

Модель оцінювання пожежного ризику промислового об'єкту (автозаправної станції) базується на принципі зіставлення розрахункової величини пожежного ризику і його нормативної величини згідно «Методики визначення розрахункових величин пожежного ризику на виробничих об'єктах» [19].

Відповідно до [19] індивідуальний пожежний ризик відповідає необхідному рівню, якщо:

$$Q_B \leq Q_B^H, \quad (\text{A.1})$$

де Q_B^H – нормативне значення індивідуального пожежного ризику, $Q_B^H = 10^{-6}$ рік⁻¹;

Q_H – розрахункова величина індивідуального пожежного ризику.

Розрахункова величина пожежного ризику в будівлі, споруді або будові визначається як максимальне значення пожежного ризику з розглянутих сценаріїв пожежі, представлена формулою (2.3) у концептуальній моделі на етапі 3.

Сценарій пожежі є варіантом розвитку пожежі з урахуванням прийнятого місця виникнення і характеру його розвитку. Сценарій пожежі визначається на основі даних про об'ємно-планувальні рішення, про розміщення горючого навантаження і людей на об'єкті. При розрахунку розглядаються сценарії пожежі, при яких реалізуються найгірші умови для забезпечення безпеки людей.

В якості сценаріїв з найгіршими умовами пожежі слід розглядати сценарії, що характеризуються найбільш затрудженими умовами евакуації людей і (або) найбільш високою динамікою наростання небезпечних факторів пожежі (далі – НФП), а саме пожежі:

– в приміщеннях, розрахованих на одноразову присутність 50 і більше осіб;

- в системах приміщень, в яких через поширення НФП можливе швидке блокування шляхів евакуації (коридорів, евакуаційних виходів тощо). При цьому вогнище пожежі вибирається в приміщенні малого обсягу поблизу від одного з евакуаційних виходів, або в приміщенні з великою кількістю горючого навантаження, що характеризується високою швидкістю поширення полум'я;
- в приміщеннях і системах приміщень атріумного типу;
- в системах приміщень, в яких через недостатню пропускну спроможність шляхів евакуації можливе виникнення тривалих скупчень людських потоків.

У випадках, коли перераховані типи сценаріїв не відображають всіх особливостей об'єкта, можливий розгляд інших сценаріїв пожежі.

У приміщенні, яке має два і більше евакуаційні виходи, вогнище пожежі слід розміщувати поблизу виходу, що має найбільшу пропускну здатність. При цьому даний вихід вважається блокованим з перших секунд пожежі та при визначенні розрахункового часу евакуації не враховується. У приміщенні з одним евакуаційним виходом час блокування виходу визначається розрахунковим методом.

Сценарії пожежі, які не можливі при нормальному режимі експлуатації об'єкта (теракти, підпали, зберігання горючої навантаження, не передбаченої призначенням об'єкта і т.д.), не розглядаються.

Розрахункова величина індивідуального пожежного ризику для i -го сценарію пожежі Q_B розраховується за формулою (2.2), яка наведена в концептуальній моделі на етапі 4.

Значення коефіцієнту, що враховує відповідність АУП вимогам нормативних документів з пожежної безпеки параметра, приймається рівним $K_{АП} = 0,9$, якщо виконується хоча б одна з таких умов:

- будівля обладнана системою АУП, що відповідає вимогам нормативних документів з пожежної безпеки;
- обладнання будівлі системою АУП не потрібно відповідно до вимог нормативних документів з пожежної безпеки.

В інших випадках приймається рівною нулю.

Ймовірність присутності людей в будівлі, $P_{ПРi}$, визначається з співвідношення:

$$P_{ПРi} = \frac{t_{функ}}{24}, \quad (A.2)$$

де $t_{функ}$ – час знаходження людей в будівлі в годинах;

Коефіцієнт, що враховує відповідність системи протипожежного захисту вимогам нормативних документів з пожежної безпеки, $K_{П.З.i}$, розраховується за формулою:

$$K_{П.З.i} = 1 - (1 - K_{обнi} \cdot K_{COVEi}) \cdot (1 - K_{обнi} \cdot K_{ПДЗi}), \quad (A.3)$$

де $K_{обнi}$ – коефіцієнт, що враховує відповідність системи пожежної сигналізації вимогам нормативних документів з пожежної безпеки. Значення параметра приймається рівним $K_{обн} = 0,8$, якщо виконується хоча б одна з таких умов:

- будівля обладнана системою пожежної сигналізації, яка відповідає вимогам нормативних документів з пожежної безпеки;
- обладнання будівлі системою пожежної сигналізації не потрібно відповідно до вимог нормативних документів з пожежної безпеки.

В інших випадках $K_{обн}$ приймається рівним нулю.

K_{COVEi} – коефіцієнт, що враховує відповідність системи оповіщення людей про пожежу та управління евакуацією людей вимогам нормативних документів з пожежної безпеки.

Значення параметра приймається рівним $K_{COVE} = 0,8$, якщо виконується хоча б одна з таких умов:

- будівля обладнана системою оповіщення людей про пожежу та управління евакуацією людей, що відповідає вимогам нормативних документів з пожежної безпеки;

– обладнання будівлі системою оповіщення людей про пожежу та управління евакуацією людей не потрібне відповідно до вимог нормативних документів з пожежної безпеки.

В інших випадках K_{COVE} приймається рівним нулю.

$K_{ПДЗ i}$ – коефіцієнт, що враховує відповідність системи протидимового захисту вимогам нормативних документів з пожежної безпеки.

Значення параметра приймається рівним $K_{ПДЗ} = 0,8$, якщо виконується хоча б одна з таких умов:

- будівля обладнана системою протидимного захисту, що відповідає вимогам нормативних документів з пожежної безпеки;
- обладнання будівлі системою протидимного захисту не потрібне відповідно до вимог нормативних документів з пожежної безпеки.

В інших випадках $K_{ПДЗ}$ приймається рівним нулю.

Імовірність евакуації P_{Ei} розраховується за формулою:

$$P_{Ei} = \begin{cases} \frac{0,8 \cdot t_{\text{бл}}}{t_p}, & \text{якщо } t_p < 0,8 \cdot t_{\text{бл}} < t_p + t_{ne} \text{ та } t_{ck} \leq 6 \text{ хв} \\ 0,999, & \text{якщо } t_p + t_{ne} \leq 0,8 \cdot t_{\text{бл}} \text{ та } t_{ck} \leq 6 \text{ хв} \\ 0,000, & \text{якщо } t_p \geq 0,8 \cdot t_{\text{бл}} \text{ або } t_{ck} > 6 \text{ хв} \end{cases}, \quad (\text{A.4})$$

де t_{ne} – час початку евакуації, хв; t_p – розрахунковий час евакуації людей, хв.; $t_{\text{бл}}$ – час блокування шляхів евакуації, хв.; t_{ck} – час існування скупчень людей на ділянках шляху (щільність людського потоку на шляхах евакуації перевищує значення 0,5).

Припустимо, що показники $t_{ne}, t_p, t_{\text{бл}}, t_{ck}$ є випадковими величинами, які формують можливі сценарії розвитку пожежі через розрахунок ймовірності швидкості евакуації людей під час пожежі.

Для визначення наведених випадкових величин необхідно обрати один із законів розподілу випадкової величини. Для прогнозування найбільш песимістичних сценаріїв евакуації логічно використовувати різні закони розподілу для будівель різних класів функціональної пожежної небезпеки.

Однак, зважаючи на дослідження, проведені А. Самошиним [20] найбільш коректними законами є рівномірний (А.5) та нормальний (А.6) закони розподілу випадкової величини:

$$t_{nei} = R(b - a) + a, \quad (A.5)$$

де R – випадкова величина, розподілена рівномірно на відріжку $[0; 1]$.

$$t_{nei} = \mu + \delta \cdot \sqrt{-2 \cdot \ln R_1} \cdot \sin(2 \cdot \pi \cdot R_2), \quad (A.6)$$

де R_1 та R_2 – випадкові величини, розподілені на відріжку $[0; 1]$.

При цьому доцільно розраховувати значення $t_{пе}^{max}$ та $t_{пе}^{min}$ для формування діапазону зміни випадкової величини.

Враховуючи, що на всі показники $t_{пе}$, $t_{бл}$, t_p , $t_{ск}$ впливають однакові фактори, то можна припустити, що вони мають однаковий закон розподілу – рівномірний або нормальний.

Таким чином, для кожного сценарію за рівномірним або нормальним законом розподілу випадкової величини генеруються свої (випадкові) значення показників $t_{пе}$, $t_{бл}$, t_p , $t_{ск}$ (їх нижня та верхня межі).

Далі, для формування сценаріїв розвитку пожежі важливо визначити фактори внутрішнього та зовнішнього середовища, що безпосередньо впливають на тенденцію (масштаб) розвитку пожежі (описані в концептуальній моделі у попередньому підрозділі на етапі 1).

Коефіцієнт, що характеризує інтенсивність розвитку пожежі, K_p розраховується за формулою:

$$K_p = \frac{1}{\frac{\sum_{i=1}^n R_i}{n}}, \quad (A.7)$$

де R_i – експертна оцінка в балах за i -м фактором; n – кількість факторів (дорівнює 5).

В разі, якщо розрахована величина індивідуального пожежного ризику перевищує допустиме значення, то слід розробити додаткові протипожежні заходи щодо зниження пожежного ризику (табл. А.1).

Протипожежні заходи, що спрямовані на зниження величини пожежного ризику, були описані також у концептуальній моделі на етапі 5.

Виходячи з того, що кожен захід потребує фінансових вкладень, для розподілу наявної у організації суми між наведеними альтернативами пропонується використати метод динамічного програмування, в основі якого лежить розв'язок рівняння Беллмана.

Рівняння Беллмана має вигляд:

$$F_{n-k}(x_k) = \max[W_{k+1}(x_k, U_{k+1}) + F_{n-k-1}(x_{k+1})], \quad (k = \overline{0, n}) \quad (\text{A.8})$$

де стан системи S на k -му кроці ($k = \overline{1, n}$) визначається сукупністю чисел $x_k^i = x_k^1, x_k^2, \dots, x_k^m$, отриманих в результаті реалізації управління U_k , яке забезпечує перехід системи зі стану x_{k-1} у стан x_k , в який перейшла система S , залежить від даного стану x_{k-1} та вибраного управління U_k і не залежить від того, яким чином система S прийшла до стану x_{k-1} . Ця умова називається відсутністю наслідків. Далі припускається, якщо в результаті реалізації k -го кроку забезпечений певний виграш, який також залежить від первинного стану системи x_{k-1} та обраного управління U_k і дорівнює $W_k(x_{k-1}, U_k)$, то загальний виграш за n кроків складе $F = \sum_{k=1}^n W_k(x_{k-1}, U_k)$, тобто загальний виграш дорівнює сумі виграшів на кожному етапі.

В нашому випадку під виграшем розуміється найбільше значення коефіцієнту зниження пожежного ризику.

Враховуючи кількість заходів, цільові функції матимуть вигляд:

$$F_5(x) = \max_{0 \leq x_5 \leq x} \{(f_5(x_5))\}; \quad F_4(x) = \max_{0 \leq x_4 \leq x} \{(f_4(x_4) + F_5(x - x_5))\};$$

$$F_3(x) = \max_{0 \leq x_3 \leq x} \{(f_3(x_3) + F_4(x - x_3))\}; \quad F_2(x) = \max_{0 \leq x_2 \leq x} \{(f_2(x_2) + F_3(x - x_2))\};$$

$$F_1(x) = \max_{0 \leq x_1 \leq x} \{(f_1(x_1) + F_2(x - x_1))\}.$$

Ефективність заходів, визначена експертним шляхом, в залежності від величини фінансових вкладень наведена в таблиці А.1.

Таблиця А.1 – Заходи щодо зниження пожежного ризику та їх ефективність в залежності від величини фінансових вкладень

Фінансові вкладення	Ефективність заходу – коефіцієнт зниження пожежного ризику				
	Захід №1	Захід №2	Захід №3	Захід №4	Захід №5
0	0,000	0,000	0,000	0,000	0,000
25	0,010	0,018	0,012	0,015	0,010
50	0,015	0,020	0,017	0,016	0,017
75	0,020	0,024	0,018	0,021	0,018
100	0,025	0,026	0,021	0,030	0,021
125	0,030	0,029	0,030	0,032	0,028
150	0,035	0,040	0,034	0,035	0,041

З урахуванням ефективності нових заходів щодо зниження пожежного ризику здійснюється повторний розрахунок величини індивідуального пожежного ризику: знаходиться добуток отриманого значення коефіцієнту зниження пожежного ризику Z та оцінки пожежного ризику Q , отриманої до впровадження додаткових заходів.

Необхідно враховувати, що пожежі на автозаправних станціях супроводжуються певними наслідками, наймасштабніших з яких є:

- кількість рідкого моторного палива (РМП), що випарувався з протоки;
- розміри зон вибухонебезпечних концентрацій;
- розміри зон ураження при пожежі-спалаху;
- доза теплового випромінювання, яку отримує людина.

Маса рідини, що випарувалась, за час τ , знаходиться за формулою:

$$m_e = W \cdot F \cdot \tau, \quad (\text{A.9})$$

де F – площа протоки при розтіканні на необмеженій поверхні; W – інтенсивність випаровування; τ – тривалість випаровування бензину.

Інтенсивність випаровування розраховується як:

$$W = 10^{-6} \cdot \sqrt{M} \cdot p_s, \quad (\text{A.10})$$

де M – молярна маса бензину, згідно з [21] $M = 95 \text{ кг/кмоль}$; p_s – тиск насичених парів бензину.

Тиск насичених парів бензину розраховується за формулою:

$$\lg p_s = A - B / (C_A + t_p), \quad (\text{A.11})$$

де A , B , C_A – константи рівняння Антуана, що згідно [21] дорівнюють: $A = 4.26511$, $B = 695.019$, $C_A = 223.220$; t_p – розрахункова температура рідини, згідно [22] $t_p = 37^\circ\text{C}$.

Площа протоки при розтіканні на необмеженій поверхні знаходиться за формулою:

$$F = f_p \cdot V, \quad (\text{A.12})$$

де f_p – коефіцієнт розлиття при розтіканні на бетонному або асфальтовому покритті, згідно [19] $f_p = 150 \text{ м}^{-1}$; V – обсяг легкозаймистої речовини (ЛЗР), що вийшов на територію АЗС. Обсяг визначається за формулою:

$$V = q_n \cdot \tau_{откл} + l_{шл} \cdot \frac{\pi \cdot d_{шл}^2}{4}, \quad (\text{A.13})$$

де $d_{шл}$ – внутрішній діаметр шланга; $l_{шл}$ – довжина шланга; q_n – продуктивність насоса, що становить $q_n = 100 \text{ л/мин} = 1.667 \cdot 10^{-3} \text{ м}^3 / \text{с}$ [22];

$\tau_{откл}$ – ймовірність відмови систем автоматичного відключення трубопроводів (ТРК).

Ймовірність відмови систем автоматичного відключення трубопроводів може приймати наступні значення:

1) при середньостатистичній тривалості спрацьовування пристроїв, що відтинають, у системі по прямому струму $\tau_{откл} = 1 \text{ с}$;

2) з урахуванням ймовірності ризику пожежі до реалізації заходів
 $\tau_{откл} = 1 \cdot (1 + Q_{\theta})$.

3) з урахуванням ймовірності ризику пожежі після реалізації заходів
 $\tau_{откл} = 1 \cdot (1 + Q_{\theta})$.

Тривалість випаровування бензину знаходять за формулою:

$$\tau = \min\left(\frac{m_{жс}}{F \cdot W}; 900 \text{ с}\right), \quad (\text{A.14})$$

$m_{жс}$ – маса РМП, що вийшла на територію АЗС, яка розраховується як:

$$m_{жс} = \rho_{жс} \cdot V, \quad (\text{A.15})$$

де $\rho_{жс}$ – щільність бензину, згідно [33] $\rho_{жс} = 729 \text{ кг/м}^3$.

У випадку, коли маса РМП, що вийшла на територію АЗС, менше 20 кг, згідно [21] тривалість випаровування бензину приймається рівною $\tau = 900 \text{ с}$. В іншому випадку – знаходять за формулою:

$$\tau = \left(\frac{m_{жс}}{F \cdot W}\right). \quad (\text{A.16})$$

Вихід РМП з гумово-металевого шлангу на територію АЗС при відсутності миттєвого займання та швидкості вітру менше 1 м/с супроводжується утворенням горючих пароповітряних сумішей – зон вибухонебезпечних концентрацій. Розрахункове визначення розмірів зон, обмежених нижньою концентраційною межею поширення полум'я (НКМП), визначають відповідно до методу, наведеному в [21]. При випаровуванні бензину з поверхні розливу радіус $R_{НКМП}$, що обмежує область концентрацій, що перевищують НКМП, розраховується за формулою:

$$R_{НКМП} = 3.2 \cdot \left(\frac{\tau}{3600}\right) \cdot \left(\frac{P_S}{\varphi_H}\right)^{0.8} \cdot \left(\frac{m_e}{\rho_{II} \cdot P_S}\right)^{0.33}, \quad (\text{A.17})$$

де φ_H – нижня концентраційна межа поширення полум'я, згідно [33]
 $\varphi_H = 1.1\% \text{ (об.)}$.

ρ_{II} – щільність пари бензину при робочій температурі і атмосферному тиску. Щільність пари бензину розраховується за формулою:

$$\rho_{\Pi} = \frac{M}{V_0 \cdot (1 + 0.00367 \cdot t_p)}, \quad (\text{A.18})$$

де M – молярна маса бензину, згідно з [33] $M = 95 \text{ кг/кмоль}$; V_0 – молярний об'єм пари, згідно з [22] $V_0 = 22/41 \text{ м}^3 / \text{кмоль}$.

У разі утворення пароповітряної суміші в не захищеному технологічним устаткуванням приміщенні та його запалюванні відносно слабким джерелом (наприклад, іскрою), згоряння цієї суміші відбувається, як правило, з невеликими видимими швидкостями полум'я. При цьому амплітуди хвилі стиснення малі і можуть не прийматися до уваги при оцінці вражаючої дії. В цьому випадку реалізується так званий пожежа-спалах, при якому зона ураження високотемпературними продуктами згоряння пароповітряної суміші практично збігається з максимальним розміром хмари продуктів згорання (тобто уражаються в основному об'єкти, що потрапляють в цю хмару).

Радіус зони ураження людей високотемпературними продуктами згоряння пароповітряної суміші визначають за формулою [21]:

$$R_F = 1.2 \cdot R_{HKMP}, \quad (\text{A.19})$$

Доза теплового випромінювання, Q , яку отримає людина, якщо буде перебувати під його впливом деякий час, знаходиться за формулою:

$$Q = 1000 \cdot q \cdot \tau_{\text{бл}}, \quad (\text{A.20})$$

де $\tau_{\text{бл}}$ – час блокування шляхів евакуації; q – інтенсивність теплового випромінювання на поверхні даного об'єкту від «вогняного шару», кВт / м².

Інтенсивність розраховується як:

$$q = E_f \cdot F_q \cdot \tau, \quad (\text{A.21})$$

де E_f – середньо-поверхнева щільність теплового випромінювання полум'я факела, кВт/м². Через відсутність експериментальних даних величини середньо-поверхневої щільності теплового випромінювання полум'я E_f відповідно до вищенаведених рекомендацій приймаємо $E_f = 450 \text{ кВт/м}^2$.

τ – коефіцієнт пропускання атмосфери, який знаходять за формулою:

$$\tau = \exp\left[-0.0007 \cdot \left(\sqrt{r^2 + H^2} - 0.5 \cdot D_s\right)\right], \quad (\text{A.22})$$

де r – відстань від геометричного центру протоки до об'єкту, що опромінюється. Відстань від об'єкту, що опромінюється, до точки на поверхні землі безпосередньо під центром «вогненної кулі» приймаємо рівним $r = 500 \text{ м}$;

F_q – кутовий коефіцієнт опромінення, що характеризує, яка частка випромінювання від полум'я факела потрапляє на даний об'єкт.

Кутовий коефіцієнт опромінення розраховується як:

$$F_q = \frac{\frac{H}{D_s} + 0.5}{4 \cdot \left[\left(\frac{H}{D_s} + 0.5 \right)^2 + \left(\frac{r}{D_s} \right)^2 \right]^{1.5}}, \quad (\text{A.23})$$

де H – висота центру «вогняного центру», м; де D_s – ефективний діаметр «вогняного шару», м.

В свою чергу, висота центру «вогняного центру» визначається за формулою:

$$H = 0.5 \cdot D_s. \quad (\text{A.24})$$

Ефективний діаметр «вогняного шару» знаходять за формулою:

$$D_s = 5.33 \cdot m_T^{0.327}, \quad (\text{A.25})$$

де m_T – маса розлитого ГР або ЛЗР, кг. Маса розраховується як:

$$m_T = m_B \cdot Z, \quad (\text{A.26})$$

де Z – коефіцієнт участі пального у вибуху, згідно з [31] $Z = 0.1$.

Отримане значення інтенсивності теплового випромінювання на поверхні порівнюють з гранично допустимим значенням:

$$q_B \leq q_B^H, \quad (\text{A.27})$$

де q_B^H – гранично допустима інтенсивність теплового випромінювання, $q_B^H = 3.2 \cdot 10^5 \text{ Дж/м}^2$.

В разі, якщо отримане значення інтенсивності теплового випромінювання від «вогняного шару» на поверхні тіла людини перевищує гранично допустиму

інтенсивність теплового випромінювання, то при такому тепловому навантаженні людина без вогнезахисного комбінезона отримає опіки 3-го і 4-го ступеня. В іншому випадку, людина отримає опіки 1-го або 2-го ступеня.

ДОДАТОК Б. Об'єктно-орієнтована модель програмного забезпечення

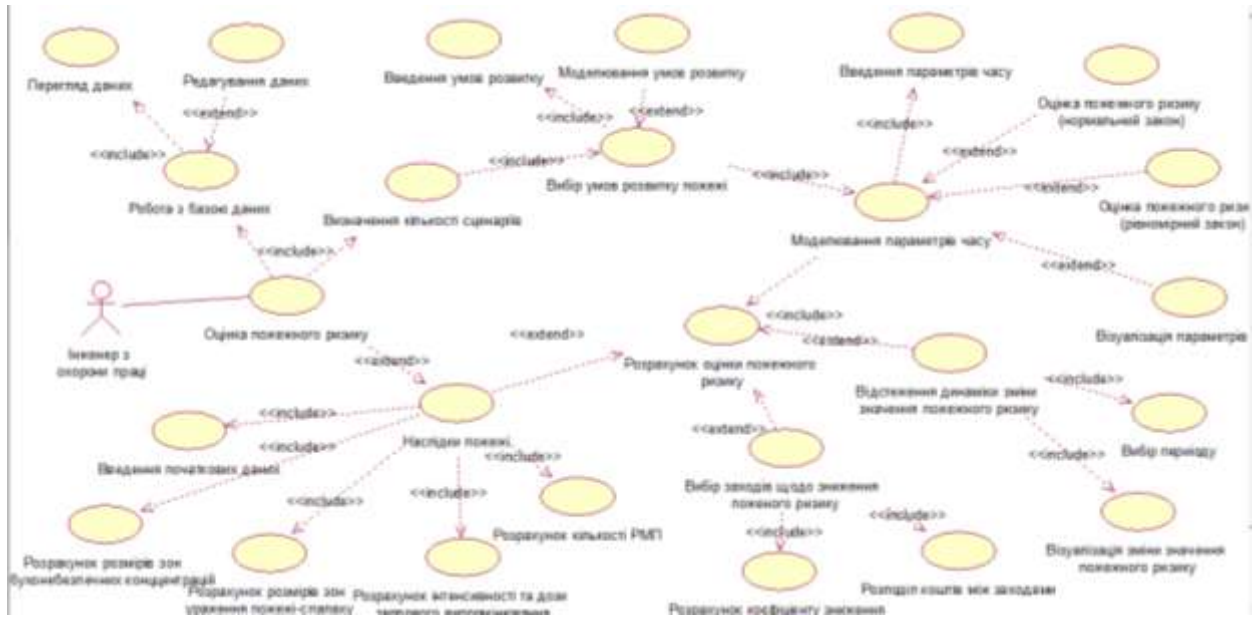


Рисунок Б.1 – Діаграма варіантів використання

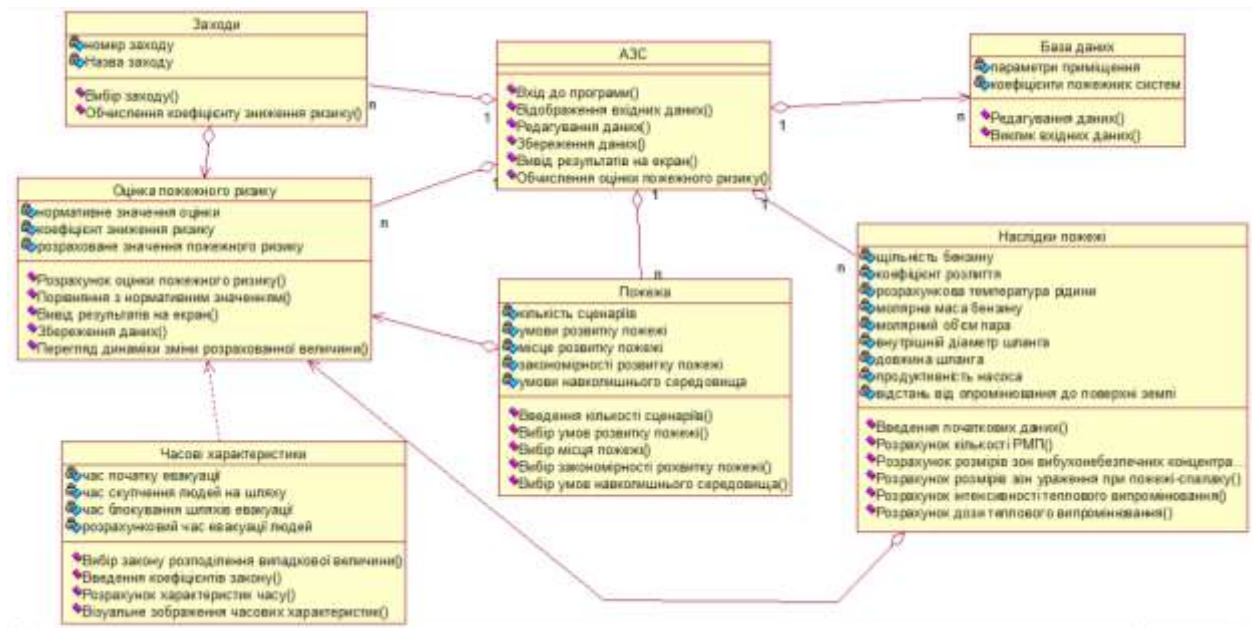


Рисунок Б.2 – Діаграма класів

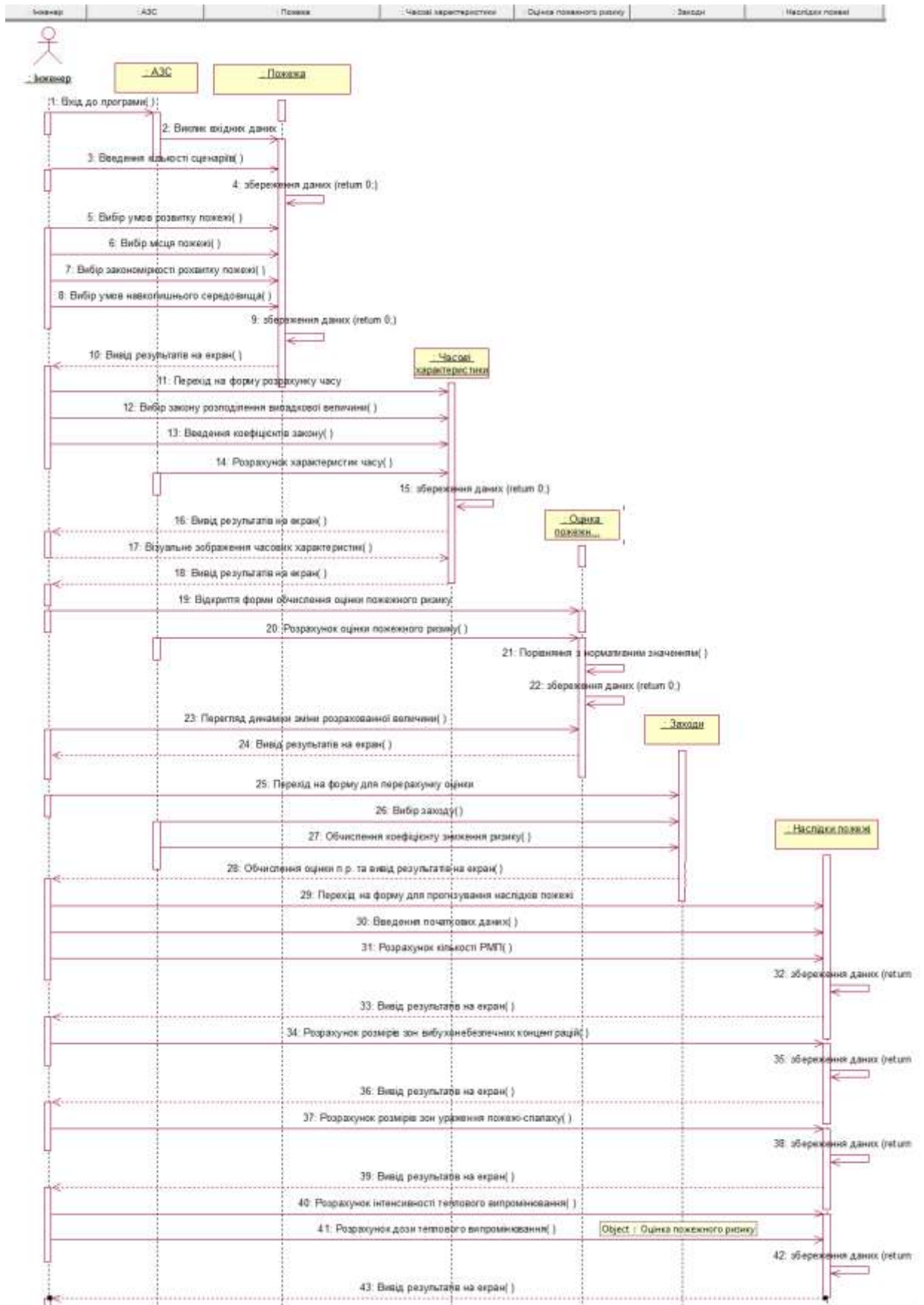


Рисунок Б.3 – Діаграма послідовності

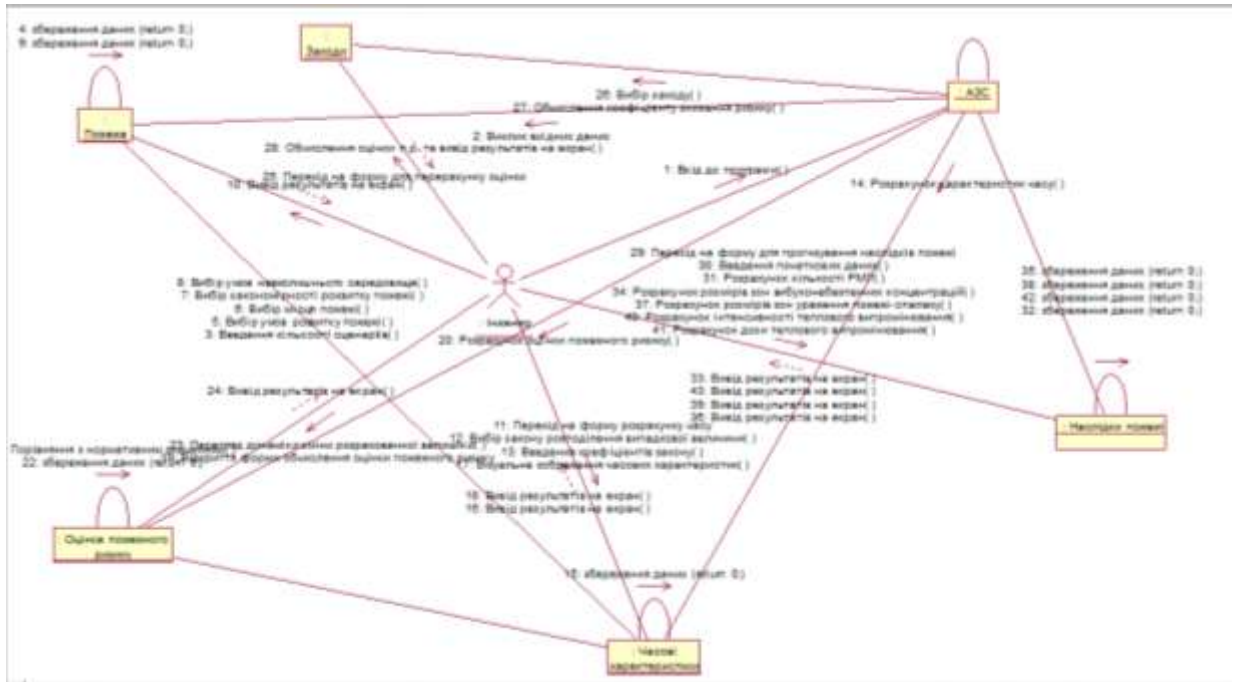


Рисунок Б.4 – Діаграма кооперації

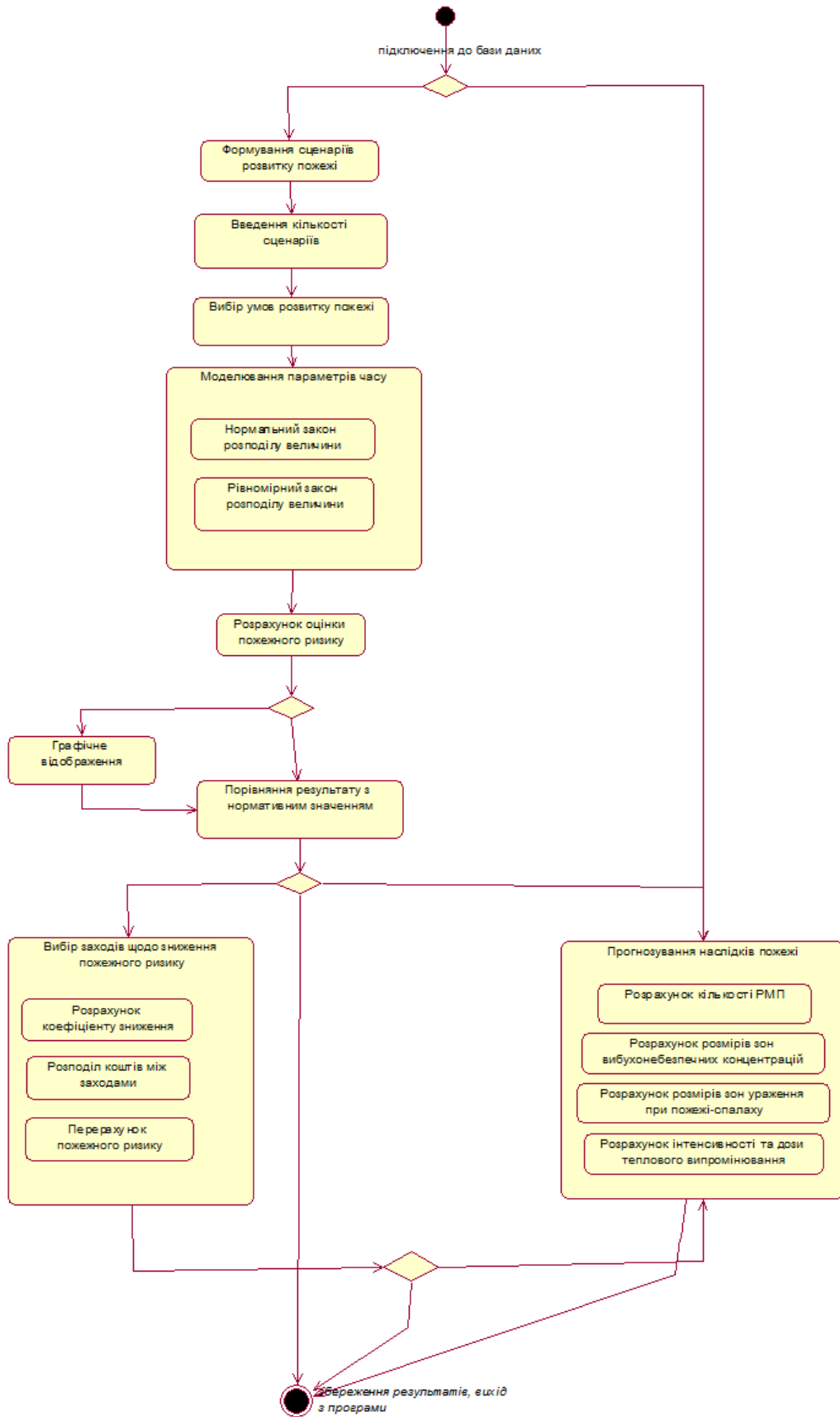


Рисунок Б.5 – Діаграма станів

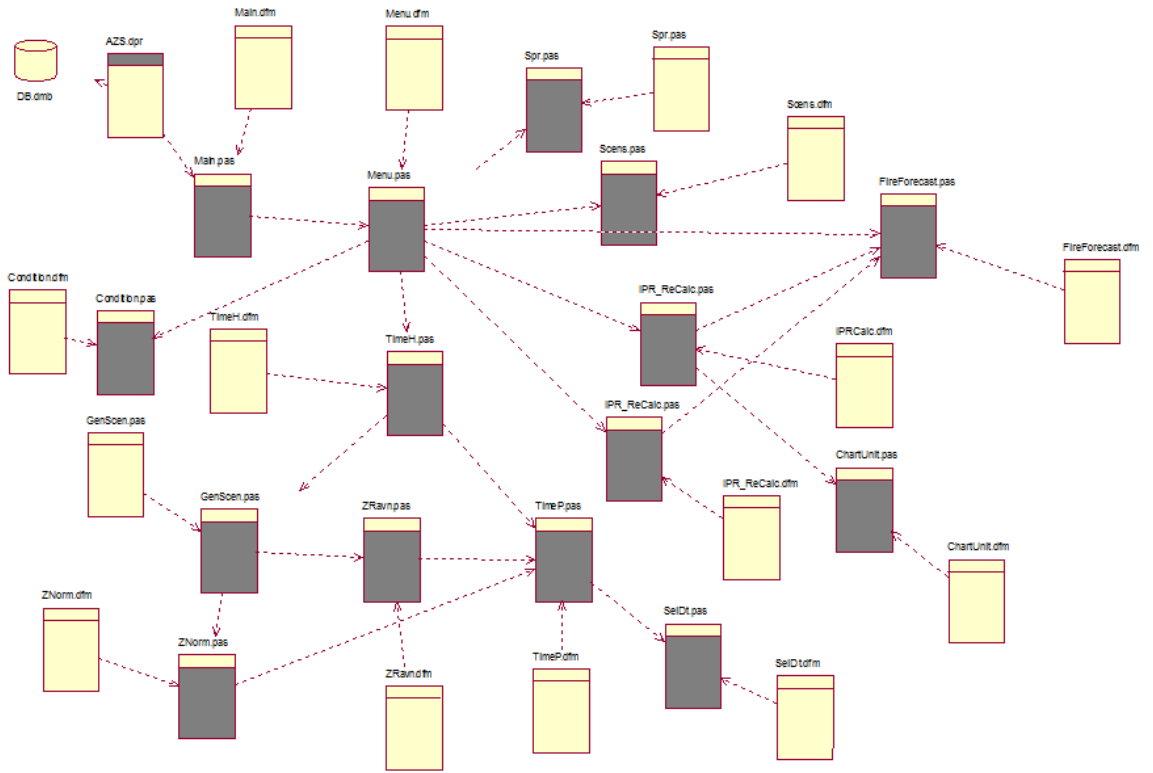


Рисунок Б.6 – Діаграма компонентів

ДОДАТОК В. Лістинг програми

```

unit Main;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
Vcl.ExtCtrls,
  Vcl.Imaging.pngimage, Menu;
type
  TMainForm = class(TForm)
    LogoImage: TImage;
    MainFormMemo: TMemo;
    Button1: TButton;
    procedure FormShow(Sender: TObject);
    procedure MainFormMemoEnter(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  MainForm: TMainForm;
implementation
{$R *.dfm}
//-----
procedure TMainForm.Button1Click(Sender: TObject);
begin
  Hide;
  MenuForm.Show;
end;
//-----
procedure TMainForm.FormShow(Sender: TObject);
begin
  {при открытии формы, делаю фон MainFormMemo
синим}
  MainFormMemo.Color := RGB(191, 205, 219);
end;
//-----
procedure TMainForm.MainFormMemoEnter(Sender:
TObject);
begin
  {передаю фокус на кнопку, при клике на
MainFormMemo}
  Button1.SetFocus;
end;
//-----
end.

unit Menu;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
Vcl.Imaging.pngimage,
  Vcl.ExtCtrls, Vcl.Menus, Scens, Condition, TimeH,
IPRCalc, IPR_ReCalc, Spr, Data.DB, Data.Win.ADODB;

type
  TMenuForm = class(TForm)
    LogoImage: TImage;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    N11: TMenuItem;
    N12: TMenuItem;
    N13: TMenuItem;
    N14: TMenuItem;
    Panel1: TPanel;
    MenuFormMemo: TMemo;
    DataSource1: TDataSource;
    ADOConnection1: TADOConnection;
    ADOQuery1: TADOQuery;
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    procedure MenuFormMemoEnter(Sender: TObject);
    procedure N4Click(Sender: TObject);
    procedure N5Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N9Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure N11Click(Sender: TObject);
    procedure N12Click(Sender: TObject);
    procedure N13Click(Sender: TObject);
    procedure N14Click(Sender: TObject);
    procedure N7Click(Sender: TObject);
    procedure N8Click(Sender: TObject);
  private
    { Private declarations }
    procedure TestConnection;
  public
    { Public declarations }
  end;
var
  MenuForm: TMenuForm;
implementation
uses
  Main;
{$R *.dfm}
//-----
procedure TMenuForm.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
  {при закрытии закрываю главное окно}
  MainForm.Close;
end;
//-----
procedure TMenuForm.TestConnection;
var DB_file : string;
begin
  DB_file := ExtractFileDir(Application.ExeName) +
'db.accdb';

  if Trim(ADOConnection1.ConnectionString) = " then
    ADOConnection1.ConnectionString :=
'Provider=Microsoft.ACE.OLEDB.12.0;Password='"; Data Source='
+ DB_file;

```

```

ADOConnection1.LoginPrompt := False;
DataSource1.DataSet := ADOQuery1;
ADOQuery1.Connection := ADOConnection1;

ADOQuery1.SQL.Text := 'Select count(*) AS ecc from
[Вхідні параметри - операції]';

try
  ADOQuery1.Open;
except
  ShowMessage('Неможливо виконати запит.
Перевірте зв'язок з базою даних.');
```

N4.Enabled := False;
N5.Enabled := False;
N6.Enabled := False;

```

  exit;
end;

N4.Enabled := True;

end;

//-----
procedure TMenuForm.FormShow(Sender: TObject);
begin
  MenuFormMemo.Color := RGB(240, 240, 240);
  TestConnection;
end;

//-----
procedure TMenuForm.MenuFormMemoEnter(Sender:
TObject);
begin
  {передаю фокус на панель, при кліке на
MainFormMemo}
  Panel1.SetFocus;
end;

//-----
procedure TMenuForm.N4Click(Sender: TObject);
begin
  ScensForm.Show;
  ScensForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N5Click(Sender: TObject);
begin
  Hide;
  ConditionForm.Show;
  ConditionForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N6Click(Sender: TObject);
begin
  Hide;
  TimeForm.Show;
  ConditionForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N7Click(Sender: TObject);
begin
  Hide;
  IPRCalcForm.Show;
  IPRCalcForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N8Click(Sender: TObject);
begin
  Hide;
  IPR_ReCalcForm.Show;
  IPR_ReCalcForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N9Click(Sender: TObject);
begin
  SprForm.Tag := 9;
  Hide;
  SprForm.Show;
  SprForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N10Click(Sender: TObject);
begin
  SprForm.Tag := 10;
  Hide;
  SprForm.Show;
  SprForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N11Click(Sender: TObject);
begin
  SprForm.Tag := 11;
  Hide;
  SprForm.Show;
  SprForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N12Click(Sender: TObject);
begin
  SprForm.Tag := 12;
  Hide;
  SprForm.Show;
  SprForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N13Click(Sender: TObject);
begin
  SprForm.Tag := 13;
  Hide;
  SprForm.Show;
  SprForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TMenuForm.N14Click(Sender: TObject);
begin
  SprForm.Tag := 14;
  Hide;
  SprForm.Show;
  SprForm.FormStyle := fsStayOnTop;
end;

//-----
unit Scens;
  {ввод кол-ва сценариев}
interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type
  TScensForm = class(TForm)
    Label1: TLabel;
    ScensEdit: TEdit;
    ScensButton: TButton;
    procedure ScensEditKeyPress(Sender: TObject; var
Key: Char);
    procedure ScensEditKeyUp(Sender: TObject; var Key:
Word;
      Shift: TShiftState);
    procedure ScensButtonClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
  private
    { Private declarations }
  public

```

```

    { Public declarations }
end;
var
    ScensForm: TScensForm;
implementation
uses Menu;
{$R *.dfm}

//-----
Action: TCloseAction);
procedure TScensForm.FormClose(Sender: TObject; var
begin
    MenuForm.Show;
    MenuForm.FormStyle := fsStayOnTop;
end;

//-----
TObject);
procedure TScensForm.ScensButtonClick(Sender:
begin
    if Trim(ScensEdit.Text) <> " then
    begin
        if MenuForm.Tag = 0 then
        begin
            MenuForm.ADOQuery1.SQL.Text := 'Insert into
[Вхідні параметри - операції] ([Кількість сценаріїв]' +
'Values (' + Trim(ScensEdit.Text) + ');';
            try
                {запоминаю число сценариев в таблицу
[Вхідні параметри - операції]}
                MenuForm.ADOQuery1.ExecSQL;

                MenuForm.ADOQuery1.SQL.Text := 'Select
Max([Код]) AS id from [Вхідні параметри - операції]';
                MenuForm.ADOQuery1.Open;

                {запоминаю текущий сеанс с базой}
                MenuForm.Tag :=
MenuForm.DataSource1.DataSet.FieldByName('id').AsInteger;
            except
                ShowMessage('Помилка в запиті до бази
даних.');
```

```

procedure TConditionForm.BackButtonClick(Sender:
Object);
begin
  Hide;
  //Close;
  MenuForm.N4.Click;
end;

//-----
procedure TConditionForm.ConditionButtonClick(Sender: TObject);
begin
  if RGroup.ItemIndex = -1 then
    exit;

  DSGrid.DataSet.Edit;
  DSGrid.DataSet.Post;

  if not CheckGrid then
    begin
      ShowMessage('Заповнені не всі параметри');
      exit;
    end;

  Hide;
  Close;
  MenuForm.N6.Enabled := True;
  MenuForm.N6.Click;
end;

//-----
procedure TConditionForm.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
  MenuForm.Show;
  MenuForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TConditionForm.FormShow(Sender: TObject);
begin
  {при откритті форми скриваю ґрид}
  RGroup.ItemIndex := -1;
  DBGrid1.Visible := false;

  ADOQueryGrid.CursorLocation := clUseServer;
end;

//-----
function TConditionForm.DelParams(id : String) :
boolean;
var res : boolean;
begin
  res := true;

  {удаляю временные параметры из [Вхідні параметри
- умови розвитку пожежі]}
  MenuForm.ADOQuery1.SQL.Text := 'Delete from
[Вхідні параметри - умови розвитку пожежі] ' +
  'where [id_operation] = ' + id;

  try
    MenuForm.ADOQuery1.ExecSQL;
  except
    ShowMessage('Помилка при роботі з базою');
    res := false;
  end;

  Result := res;
end;

//-----
function TConditionForm.AddRows(rows, empty :
integer) : boolean;
var
  n : integer;
  res : boolean;
begin
  res := true;
  try
    MenuForm.ADOQuery1.ExecSQL;
  except
    ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
    res := false;
    Result := res;
    exit;
  end;

  MenuForm.ADOQuery1.SQL.Text := 'select min([Код])
as n ' +
'from [Вхідні параметри - умови розвитку пожежі]
where [id_operation] = ' + IntToStr(MenuForm.Tag);
  MenuForm.ADOQuery1.Open;

  try
    MenuForm.ADOQuery1.Open;
  except
    ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
    res := false;
    Result := res;
    exit;
  end;

  MenuForm.ADOQuery1.SQL.Text := 'INSERT INTO
[Вхідні параметри - умови розвитку пожежі] ' +
'SELECT * FROM (SELECT TOP ' + IntToStr(rows) +
' Max([Вхідні параметри - операції].[Код]) AS id_operation, ' +
'[Місця виникнення пожежі].[№] AS id_place,
[Закономірності розповсюдження пожежі].[№] AS id_expansion, '
+
'[Початкова область пожежі].[№] AS id_start_region,
[Параметри навколишнього середовища].[№] AS id_params, ' +
'[Початкові параметри приміщення].[№] AS
id_start_params, 0 as scen_num ' +
'FROM [Вхідні параметри - операції], [Місця
виникнення пожежі], [Закономірності розповсюдження пожежі], '
+
'[Початкова область пожежі], [Параметри
навколишнього середовища], [Початкові параметри
приміщення] ' +
'GROUP BY [Місця виникнення пожежі].[№],
[Закономірності розповсюдження пожежі].[№], [Початкова
область пожежі].[№], ' +
'[Параметри навколишнього середовища].[№],
[Початкові параметри приміщення].[№] ' +
'ORDER BY Rnd([Місця виникнення пожежі].[№]),
Rnd([Закономірності розповсюдження пожежі].[№]), ' +
'Rnd([Початкова область пожежі].[№]),
Rnd([Параметри навколишнього середовища].[№]), ' +
'Rnd([Початкові параметри приміщення].[№])) AS
[%$##@_Alias]';

  else
    {добавляю пустые строки во [Вхідні параметри -
умови розвитку пожежі] по кол-ву rows}
    MenuForm.ADOQuery1.SQL.Text := 'INSERT INTO
[Вхідні параметри - умови розвитку пожежі] ' +
'SELECT * FROM (SELECT TOP ' + IntToStr(rows) +
' Max([Вхідні параметри - операції].[Код]) AS id_operation, 0 as
scen_num ' +
'FROM [Вхідні параметри - операції], [Місця
виникнення пожежі], [Закономірності розповсюдження пожежі], '
+
'[Початкова область пожежі], [Параметри
навколишнього середовища], [Початкові параметри
приміщення] ' +
'GROUP BY [Місця виникнення пожежі].[№],
[Закономірності розповсюдження пожежі].[№], [Початкова
область пожежі].[№], ' +
'[Параметри навколишнього середовища].[№],
[Початкові параметри приміщення].[№] ' +
'ORDER BY Rnd([Місця виникнення пожежі].[№]),
Rnd([Закономірності розповсюдження пожежі].[№]), ' +
'Rnd([Початкова область пожежі].[№]),
Rnd([Параметри навколишнього середовища].[№]), ' +
'Rnd([Початкові параметри приміщення].[№])) AS
[%$##@_Alias]';

  end;
end;

```

```

n :=
MenuForm.ADOQuery1.FieldByName('n').AsInteger;
except
  ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
res := false;
Result := res;
exit;
end;

MenuForm.ADOQuery1.SQL.Text := 'update [Вхідні
параметри - умови розвитку пожежі] ' +
'set [scen_num] = ([Код] - ' + IntToStr(n) + ' + 1)
where [id_operation] = ' + IntToStr(MenuForm.Tag);

try
  MenuForm.ADOQuery1.ExecSQL;

except
  ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
res := false;
Result := res;
exit;
end;

Result := res;
end;

//-----
procedure
TConditionForm.CreatePersistentFields(ADataset: TDataset);
Var
  i: Integer;
Begin
  ADataset.FieldDefs.Update;
  for i := 0 to ADataset.FieldDefs.Count - 1 do
    if ADataset.FindField(ADataset.FieldDefs[i].Name) =
Nil then
      ADataset.FieldDefs.Items[i].CreateField(ADataset);
End;

//-----
procedure TConditionForm.CreateLookupField( ATable:
TDataSet; AFieldName: String; ALookupDataset: TDataset;
  AKeyfields: String; ALookupKeyfields: String;
ALookupResultField : String);
Var
  NewField : TField;
Begin
  with ATable do begin
    if FieldDefs.Updated = False then
      FieldDefs.Update;
    If FindField(AFieldName) = Nil then
      begin
        NewField := TStringField.Create(ATable);
        NewField.FieldName := AFieldName;
        NewField.KeyFields := AKeyFields;
        NewField.LookupDataSet := ALookupDataset;
        NewField.LookupKeyFields := ALookupKeyFields;
        NewField.LookupResultField :=
ALookupResultField;
        NewField.FieldKind := fkLookup;
        NewField.Dataset := ATable;
        NewField.Size := 255;
      end;
    end;
  End;

//-----
procedure TConditionForm.InitGrid;
begin
  DBGrid1.Columns[0].ReadOnly := true;

  DBGrid1.Columns[1].Visible := false;
  DBGrid1.Columns[2].Visible := false;
  DBGrid1.Columns[3].Visible := false;
  DBGrid1.Columns[4].Visible := false;
  DBGrid1.Columns[5].Visible := false;
  DBGrid1.Columns[0].Title.Caption := 'Сценарій';
  DBGrid1.Columns[6].Title.Caption := 'Місце
виникнення пожежі';
  DBGrid1.Columns[7].Title.Caption := 'Закономірності
розповсюдження';
  DBGrid1.Columns[8].Title.Caption := 'Початкова
область';
  DBGrid1.Columns[9].Title.Caption := 'Параметри
середовища';
  DBGrid1.Columns[10].Title.Caption := 'Параметри
приміщення';

  (DBGrid1.Columns[0] as TColumn).Width := 80;
  (DBGrid1.Columns[6] as TColumn).Width := 210;
  (DBGrid1.Columns[7] as TColumn).Width := 350;
  (DBGrid1.Columns[8] as TColumn).Width := 100;
  (DBGrid1.Columns[9] as TColumn).Width := 180;
  (DBGrid1.Columns[10] as TColumn).Width := 180;
  DBGrid1.Visible := true;
end;

//-----
function TConditionForm.QueryExec(AQ: TADOQuery)
: boolean;
var res : boolean;
begin
  {процедура запускає запити з виводом помилок}
  res := true;

  try
    AQ.Open;
  except
    ShowMessage('Помилка в запиті: ' + AQ.SQL.Text);
    res := false;
  end;

  Result := res;
end;

//-----
procedure TConditionForm.RGroupClick(Sender:
TObject);
var r : integer;
begin
  if not DelParams(IntToStr(MenuForm.Tag)) then
    exit;

    MenuForm.ADOQuery1.SQL.Text := 'SELECT
[Кількість сценаріїв] AS row_number ' +
'FROM [Вхідні параметри - операції] ' +
'WHERE [Код] = ' + IntToStr(MenuForm.Tag) + ';';

    try
      MenuForm.ADOQuery1.Open;

      {кол-во сценаріїв}
      r :=
MenuForm.DataSource1.DataSet.FieldByName('row_number').AsInt
eger;

    except
      ShowMessage('Помилка при роботі з базою');
      exit;
    end;

    if RGroup.ItemIndex > 0 then
      begin
        {випадково}
        if not AddRows(r, 0) then
          exit;

          DBGrid1.ReadOnly := true;
        end
      else
        begin
          {користувач}
          if not AddRows(r, 1) then

```

```

        exit;

        DBGrid1.ReadOnly := false;
    end;

    {входящие параметры сохраняю в базу }
    MenuForm.ADOQuery1.SQL.Text := 'Update [Вхідні
параметри - операції] ' +
    'set [Вибір УПП] = ' + IntToStr(RGroup.ItemIndex) +
    ' WHERE [Код] = ' + IntToStr(MenuForm.Tag) + ';';

    try
        MenuForm.ADOQuery1.ExecSQL;
    except
        ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
    end;

    {основной датасет}
    ADOQueryGrid.Connection :=
MenuForm.ADOConnection1;

    DBGrid1.DataSource := DSGrid;

    ADOQueryGrid.SQL.Text := 'Select scen_num,
id_place, id_expansion, id_start_region, id_params, id_start_params ' +
    'from [Вхідні параметри - умови розвитку пожежі] ' +
    'Where id_operation = ' + IntToStr(MenuForm.Tag) + '
order by [Код];';

    if not QueryExec(ADOQueryGrid) then
        exit;

    DSGrid.DataSet := ADOQueryGrid;

    {5 справочников}

    DS1.DataSet := ADOQuery1;
    DS2.DataSet := ADOQuery2;
    DS3.DataSet := ADOQuery3;
    DS4.DataSet := ADOQuery4;
    DS5.DataSet := ADOQuery5;

    ADOQuery1.Connection :=
MenuForm.ADOConnection1;
    ADOQuery2.Connection :=
MenuForm.ADOConnection1;
    ADOQuery3.Connection :=
MenuForm.ADOConnection1;
    ADOQuery4.Connection :=
MenuForm.ADOConnection1;
    ADOQuery5.Connection :=
MenuForm.ADOConnection1;

    ADOQuery1.SQL.Text := 'Select [№] as id, [Можливий
варіант] as name ' +
    'from [Місця виникнення пожежі] ' +
    'order by [№];';

    ADOQuery2.SQL.Text := 'Select [№] as id, [Можливий
варіант] as name ' +
    'from [Закономірності розповсюдження пожежі] ' +
    'order by [№];';

    ADOQuery3.SQL.Text := 'Select [№] as id, [Можливий
варіант] as name ' +
    'from [Початкова область пожежі] ' +
    'order by [№];';

    ADOQuery4.SQL.Text := 'Select [№] as id, [Можливий
варіант] as name ' +
    'from [Параметри навколишнього середовища] ' +
    'order by [№];';

    ADOQuery5.SQL.Text := 'Select [№] as id, [Можливий
варіант] as name ' +

'from [Початкові параметри приміщення] ' +
'order by [№];';

    if not QueryExec(ADOQuery1) then
        exit;
    if not QueryExec(ADOQuery2) then
        exit;
    if not QueryExec(ADOQuery3) then
        exit;
    if not QueryExec(ADOQuery4) then
        exit;
    if not QueryExec(ADOQuery5) then
        exit;

    {добавляю к основному датасету LookUp-поля
(выпад. списки)}
    DSGrid.DataSet.Close;

    CreatePersistentFields(TDataset(DSGrid.DataSet));
    CreateLookupField(TDataset(DSGrid.DataSet), 'spr1',
TDataset(DS1.DataSet), 'id_place', 'id', 'name');

    CreatePersistentFields(TDataset(DSGrid.DataSet));
    CreateLookupField(TDataset(DSGrid.DataSet), 'spr2',
TDataset(DS2.DataSet), 'id_expansion', 'id', 'name');

    CreatePersistentFields(TDataset(DSGrid.DataSet));
    CreateLookupField(TDataset(DSGrid.DataSet), 'spr3',
TDataset(DS3.DataSet), 'id_start_region', 'id', 'name');

    CreatePersistentFields(TDataset(DSGrid.DataSet));
    CreateLookupField(TDataset(DSGrid.DataSet), 'spr4',
TDataset(DS4.DataSet), 'id_params', 'id', 'name');
    CreatePersistentFields(TDataset(DSGrid.DataSet));
    CreateLookupField(TDataset(DSGrid.DataSet), 'spr5',
TDataset(DS5.DataSet), 'id_start_params', 'id', 'name');

    DSGrid.DataSet.Open;

    InitGrid;

end;

//-----
function TConditionForm.CheckGrid : boolean;
var res : boolean;
begin
    res := false;

    {проверяю на пустые параметры }

    MenuForm.ADOQuery1.SQL.Text := 'SELECT
Count(*) AS ccc ' +
    'FROM [Вхідні параметри - умови розвитку пожежі] ' +
    'WHERE [id_operation] = ' + IntToStr(MenuForm.Tag)
+ ' AND ([id_place] = 0 ' +
    'OR [id_expansion] = 0 OR [id_start_region] = 0 OR
[id_params] = 0 OR [id_start_params] = 0);';

    try
        MenuForm.ADOQuery1.Open;

        if
MenuForm.ADOQuery1.FieldByName('ccc').AsInteger > 0 then
            res := false
        else
            res := true;

        except
            ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
        end;

        Result := res;
    end;
end;

```



```

-----
//-----
end.

unit TimeH;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
  System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
  Vcl.ExtCtrls, GenScen, TimeP;

type
  TTimeForm = class(TForm)
    TimeFormMemo: TMemo;
    RGroup: TRadioGroup;
    BackButton: TButton;
    TimeFormButton: TButton;
    procedure FormShow(Sender: TObject);
    procedure TimeFormMemoEnter(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
  TCloseAction);
    procedure BackButtonClick(Sender: TObject);
    procedure TimeFormButtonClick(Sender: TObject);
  private
    { Private declarations }
  public
    function AddScensToTimeParams(n : integer) :
  boolean;
    { Public declarations }
  end;

var
  TimeForm: TTimeForm;

implementation
  uses Menu;

  {$R *.dfm}

//-----
-----

  procedure TTimeForm.BackButtonClick(Sender:
  TObject);
  begin
    Hide;
    // Close;
    MenuForm.N5.Click;
  end;

//-----
-----

  procedure TTimeForm.FormClose(Sender: TObject; var
  Action: TCloseAction);
  begin
    MenuForm.Show;
    MenuForm.FormStyle := fsStayOnTop;
  end;

//-----
-----

  procedure TTimeForm.FormShow(Sender: TObject);
  begin
    TimeFormMemo.Color := RGB (240, 240, 240);
    TimeFormMemo.Font.Color := RGB (133, 10, 10);
  end;

//-----
-----

  function TTimeForm.AddScensToTimeParams(n :
  integer) : boolean;
  var
    res : boolean;
  begin
    res := true;

    MenuForm.ADOQuery1.SQL.Text := 'Insert into
  [Вхідні параметри - часові характеристики]' +
    '(id_operation, scen_num) Values (' +
    IntToStr(MenuForm.Tag) + ', ' + IntToStr(n) + ')';

    try
      MenuForm.ADOQuery1.ExecSQL;
    except
      ShowMessage('Помилка в запиті: ' +
  MenuForm.ADOQuery1.SQL.Text);
      res := false;
    end;

    Result := res;
  end;

//-----
-----

  procedure TTimeForm.TimeFormButtonClick(Sender:
  TObject);
  var i, n : integer;
  next_form : boolean;
  begin
    if RGroup.ItemIndex = -1 then
      exit;

    {входящие параметры временных ха-рик сохраняю в
  базу }
    MenuForm.ADOQuery1.SQL.Text := 'Update [Вхідні
  параметри - операції]' +
    'set [Вибір ЧХ] = ' + IntToStr(RGroup.ItemIndex) +
    ' WHERE [Код] = ' + IntToStr(MenuForm.Tag) + ';';

    try
      MenuForm.ADOQuery1.ExecSQL;
    except
      ShowMessage('Помилка в запиті: ' +
  MenuForm.ADOQuery1.SQL.Text);
      exit;
    end;

    if RGroup.ItemIndex = 1 then
      begin
        GenForm.Show;
        GenForm.FormStyle := fsStayOnTop;
      end
    else
      begin
        next_form := true;

        {выбираю кол-во сценариев }
        MenuForm.ADOQuery1.SQL.Text := 'Select [Кількість
  сценаріїв] as ccc from [Вхідні параметри - операції]' +
        'where [Код] = ' + IntToStr(MenuForm.Tag);

        try
          MenuForm.ADOQuery1.Open;
          n :=
  MenuForm.ADOQuery1.FieldByName('ccc').AsInteger;
        except
          ShowMessage('Помилка в запиті: ' +
  MenuForm.ADOQuery1.SQL.Text);
          exit;
        end;

        {удаляю возможные сценарии, сгенерированные
  случайно}
        MenuForm.ADOQuery1.SQL.Text := 'Delete from
  [Вхідні параметри - часові характеристики]' +
        'Where id_operation = ' + IntToStr(MenuForm.Tag);
      end;
    end;
  end;

```

```

try
  MenuForm.ADOQuery1.ExecSQL;
except
  ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
exit;
end;

for i := 1 to n do
begin
  if not AddScensToTimeParams(i) then
  begin
    next_form := false;
    break;
  end;

  Application.ProcessMessages;
end;

if not next_form then
  exit;

TimeParamForm.Tag := 2;
TimeParamForm.Show;
TimeParamForm.FormStyle := fsStayOnTop;

end;

Hide;

end;

//-----
procedure TTimeForm.TimeFormMemoEnter(Sender:
TObject);
begin
  RGroup.SetFocus;
end;

//-----
end.

unit TimeP;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB,
Vcl.Grids, Vcl.DBGrids, ChartUnit,
  Vcl.StdCtrls, Data.Win.ADODB, SelDt;

type
  TTimeParamForm = class(TForm)
    TimeParamMemo: TMemo;
    Label13: TLabel;
    BackButton: TButton;
    TimeParamButton: TButton;
    DBGrid1: TDBGrid;
    GroupBox1: TGroupBox;
    LoadButton: TButton;
    Gr1Button: TButton;
    Gr2Button: TButton;
    Gr3Button: TButton;
    Gr4Button: TButton;
    ADOQueryGrid: TADOQuery;
    DSGrid: TDataSource;
  procedure BackButtonClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action:
TCloseAction);
  procedure FormShow(Sender: TObject);
  procedure TimeParamMemoEnter(Sender: TObject);
  procedure TimeParamButtonClick(Sender: TObject);
  procedure Gr1ButtonClick(Sender: TObject);
  procedure Gr2ButtonClick(Sender: TObject);
  procedure Gr3ButtonClick(Sender: TObject);
  procedure Gr4ButtonClick(Sender: TObject);
private
  { Private declarations }
  function CheckGrid : boolean;
public
  { Public declarations }
end;

var
  TimeParamForm: TTimeParamForm;

implementation
uses TimeH, ZNorm, ZRavn, Menu;
{$R *.dfm}

//-----
procedure TTimeParamForm.BackButtonClick(Sender:
TObject);
begin
  Hide;
  //Close;

  if Self.Tag = 0 then
  begin
    ZakonNormForm.Show;
    ZakonNormForm.FormStyle := fsStayOnTop;
  end
  else
  if Self.Tag = 1 then
  begin
    ZakonRavnForm.Show;
    ZakonRavnForm.FormStyle := fsStayOnTop;
  end
  else
  if Self.Tag = 2 then
  begin
    TimeForm.Show;
    TimeForm.FormStyle := fsStayOnTop;
  end;
end;

//-----
-----
procedure TTimeParamForm.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
  MenuForm.Show;
  MenuForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TTimeParamForm.FormShow(Sender:
TObject);
begin
  {выборка данных при открытии формы}
  ADOQueryGrid.CursorLocation := clUseServer;
  ADOQueryGrid.Connection :=
MenuForm.ADOConnection1;

  if Self.Tag < 2 then
  begin
    GroupBox1.Visible := true;
    DBGrid1.ReadOnly := true;
    Label13.Visible := false;
    LoadButton.Visible := false;
    BackButton.Left := 305;

    ADOQueryGrid.SQL.Text := 'Select scen_num, start,
block, crowd, calc ' +
'from [Вхідні параметри - часові характеристики]' +
'Where id_operation = (select [Код] from [Вхідні
параметри - операції] where [Код] = ' +
IntToStr(MenuForm.Tag) + ' and [Вибір ЧХ] = 1 and
[Вибір закону] = ' + IntToStr(Self.Tag + 1) +
') order by [scen_num];';
  end
  else

```

```

begin
  GroupBox1.Visible := false;
  DBGrid1.ReadOnly := false;
  Label13.Visible := true;
  LoadButton.Visible := true;
  BackButton.Left := 111;

  ADOQueryGrid.SQL.Text := 'Select scen_num, start,
block, crowd, calc ' +
  'from [Вхідні параметри - часові характеристики] ' +
  'Where id_operation = (select [Код] from [Вхідні
параметри - операції] where [Код] = ' +
  IntToStr(MenuForm.Tag) + ' and [Вибір ЧХ] = 0) order
by [scen_num];';
end;

TimeParamMemo.Color := RGB (240, 240, 240);
TimeParamMemo.Font.Color := RGB (133, 10, 10);

DBGrid1.DataSource := DSGrid;

try
  ADOQueryGrid.Open;
except
  ShowMessage('Помилка в запиті: ' +
ADOQueryGrid.SQL.Text);
  exit;
end;

DSGrid.DataSet := ADOQueryGrid;

TBCDField(DSGrid.DataSet.FieldByName('start')).DisplayFormat :=
'0.0000';

TBCDField(DSGrid.DataSet.FieldByName('block')).DisplayFormat
:= '0.0000';

TBCDField(DSGrid.DataSet.FieldByName('crowd')).DisplayFormat
:= '0.0000';

TBCDField(DSGrid.DataSet.FieldByName('calc')).DisplayFormat :=
'0.0000';

DBGrid1.Columns[0].ReadOnly := true;

DBGrid1.Columns[0].Title.Caption := 'Сценарій';
DBGrid1.Columns[1].Title.Caption := 'tпе';
DBGrid1.Columns[2].Title.Caption := 'tбл';
DBGrid1.Columns[3].Title.Caption := 'tср';
DBGrid1.Columns[4].Title.Caption := 'tp';

(DBGrid1.Columns[0] as TColumn).Width := 50;
(DBGrid1.Columns[1] as TColumn).Width := 50;
(DBGrid1.Columns[2] as TColumn).Width := 50;
(DBGrid1.Columns[3] as TColumn).Width := 50;
(DBGrid1.Columns[4] as TColumn).Width := 50;
end;

//-----
procedure TTimeParamForm.Gr1ButtonClick(Sender:
TObject);
begin
  Hide;

  ChartForm.Tag := 1;
  ChartForm.Show;
  ChartForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TTimeParamForm.Gr2ButtonClick(Sender:
TObject);
begin
  Hide;

  ChartForm.Tag := 2;
  ChartForm.Show;
  ChartForm.FormStyle := fsStayOnTop;
end;

end;

//-----
procedure TTimeParamForm.Gr3ButtonClick(Sender:
TObject);
begin
  Hide;

  ChartForm.Tag := 3;
  ChartForm.Show;
  ChartForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TTimeParamForm.Gr4ButtonClick(Sender:
TObject);
begin
  Hide;

  ChartForm.Tag := 4;
  ChartForm.Show;
  ChartForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TTimeParamForm.LoadButtonClick(Sender:
TObject);
var n : integer;
begin
  Hide;

  n := DSGrid.DataSet.RecordCount;

  SelDatesForm.Label1.Caption := 'Кількість сценаріїв '
+ IntToStr(n) + ' ';
  SelDatesForm.Tag := n;
  SelDatesForm.Show;
  SelDatesForm.FormStyle := fsStayOnTop;
end;

//-----
function TTimeParamForm.CheckGrid : boolean;
var res : boolean;
begin

  res := false;

  {перевіряю на пустые параметри }

  MenuForm.ADOQuery1.SQL.Text := 'SELECT
Count(*) AS ccc ' +
  'FROM [Вхідні параметри - часові характеристики] '
+
  'WHERE [id_operation] = ' + IntToStr(MenuForm.Tag)
+
  ' AND ([start] < 0.01 OR [block] < 0.01 OR [crowd] <
0.01 OR [calc] < 0.01) OR (' +
  '[start] > 60 OR [block] > 60 OR [crowd] > 60 OR [calc]
> 60);';

  try
    MenuForm.ADOQuery1.Open;

    if
      MenuForm.ADOQuery1.FieldByName('ccc').AsInteger > 0 then
      res := false
    else
      res := true;

  except
    ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
  end;

  Result := res;
end;

//-----

```

```

procedure
TTimeParamForm.TimeParamButtonClick(Sender: TObject);
var Kobjn, Ksou, Kpdz, Kpz : real;
    Kstr : String;
begin
//проверка на ввод значений, если грид доступен для
редактирования
if not DBGrid1.ReadOnly then
begin
FormatSettings.DecimalSeparator := ',';

DSGrid.DataSet.Edit;
DSGrid.DataSet.Post;

{
if not CheckGrid then
begin
ShowMessage('Заповнені не всі параметри');
exit;
end;
}

end;

MenuForm.ADOQuery1.SQL.Text := 'SELECT Top 1
[Кобн] as k1, [Ксоуе] as k2, [Кпдз] as k3 from [Дані]';

// Memo1.Text := MenuForm.ADOQuery1.SQL.Text;
try
MenuForm.ADOQuery1.Open;
except
ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
exit;
end;

Kobjn :=
MenuForm.ADOQuery1.FieldByName('k1').AsFloat;
Ksou :=
MenuForm.ADOQuery1.FieldByName('k2').AsFloat;
Kpdz :=
MenuForm.ADOQuery1.FieldByName('k3').AsFloat;

Kpz := 1 - (1 - Kobjn * Ksou) * (1 - Kobjn * Kpdz);

{
сохраняю вероятность эвакуации людей в [Вхідні
параметри - часові характеристики].[Pe],
коэф. в [Вхідні параметри - часові
характеристики].[Kpz]

if (tp < 0.8 * тбл) and (0.8 * тбл < (tp + tpe)) and (tck
<= 6 мин)
Pei = 0.8 * тбл / tp
else
if ((tp + tpe) <= 0.8 * тбл) and (tck <= 6 мин)
Pei = 0.999
else
if ( tp >= 0.8 * тбл) or (tck > 6 мин)
Pei = 0.999
}

MenuForm.ADOQuery1.SQL.Text := 'UPDATE [Вхідні
параметри - часові характеристики] AS vh ' +
'SET [Pe] = If((vh.calc<0.8*vh.block) And
(0.8*vh.block<(vh.calc+vh.start) And
(vh.crowd<=6)),(0.8*vh.block/vh.calc), ' +
'(If(((vh.calc+vh.start)<=0.8*vh.block) And
(vh.crowd<=6)),0.999,If(((vh.calc>=0.8*vh.block)
(vh.crowd>6)),0,0))))), ' +
'[Kpz] = ' + FloatToStr(Kpz) +
' WHERE vh.id_operation = ' +
IntToStr(MenuForm.Tag);

//Memo1.Text := MenuForm.ADOQuery1.SQL.Text;

try
MenuForm.ADOQuery1.ExecSQL;
except
ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
exit;
end;

MenuForm.N7.Enabled := true;
MenuForm.N8.Enabled := false;
Close;
end;

//-----

procedure
TTimeParamForm.TimeParamMemoEnter(Sender: TObject);
begin
TimeParamButton.SetFocus;
end;

//-----

end.

unit GenScen;

interface

uses
Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
Vcl.ExtCtrls, ZNorm, ZRavn;

type
TGenForm = class(TForm)
BackButton: TButton;
RGroup: TRadioGroup;
GenFormButton: TButton;
GenFormMemo: TMemo;
procedure GenFormButtonClick(Sender: TObject);
procedure BackButtonClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action:
TCloseAction);
procedure FormShow(Sender: TObject);
procedure GenFormMemoEnter(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
GenForm: TGenForm;

implementation
uses Menu;
{$R *.dfm}

//-----
procedure TGenForm.BackButtonClick(Sender: TObject);
begin
Hide;
// Close;
MenuForm.N6.Click;
end;

//-----
procedure TGenForm.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
MenuForm.Show;
MenuForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TGenForm.FormShow(Sender: TObject);
begin
GenFormMemo.Color := RGB (240, 240, 240);
GenFormMemo.Font.Color := RGB (133, 10, 10);

```

```

end;

//-----
procedure TGenForm.GenFormButtonClick(Sender:
TObject);
begin
if RGroup.ItemIndex = -1 then
    exit;

    {входящие параметры временных ха-рик сохраняю
в базу }
MenuForm.ADOQuery1.SQL.Text := 'Update [Вхідні
параметри - операції] ' +
'set [Вибір закону] = ' + IntToStr(RGroup.ItemIndex +
1) +
' WHERE [Код] = ' + IntToStr(MenuForm.Tag) + ';';

try
    MenuForm.ADOQuery1.ExecSQL;
except
    ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
    exit;
end;

Hide;

if RGroup.ItemIndex = 0 then
begin
    ZakonNormForm.Show;
    GenForm.FormStyle := fsStayOnTop;
end
else
begin
    ZakonRavnForm.Show;
    GenForm.FormStyle := fsStayOnTop;
end;
end;

//-----
procedure TGenForm.GenFormMemoEnter(Sender:
TObject);
begin
    RGroup.SetFocus;
end;

//-----
end.

unit SelDt;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
Vcl.ComCtrls, Data.DB,
Data.Win.ADODB, Vcl.Grids, Vcl.DBGrids, DateUtils;

type
    TSelDatesForm = class(TForm)
        Label1: TLabel;
        SelDateMemo: TMemo;
        dtPicker1: TDateTimePicker;
        dtPicker2: TDateTimePicker;
        DBGrid1: TDBGrid;
        ADOQuery1: TADOQuery;
        DataSource1: TDataSource;
        Label2: TLabel;
        procedure FormClose(Sender: TObject; var Action:
TCloseAction);
        procedure FormShow(Sender: TObject);
        procedure SelDateMemoEnter(Sender: TObject);
        procedure dtPicker1Change(Sender: TObject);
        procedure dtPicker2Change(Sender: TObject);
        procedure DBGrid1CellClick(Column: TColumn);
    private
        { Private declarations }
        procedure SelectQuery(dt1, dt2 :String);
    public
        { Public declarations }
    end;

var
    SelDatesForm: TSelDatesForm;

implementation

uses TimeP, Menu;
{$R *.dfm}

//-----
procedure TSelDatesForm.DBGrid1CellClick(Column:
TColumn);
var kod : String;
begin
    if DataSource1.DataSet.RecordCount > 0 then
        begin
            kod
            :=
            DataSource1.DataSet.Fields.Fields[0].AsString;

            MenuForm.ADOQuery1.SQL.Text := 'UPDATE
[Вхідні параметри - часові характеристики] AS a ' +
'INNER JOIN [Вхідні параметри - часові
характеристики] AS b ON a.scen_num = b.scen_num ' +
'SET a.start = b.start, a.block = b.block, a.crowd =
b.crowd, a.calc = b.calc ' +
'WHERE a.[id_operation] = ' +
IntToStr(MenuForm.Tag) + ' and b.[id_operation] = ' + kod;

            try
                MenuForm.ADOQuery1.ExecSQL;
            except
                ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
                exit;
            end;

            Close;

        end;

    end;

//-----
procedure
TSelDatesForm.dtPicker1Change(Sender: TObject);
begin
    SelectQuery(DateToStr(dtPicker1.Date),
DateToStr(dtPicker2.Date));
end;

//-----

procedure TSelDatesForm.dtPicker2Change(Sender:
TObject);
begin
    SelectQuery(DateToStr(dtPicker1.Date),
DateToStr(dtPicker2.Date));
end;

//-----

procedure TSelDatesForm.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
    TimeParamForm.Show;
    TimeParamForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TSelDatesForm.FormShow(Sender: TObject);
begin
    SelDateMemo.Color := RGB (240, 240, 240);
    SelDateMemo.Font.Color := RGB (133, 10, 10);

```

```

Label1.Color := RGB (240, 240, 240);

{ограничиваю выбор дат}
dtPicker1.MinDate := StrToDate('01.05.2018');
dtPicker2.MinDate := StrToDate('01.05.2018');
dtPicker1.MaxDate := Date; //сегодня
dtPicker2.MaxDate := IncDay(Date, +1); //завтра

dtPicker1.Date := IncDay(Date, -1); //вчера
dtPicker2.Date := Date;

{соединяюсь с БД}
ADOQuery1.Connection :=
MenuForm.ADOConnection1;
DataSource1.DataSet := ADOQuery1;
DBGrid1.DataSource := DataSource1;
DBGrid1.ReadOnly := true;

SelectQuery(DateToStr(dtPicker1.Date),
DateToStr(dtPicker2.Date));
end;

-----
//-----
procedure TSelDatesForm.SelectQuery(dt1, dt2 :String);
begin

    ADOQuery1.SQL.Text := 'select [Код], [dt] as [Дата та
час], [Кількість сценаріїв] from ' +
' [Вхідні параметри - операції]' +
' where [dt] >= Format("'" + dt1 + "',"dd.mm.yyyy") and
[dt] <= Format("'" + dt2 + "',"dd.mm.yyyy")' +
' and [dt] >= Format("'" + DateToStr(dtPicker1.MinDate)
+ "',"dd.mm.yyyy")' +
' and [dt] <= Format("'" + DateToStr(dtPicker1.MaxDate)
+ "',"dd.mm.yyyy")' +
' and [Кількість сценаріїв] = ' + IntToStr(Self.Tag) +
' and [Код] <> ' + IntToStr(MenuForm.Tag) +
' and [Код] in (select [id_operation] from [Вхідні
параметри - часові характеристики])' +
' // group by [dt], [Кількість сценаріїв]' +
' order by [dt] desc;';

    try
        ADOQuery1.Open;
    except
        Label2.Visible := false;
        ShowMessage('Помилка в запиті: ' +
ADOQuery1.SQL.Text);
        exit;
    end;

    Label2.Visible := true;
    Label2.Caption := 'Знайдено варіантів: ' +
IntToStr(DataSource1.DataSet.RecordCount) + ' ';

    (DBGrid1.Columns[0] as TColumn).Width := 50;
    (DBGrid1.Columns[1] as TColumn).Width := 200;
    (DBGrid1.Columns[2] as TColumn).Width := 50;

end;

-----
//-----
procedure TSelDatesForm.SelDateMemoEnter(Sender:
TObject);
begin
    DBGrid1.SetFocus;
end;

-----
//-----
end.

unit ZNorm;

interface

```

```

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils,
    System.Variants, System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
    Vcl.ExtCtrls, Math, TimeP;

```

```
type
```

```
TZakonNormForm = class(TForm)
```

```
Label1: TLabel;
```

```
GroupBox1: TGroupBox;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
PEEdit_count: TEdit;
```

```
PEEdit_math: TEdit;
```

```
PEEdit_average: TEdit;
```

```
GroupBox2: TGroupBox;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
Label6: TLabel;
```

```
SKEdit_count: TEdit;
```

```
SKEdit_math: TEdit;
```

```
SKEdit_average: TEdit;
```

```
GroupBox3: TGroupBox;
```

```
Label7: TLabel;
```

```
Label8: TLabel;
```

```
Label9: TLabel;
```

```
BLEdit_count: TEdit;
```

```
BLEdit_math: TEdit;
```

```
BLEdit_average: TEdit;
```

```
GroupBox4: TGroupBox;
```

```
Label10: TLabel;
```

```
Label11: TLabel;
```

```
Label12: TLabel;
```

```
REdit_count: TEdit;
```

```
REdit_math: TEdit;
```

```
REdit_average: TEdit;
```

```
Label13: TLabel;
```

```
ZakonNormButton: TButton;
```

```
ZakonNormMemo: TMemo;
```

```
BackButton: TButton;
```

```
procedure FormShow(Sender: TObject);
```

```
procedure ZakonNormMemoEnter(Sender: TObject);
```

```
procedure BackButtonClick(Sender: TObject);
```

```
procedure FormClose(Sender: TObject; var Action:
```

```
TCloseAction);
```

```
procedure ZakonNormButtonClick(Sender: TObject);
```

```
procedure PEEdit_countKeyPress(Sender: TObject; var
```

```
Key: Char);
```

```
procedure SKEdit_countKeyPress(Sender: TObject; var
```

```
Key: Char);
```

```
procedure BLEdit_countKeyPress(Sender: TObject; var
```

```
Key: Char);
```

```
procedure REdit_countKeyPress(Sender: TObject; var
```

```
Key: Char);
```

```
procedure PEEdit_countKeyUp(Sender: TObject; var
```

```
Key: Word;
```

```
Shift: TShiftState);
```

```
procedure SKEdit_countKeyUp(Sender: TObject; var
```

```
Key: Word;
```

```
Shift: TShiftState);
```

```
procedure BLEdit_countKeyUp(Sender: TObject; var
```

```
Key: Word;
```

```
Shift: TShiftState);
```

```
procedure REdit_countKeyUp(Sender: TObject; var
```

```
Key: Word;
```

```
Shift: TShiftState);
```

```
procedure PEEdit_mathKeyPress(Sender: TObject; var
```

```
Key: Char);
```

```
procedure PEEdit_averageKeyPress(Sender: TObject;
```

```
var Key: Char);
```

```
procedure SKEdit_mathKeyPress(Sender: TObject; var
```

```
Key: Char);
```

```
procedure BLEdit_mathKeyPress(Sender: TObject; var
```

```
Key: Char);
```

```
procedure REdit_mathKeyPress(Sender: TObject; var
```

```
Key: Char);
```

```
procedure SKEdit_averageKeyPress(Sender: TObject;
```

```
var Key: Char);
```

```

    procedure BLEdit_averageKeyPress(Sender: TObject;
var Key: Char);
    procedure REdit_averageKeyPress(Sender: TObject;
var Key: Char);
    procedure PEEdit_mathKeyUp(Sender: TObject; var
Key: Word;
    Shift: TShiftState);
    procedure SKEdit_mathKeyUp(Sender: TObject; var
Key: Word;
    Shift: TShiftState);
    procedure BLEdit_mathKeyUp(Sender: TObject; var
Key: Word;
    Shift: TShiftState);
    procedure REdit_mathKeyUp(Sender: TObject; var
Key: Word;
    Shift: TShiftState);
    procedure PEEdit_averageKeyUp(Sender: TObject; var
Key: Word;
    Shift: TShiftState);
    procedure SKEdit_averageKeyUp(Sender: TObject; var
Key: Word;
    Shift: TShiftState);
    procedure BLEdit_averageKeyUp(Sender: TObject; var
Key: Word;
    Shift: TShiftState);
    procedure REdit_averageKeyUp(Sender: TObject; var
Key: Word;
    Shift: TShiftState);

private
    { Private declarations }
    function CalcAverage(param : integer) : real;
public
    { Public declarations }
    procedure IntEditKeyPress(Sender: TObject; var Key:
Char);
    procedure IntEditKeyUp(Sender: TObject; var Key:
Word; Shift: TShiftState);
    procedure FloatEditKeyPress(Sender: TObject; var Key:
Char);
    procedure FloatEditKeyUp(Sender: TObject; var Key:
Word; Shift: TShiftState);
    function CheckEdits(frm: TForm) : boolean;
    function SaveParams(scen : integer; t1, t2, t3, t4 : real):
boolean;
end;

var
    ZakonNormForm: TZakonNormForm;

implementation
    uses GenScen, Menu;
    {$R *.dfm}

//-----
TObject);
    procedure TZakonNormForm.BackButtonClick(Sender:
TObject);
    begin
        Hide;
        //Close;
        GenForm.Show;
        GenForm.FormStyle := fsStayOnTop;
    end;

//-----
TObject; var Action: TCloseAction);
    procedure TZakonNormForm.FormClose(Sender:
TObject; var Action: TCloseAction);
    begin
        MenuForm.Show;
        MenuForm.FormStyle := fsStayOnTop;
    end;

//-----
TObject);
    procedure TZakonNormForm.FormShow(Sender:
TObject);
    var hint : String;
    begin
        ZakonNormMemo.Color := RGB (240, 240, 240);
        ZakonNormMemo.Font.Color := RGB (133, 10, 10);
        hint := 'Величина середнього відхилення завжди є
меньшою величиною, ніж математичне очікування';

        PEEdit_average.Hint := hint;
        SKEdit_average.Hint := hint;
        BLEdit_average.Hint := hint;
        REdit_average.Hint := hint;

        PEEdit_average.ShowHint := true;
        SKEdit_average.ShowHint := true;
        BLEdit_average.ShowHint := true;
        REdit_average.ShowHint := true;

    end;

//-----
: real;
    function TZakonNormForm.CalcAverage(param : integer)
: real;
    var
        R1, R2 : real; //случайные числа от 0 до 1
        i : integer; //счётчик
        k : integer; //кол-во экспериментов
        m, s : real; {матем. ожидание и среднее отклонение}
        arr : array of real; {динамический массив для
вычисления среднего значения}
        res : real; //результат
    begin
        case param of
            1: begin {время начала эвакуации}
                k := StrToInt(PEEdit_count.Text);
                m := StrToFloat(PEEdit_math.Text);
                s := StrToFloat(PEEdit_average.Text);
                end;
            2: begin {время существования толпы людей}
                k := StrToInt(SKEdit_count.Text);
                m := StrToFloat(SKEdit_math.Text);
                s := StrToFloat(SKEdit_average.Text);
                end;
            3: begin {время блокировки путей эвакуации}
                k := StrToInt(BLEdit_count.Text);
                m := StrToFloat(BLEdit_math.Text);
                s := StrToFloat(BLEdit_average.Text);
                end;
            4: begin {расчётное время эвакуации}
                k := StrToInt(REdit_count.Text);
                m := StrToFloat(REdit_math.Text);
                s := StrToFloat(REdit_average.Text);
                end;
        end;

        SetLength(arr, k); //выделить память под k элементов

        {заполняю массив значениями по формуле норм.
закона распределения}
        //Memo1.Clear;
        //Memo1.Visible := false;
        for i := 0 to k - 1 do
            begin
                R1 := Random;
                R2 := Random;
                arr[i] := m + s * Sqrt(-2 * Ln(R1)) * Sin(2 * PI * R2);
                {
                Memo1.Lines.Add( FloatToStr(R1) + ';' +
FloatToStr(R2) + ';' +
FloatToStr(Ln(R1)) + ';' + FloatToStr(Sqrt(-2 *
Ln(R1))) + ';' +
FloatToStr(Sin(2 * PI * R2)) + ';' + FloatToStr(arr[i]));
                }
                Application.ProcessMessages;
            end;

        // Memo1.Lines.SaveToFile(IntToStr(param) + '.csv');

        res := 0;

        for i:=0 to high(arr) do
            begin

```

```

res := res + arr[i]; //сложим все элементы

Application.ProcessMessages;
end;

res := res / (high(arr)+1); {разделим на количество,
оно на 1 больше чем старший индекс}

{округление результата 4 знака после запятой}
Result := RoundTo(res, -4);
end;

//-----
function TZakonNormForm.SaveParams(scen : integer;
t1, t2, t3, t4 : real) : boolean;
var update : integer;
res : boolean;
begin
res := true;

MenuForm.ADOQuery1.SQL.Text := 'SELECT count(*)
as n '+
'FROM [Вхідні параметри - часові характеристики] '
+
'WHERE [id_operation] = ' + IntToStr(MenuForm.Tag)
+ ' and [scen_num] = ' + IntToStr(scen) + ';;

try
MenuForm.ADOQuery1.Open;
except
ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
end;

update :=
MenuForm.ADOQuery1.FieldByName('n').AsInteger;

if update = 0 then {запрос на вставку}
MenuForm.ADOQuery1.SQL.Text := 'Insert into
[Вхідні параметри - часові характеристики] ' +
'[id_operation], [start], [crowd], [block], [calc],
[scen_num] Values ' +
'(' + IntToStr(MenuForm.Tag) + ', ' + FloatToStr(t1) +
', ' + FloatToStr(t2) +
', ' + FloatToStr(t3) + ', ' + FloatToStr(t4) + ', ' +
IntToStr(scen) + ');'
else
{...на обновление}
MenuForm.ADOQuery1.SQL.Text := 'Update
[Вхідні параметри - часові характеристики] ' +
'set [start] = ' + FloatToStr(t1) + ', [crowd] = ' +
FloatToStr(t2) +
', [block] = ' + FloatToStr(t3) + ', [calc] = ' +
FloatToStr(t4) +
' Where [id_operation] = ' +
IntToStr(MenuForm.Tag) + ' and [scen_num] = ' + IntToStr(scen) +
';;

try
MenuForm.ADOQuery1.ExecSQL;
except
ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
res := false;
exit;
end;

Result := res;

end;

//-----
procedure
TZakonNormForm.ZakonNormButtonClick(Sender: TObject);
var
t1, t2, t3, t4 : real; // временные ха-ки
i, n : integer; //счётчик, кол-во сценариев
ex : boolean;
begin
if not CheckEdits(Self) then
begin
ShowMessage('Невірно або не повністю заповнені
деякі поля');
exit;
end;

ex := true;

{выбираю сценарий}
MenuForm.ADOQuery1.SQL.Text := 'SELECT
[Кількість сценаріїв] as n '+
'FROM [Вхідні параметри - операції] ' +
'WHERE [Код] = ' + IntToStr(MenuForm.Tag) + ';;

try
MenuForm.ADOQuery1.Open;
n :=
MenuForm.ADOQuery1.FieldByName('n').AsInteger;
except
ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
end;

{
Для каждого сценария необходимо сгенерировать
параметры времени столько раз,
сколько прописано в графе "кількість
експериментів".
и из этого кол-ва находится среднее значение
каждого параметра
}
for i := 1 to n do
begin
t1 := CalcAverage(1);
t2 := CalcAverage(2);
t3 := CalcAverage(3);
t4 := CalcAverage(4);

//вставляю в табл. [Вхідні параметри - часові
характеристики]
if not SaveParams(i, t1, t2, t3, t4) then
begin
ex := false;
break;
end;

Application.ProcessMessages;
end;

if not ex then
exit;

Hide;
//Close;
TimeParamForm.Tag := 0;
TimeParamForm.Show;
TimeParamForm.FormStyle := fsStayOnTop;
end;

//-----
procedure
TZakonNormForm.ZakonNormMemoEnter(Sender: TObject);
begin
ZakonNormButton.SetFocus;
end;

//-----
procedure TZakonNormForm.IntEditKeyPress(Sender:
TObject; var Key: Char);
begin
{ ввод только цифр и backspace }
if not (Key in ['0'..'9', #8]) then
Key:=#0;
end;

```



```

//-----
procedure TZakonNormForm.IntEditKeyUp(Sender:
TObject; var Key: Word; Shift: TShiftState);
begin

    if Trim((Sender as TEdit).Text) = " then
        exit;

        {если первый 0 - удаляю его}
        while (Length((Sender as TEdit).Text) > 1) and ((Sender
as TEdit).Text[1] = '0') do
            (Sender as TEdit).Text := Copy((Sender as TEdit).Text,
2, Length((Sender as TEdit).Text) - 1);

            {максимум 1000}
            if (StrToInt((Sender as TEdit).Text) > 1000) or
(StrToInt((Sender as TEdit).Text) < 1) then
                begin
                    (Sender as TEdit).Font.Color := clRed;
                    (Sender as TEdit).Hint := 'Введіть число від 1 до
1000';
                    (Sender as TEdit).ShowHint := True;
                end
            else
                begin
                    (Sender as TEdit).Font.Color := clWindowText;
                    (Sender as TEdit).ShowHint := False;
                end;

            end;

//-----
procedure
TZakonNormForm.PEEdit_countKeyPress(Sender: TObject; var
Key: Char);
begin
    IntEditKeyPress(Sender, Key);
end;

    procedure
TZakonNormForm.PEEdit_countKeyUp(Sender: TObject; var Key:
Word;
    Shift: TShiftState);
    begin
        IntEditKeyUp(Sender, Key, Shift);
    end;

//-----
procedure
TZakonNormForm.SKEdit_countKeyPress(Sender: TObject; var
Key: Char);
begin
    IntEditKeyPress(Sender, Key);
end;

    procedure
TZakonNormForm.SKEdit_countKeyUp(Sender: TObject; var Key:
Word;
    Shift: TShiftState);
    begin
        IntEditKeyUp(Sender, Key, Shift);
    end;

//-----
procedure
TZakonNormForm.BLEdit_countKeyPress(Sender: TObject; var
Key: Char);
begin
    IntEditKeyPress(Sender, Key);
end;

    procedure
TZakonNormForm.BLEdit_countKeyUp(Sender: TObject; var Key:
Word;
    Shift: TShiftState);
    begin
        IntEditKeyUp(Sender, Key, Shift);
    end;

//-----
procedure
TZakonNormForm.REdit_countKeyPress(Sender: TObject; var Key:
Char);
begin
    IntEditKeyPress(Sender, Key);
end;

    procedure
TZakonNormForm.REdit_countKeyUp(Sender: TObject; var Key:
Word;
    Shift: TShiftState);
    begin
        IntEditKeyUp(Sender, Key, Shift);
    end;

//-----
procedure TZakonNormForm.FloatEditKeyPress(Sender:
TObject; var Key: Char);
begin
    { ввод только цифр, разделителя и backspace }
    if not (Key in ['0'..'9', '#8, '.']) then
        Key:=#0;
end;

//-----
function SymbolEntersCount(s: string; ch: char): integer;
{Кол-во вхождений символа в строку}
var
    i: integer;
begin
    Result := 0;
    if Trim(s) <> " then
        for i := 1 to Length(s) do
            if s[i] = ch then
                inc(Result);
end;

//-----
procedure TZakonNormForm.FloatEditKeyUp(Sender:
TObject; var Key: Word; Shift: TShiftState);
var Ed : TEdit;
min, max : real;
name : String;
begin
    Ed := (Sender as TEdit);
    name := Ed.Name;

    if Trim(Ed.Text) = " then
        exit;

        {разделитель - точка}
        FormatSettings.DecimalSeparator := '.';

        {удаляю разделитель больше 1го раза}
        while (SymbolEntersCount(Ed.Text, '.') > 1) do
            begin
                Ed.Text := Copy(Ed.Text, 1, Length(Ed.Text) - 1);
            end;

        {удаляю первый разделитель}
        if Ed.Text[1] = '.' then
            begin
                if Length(Ed.Text) > 1 then
                    Ed.Text := Copy(Ed.Text, 2, Length(Ed.Text) - 1)
                else
                    begin
                        Ed.Text := "";
                        exit;
                    end;
            end;

        {удаляю первый ноль, если за ним не разделитель}
        while (Length(Ed.Text) > 1) and (Ed.Text[1] = '0') and
(Ed.Text[2] <> '.') do
            begin
                Ed.Text := Copy(Ed.Text, 2, Length(Ed.Text) - 1);
            end;

```

```

if Ed.Text[Length(Ed.Text)] = '.' then
  exit;

  {Матем. ожидание от 0.001 до 50}
if Pos('Edit_math', name) > 0 then
  begin
    min := 0.001;
    max := 50;
  end
else
  {Среднее отклонение - любое положительное
число}
  if Pos('Edit_average', name) > 0 then
    begin
      min := 0.001;
      max := 50;
    end
  else
    {ограничения для равномерного закона от 0 до 15
(включая числа с запятой)}
    if (Pos('Edit_low', name) > 0) or (Pos('Edit_high', name)
> 0) then
      begin
        min := 0;
        max := 15;
      end
    else
      exit;

      {минимум min максимум max}
      if (StrToFloat(Ed.Text) > max) or (StrToFloat(Ed.Text)
< min) then
        begin
          Ed.Font.Color := clRed;
          Ed.Hint := 'Введіть число від ' + FloatToStr(min) + '
до ' + FloatToStr(max);
          Ed.ShowHint := True;
        end
        else
          begin
            Ed.Font.Color := clWindowText;
            Ed.ShowHint := False;
          end;

        end;

        //-----
        procedure
TZakonNormForm.PEEdit_mathKeyPress(Sender: TObject; var
Key: Char);
        begin
          FloatEditKeyPress(Sender, Key);
        end;

        procedure
TZakonNormForm.PEEdit_mathKeyUp(Sender: TObject; var Key:
Word;
          Shift: TShiftState);
        begin
          FloatEditKeyUp(Sender, Key, Shift);
        end;

        //-----
        procedure
TZakonNormForm.SKEdit_mathKeyPress(Sender: TObject; var
Key: Char);
        begin
          FloatEditKeyPress(Sender, Key);
        end;

        procedure
TZakonNormForm.SKEdit_mathKeyUp(Sender: TObject; var Key:
Word;
          Shift: TShiftState);
        begin
          FloatEditKeyUp(Sender, Key, Shift);
        end;

        //-----
        procedure
TZakonNormForm.BLEdit_mathKeyPress(Sender: TObject; var
Key: Char);
        begin
          FloatEditKeyPress(Sender, Key);
        end;

        procedure
TZakonNormForm.BLEdit_mathKeyUp(Sender: TObject; var Key:
Word;
          Shift: TShiftState);
        begin
          FloatEditKeyUp(Sender, Key, Shift);
        end;

        //-----
        procedure
TZakonNormForm.REdit_mathKeyPress(Sender: TObject; var Key:
Char);
        begin
          FloatEditKeyPress(Sender, Key);
        end;

        procedure
TZakonNormForm.REdit_mathKeyUp(Sender:
TObject; var Key: Word;
          Shift: TShiftState);
        begin
          FloatEditKeyUp(Sender, Key, Shift);
        end;

        //-----
        procedure
TZakonNormForm.PEEdit_averageKeyPress(Sender: TObject; var
Key: Char);
        begin
          FloatEditKeyPress(Sender, Key);
        end;

        procedure
TZakonNormForm.PEEdit_averageKeyUp(Sender: TObject; var
Key: Word;
          Shift: TShiftState);
        begin
          FloatEditKeyUp(Sender, Key, Shift);
        end;

        //-----
        procedure
TZakonNormForm.SKEdit_averageKeyPress(Sender: TObject; var
Key: Char);
        begin
          FloatEditKeyPress(Sender, Key);
        end;

        procedure
TZakonNormForm.SKEdit_averageKeyUp(Sender: TObject; var
Key: Word;
          Shift: TShiftState);
        begin
          FloatEditKeyUp(Sender, Key, Shift);
        end;

        //-----
        procedure
TZakonNormForm.BLEdit_averageKeyPress(Sender: TObject; var
Key: Char);
        begin
          FloatEditKeyPress(Sender, Key);
        end;

        procedure
TZakonNormForm.BLEdit_averageKeyUp(Sender: TObject; var
Key: Word;
          Shift: TShiftState);
        begin
          FloatEditKeyUp(Sender, Key, Shift);
        end;

```

```

    FloatEditKeyUp(Sender, Key, Shift);
end;

//-----
procedure
TZakonNormForm.REdit_averageKeyPress(Sender: TObject; var
Key: Char);
begin
    FloatEditKeyPress(Sender, Key);
end;

procedure
TZakonNormForm.REdit_averageKeyUp(Sender: TObject; var Key:
Word;
    Shift: TShiftState);
begin
    FloatEditKeyUp(Sender, Key, Shift);
end;

//-----
function TZakonNormForm.CheckEdits(frm: TForm) :
boolean;
var
    res : boolean;
    i : integer;
    Com : TComponent;
begin
    res := true;

    for i := 0 to frm.ComponentCount - 1 do
    begin
        Com := frm.Components[i];

        if Com.ClassName = 'TEdit' then
        begin
            if (Trim((Com as TEdit).Text) = "") or
                ((Com as TEdit).Font.Color = clRed)
            then
            begin
                res := false;
                break;
            end;
        end;
    end;

    Result := res;
end;

//-----
end.

unit ZRavn;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils,
    System.Variants, System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, ZNorm,
    Vcl.StdCtrls, Math, TimeP;

type
    TZakonRavnForm = class(TForm)
        BackButton: TButton;
        GroupBox1: TGroupBox;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        PEEdit_count: TEdit;
        PEEdit_low: TEdit;
        PEEdit_high: TEdit;
        GroupBox2: TGroupBox;
        Label4: TLabel;
        SKEdit_count: TEdit;
        SKEdit_low: TEdit;
        SKEdit_high: TEdit;
        GroupBox3: TGroupBox;
        Label7: TLabel;
        BLEdit_count: TEdit;
        BLEdit_low: TEdit;
        BLEdit_high: TEdit;
        GroupBox4: TGroupBox;
        Label10: TLabel;
        REdit_count: TEdit;
        REdit_low: TEdit;
        REdit_high: TEdit;
        Label13: TLabel;
        ZakonRavnButton: TButton;
        ZakonRavnMemo: TMemo;
        Label5: TLabel;
        Label6: TLabel;
        Label8: TLabel;
        Label9: TLabel;
        Label11: TLabel;
        Label12: TLabel;
        procedure BackButtonClick(Sender: TObject);
        procedure FormClose(Sender: TObject; var Action:
        TCloseAction);
        procedure FormShow(Sender: TObject);
        procedure ZakonRavnButtonClick(Sender: TObject);
        procedure ZakonRavnMemoEnter(Sender: TObject);
        procedure PEEdit_countKeyPress(Sender: TObject; var
        Key: Char);
        procedure PEEdit_countKeyUp(Sender: TObject; var
        Key: Word;
            Shift: TShiftState);
        procedure SKEdit_countKeyPress(Sender: TObject; var
        Key: Char);
        procedure SKEdit_countKeyUp(Sender: TObject; var
        Key: Word;
            Shift: TShiftState);
        procedure BLEdit_countKeyPress(Sender: TObject; var
        Key: Char);
        procedure BLEdit_countKeyUp(Sender: TObject; var
        Key: Word;
            Shift: TShiftState);
        procedure REdit_countKeyPress(Sender: TObject; var
        Key: Char);
        procedure REdit_countKeyUp(Sender: TObject; var
        Key: Word;
            Shift: TShiftState);
        procedure PEEdit_lowKeyPress(Sender: TObject; var
        Key: Char);
        procedure PEEdit_lowKeyUp(Sender: TObject; var
        Key: Word; Shift: TShiftState);
        procedure PEEdit_highKeyUp(Sender: TObject; var
        Key: Word; Shift: TShiftState);
        procedure PEEdit_highKeyPress(Sender: TObject; var
        Key: Char);
        procedure SKEdit_lowKeyPress(Sender: TObject; var
        Key: Char);
        procedure SKEdit_lowKeyUp(Sender: TObject; var
        Key: Word; Shift: TShiftState);
        procedure SKEdit_highKeyPress(Sender: TObject; var
        Key: Char);
        procedure SKEdit_highKeyUp(Sender: TObject; var
        Key: Word; Shift: TShiftState);
        procedure BLEdit_lowKeyUp(Sender: TObject; var
        Key: Word; Shift: TShiftState);
        procedure BLEdit_lowKeyPress(Sender: TObject; var
        Key: Char);
        procedure BLEdit_highKeyPress(Sender: TObject; var
        Key: Char);
        procedure BLEdit_highKeyUp(Sender: TObject; var
        Key: Word; Shift: TShiftState);
        procedure REdit_lowKeyPress(Sender: TObject; var
        Key: Char);
        procedure REdit_lowKeyUp(Sender: TObject; var Key:
        Word; Shift: TShiftState);
        procedure REdit_highKeyPress(Sender: TObject; var
        Key: Char);
        procedure REdit_highKeyUp(Sender: TObject; var
        Key: Word; Shift: TShiftState);
        private
            { Private declarations }
            function CalcAverage(param : integer) : real;
    end;

```

```

public
  { Public declarations }
end;

var
  ZakonRavnForm: TZakonRavnForm;

implementation
uses GenScen, Menu;

{$R *.dfm}

-----

procedure TZakonRavnForm.BackButtonClick(Sender:
TObject);
begin
  Hide;
  //Close;
  GenForm.Show;
  GenForm.FormStyle := fsStayOnTop;
end;

-----

procedure TZakonRavnForm.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
  MenuForm.Show;
  MenuForm.FormStyle := fsStayOnTop;
end;

-----

procedure TZakonRavnForm.FormShow(Sender:
TObject);
begin
  ZakonRavnMemo.Color := RGB (240, 240, 240);
  ZakonRavnMemo.Font.Color := RGB (133, 10, 10);
end;

-----

function TZakonRavnForm.CalcAverage(param : integer)
: real;
var
  R : real; //случайное число от 0 до 1
  i : integer; //счётчик
  k : integer; //кол-во экспериментов
  a, b : real; {нижняя и верхняя граница числовьг ха-к}
  arr : array of real; {динамический массив для
вычисления среднего значения}
  res : real; //результат
begin
  case param of
    1: begin {время начала эвакуации}
        k := StrToInt(PEEdit_count.Text);
        a := StrToFloat(PEEdit_low.Text);
        b := StrToFloat(PEEdit_high.Text);
        end;
    2: begin {время существования толпы людей}
        k := StrToInt(SKEdit_count.Text);
        a := StrToFloat(SKEdit_low.Text);
        b := StrToFloat(SKEdit_high.Text);
        end;
    3: begin {время блокировки путей эвакуации}
        k := StrToInt(BLEdit_count.Text);
        a := StrToFloat(BLEdit_low.Text);
        b := StrToFloat(BLEdit_high.Text);
        end;
    4: begin {расчётное время эвакуации}
        k := StrToInt(REdit_count.Text);
        a := StrToFloat(REdit_low.Text);
        b := StrToFloat(REdit_high.Text);
        end;
  end;
  SetLength(arr, k); //выделить память под k элементов

```

```

{заполняю массив значениями по формуле норм.
закона распределения}
//Memo1.Clear;
//Memo1.Visible := false;
for i := 0 to k - 1 do
begin
  R := Random;

  arr[i] := R * (b - a) + a;
  {
  Memo1.Lines.Add( FloatToStr(R) + ';' + FloatToStr(a)
+ ';' +
  FloatToStr(b) + ';' + FloatToStr(SR * (b - a) + a));
  }
  Application.ProcessMessages;
end;

// Memo1.Lines.SaveToFile(IntToStr(param) + '.csv');

res := 0;

for i:=0 to high(arr) do
begin
  res := res +arr[i]; //сложим все элементы

  Application.ProcessMessages;
end;

res := res / (high(arr)+1); {разделим на количество,
оно на 1 больше чем старший индекс}

{округление результата 4 знака после запятой}
Result := RoundTo(res, -4);
end;

-----

procedure
TZakonRavnForm.ZakonRavnButtonClick(Sender: TObject);
var
  t1, t2, t3, t4 : real; // временные ха-ки
  i, n : integer; //счётчик, кол-во сценариев
  ex : boolean;
begin

  if not ZakonNormForm.CheckEdits(Self) then
  begin
    ShowMessage('Невірно або не повністю заповнені
деякі поля');
    exit;
  end;

  ex := true;

  {выбираю сценарий}
  MenuForm.ADOQuery1.SQL.Text := 'SELECT
[Кількість сценаріїв] as n '+
  'FROM [Вхідні параметри - операції] '+
  'WHERE [Код] = ' + IntToStr(MenuForm.Tag) + ' ';

  try
    MenuForm.ADOQuery1.Open;
    n :=
MenuForm.ADOQuery1.FieldByName('n').AsInteger;
  except
    ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
  end;
end;

{
Для каждого сценария необходимо сгенерировать
параметры времени столько раз,
сколько прописано в графе "кількість
експериментів".
}

```

```

        и из этого кол-ва находится среднее значение
каждого параметра
    }
    for i := 1 to n do
    begin
        t1 := CalcAverage(1);
        t2 := CalcAverage(2);
        t3 := CalcAverage(3);
        t4 := CalcAverage(4);

        //вставляю в табл. [Вхідні параметри - часові
характеристики]
        if not ZakonNormForm.SaveParams(i, t1, t2, t3, t4)
then
        begin
            ex := false;
            break;
        end;

        Application.ProcessMessages;
    end;

    if not ex then
        exit;

    Hide;
    // Close;
    TimeParamForm.Tag := 1;
    TimeParamForm.Show;
    TimeParamForm.FormStyle := fsStayOnTop;
end;

//-----
procedure
TZakonRavnForm.ZakonRavnMemoEnter(Sender: TObject);
begin
    ZakonRavnButton.SetFocus;
end;

//-----
procedure
TZakonRavnForm.PEEdit_lowKeyPress(Sender: TObject; var Key:
Char);
begin
    ZakonNormForm.FloatEditKeyPress(Sender, Key);
end;

    procedure TZakonRavnForm.PEEdit_lowKeyUp(Sender:
TObject; var Key: Word;
        Shift: TShiftState);
    begin
        ZakonNormForm.FloatEditKeyUp(Sender, Key, Shift);
    end;

//-----
procedure
TZakonRavnForm.PEEdit_highKeyPress(Sender: TObject; var Key:
Char);
begin
    ZakonNormForm.FloatEditKeyPress(Sender, Key);
end;

    procedure TZakonRavnForm.PEEdit_highKeyUp(Sender:
TObject; var Key: Word;
        Shift: TShiftState);
    begin
        ZakonNormForm.FloatEditKeyUp(Sender, Key, Shift);
    end;

//-----
procedure
TZakonRavnForm.PEEdit_countKeyPress(Sender: TObject; var
Key: Char);
begin
    ZakonNormForm.IntEditKeyPress(Sender, Key);
end;

        procedure
TZakonRavnForm.PEEdit_countKeyUp(Sender: TObject; var Key:
Word;
            Shift: TShiftState);
        begin
            ZakonNormForm.IntEditKeyUp(Sender, Key, Shift);
        end;

//-----
procedure
TZakonRavnForm.BLEdit_lowKeyPress(Sender: TObject; var Key:
Char);
begin
    ZakonNormForm.FloatEditKeyPress(Sender, Key);
end;

    procedure TZakonRavnForm.BLEdit_lowKeyUp(Sender:
TObject; var Key: Word;
        Shift: TShiftState);
    begin
        ZakonNormForm.FloatEditKeyUp(Sender, Key, Shift);
    end;

//-----
procedure
TZakonRavnForm.BLEdit_highKeyPress(Sender: TObject; var Key:
Char);
begin
    ZakonNormForm.FloatEditKeyPress(Sender, Key);
end;

    procedure TZakonRavnForm.BLEdit_highKeyUp(Sender:
TObject; var Key: Word;
        Shift: TShiftState);
    begin
        ZakonNormForm.FloatEditKeyUp(Sender, Key, Shift);
    end;

```

```

    Shift: TShiftState);
begin
    ZakonNormForm.FloatEditKeyUp(Sender, Key, Shift);
end;

//-----
procedure
TZakonRavnForm.BLEdit_countKeyPress(Sender: TObject; var
Key: Char);
begin
    ZakonNormForm.IntEditKeyPress(Sender, Key);
end;

procedure
TZakonRavnForm.BLEdit_countKeyUp(Sender: TObject; var Key:
Word;
    Shift: TShiftState);
begin
    ZakonNormForm.IntEditKeyUp(Sender, Key, Shift);
end;

//-----
procedure TZakonRavnForm.REdit_lowKeyPress(Sender:
TObject; var Key: Char);
begin
    ZakonNormForm.FloatEditKeyPress(Sender, Key);
end;

procedure TZakonRavnForm.REdit_lowKeyUp(Sender:
TObject; var Key: Word;
    Shift: TShiftState);
begin
    ZakonNormForm.FloatEditKeyUp(Sender, Key, Shift);
end;

//-----
procedure
TZakonRavnForm.REdit_highKeyPress(Sender: TObject; var Key:
Char);
begin
    ZakonNormForm.FloatEditKeyPress(Sender, Key);
end;

procedure TZakonRavnForm.REdit_highKeyUp(Sender:
TObject; var Key: Word;
    Shift: TShiftState);
begin
    ZakonNormForm.FloatEditKeyUp(Sender, Key, Shift);
end;

//-----
procedure
TZakonRavnForm.REdit_countKeyPress(Sender: TObject; var Key:
Char);
begin
    ZakonNormForm.IntEditKeyPress(Sender, Key);
end;

procedure TZakonRavnForm.REdit_countKeyUp(Sender:
TObject; var Key: Word;
    Shift: TShiftState);
begin
    ZakonNormForm.IntEditKeyUp(Sender, Key, Shift);
end;

//-----
end.

unit ChartUnit;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils,
    System.Variants, System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
    VclTee.TeeGDIPlus, VCLTee.TeEngine,
    VCLTee.Series, Vcl.ExtCtrls, VCLTee.TeeProc,
    VCLTee.Chart, Data.DB,
    Data.Win.ADODB;

type
    TChartForm = class(TForm)
        ADOQuery1: TADOQuery;
        DataSource1: TDataSource;
        Chart1: TChart;
        Series1: TBarSeries;
        procedure FormClose(Sender: TObject; var Action:
TCloseAction);
        procedure FormShow(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    ChartForm: TChartForm;

implementation
uses TimeP, Menu;
{$R *.dfm}

//-----
procedure TChartForm.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
    Hide;
    TimeParamForm.Show;
    TimeParamForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TChartForm.FormShow(Sender: TObject);
var x, y: real;
    field: String;
    clr: TColor;
begin
    case Self.Tag of
    1:
        begin
            Self.Caption := 'Графік тпе [Оцінка пожежного
ризикy]';
            field := 'start';
            clr := RGB(68, 102, 163);
        end;
    2:
        begin
            Self.Caption := 'Графік тск [Оцінка пожежного
ризикy]';
            field := 'crowd';
            clr := RGB(141, 81, 171);
        end;
    3:
        begin
            Self.Caption := 'Графік тбл [Оцінка пожежного
ризикy]';
            field := 'block';
            clr := RGB(205, 76, 82);
        end;
    4:
        begin
            Self.Caption := 'Графік тр [Оцінка пожежного
ризикy]';
            field := 'calc';
        end;
    end;

    ADOQuery1.Connection :=
MenuForm.ADOConnection1;
    DataSource1.DataSet := ADOQuery1;
    ADOQuery1.SQL.Text := 'select Top 100 ' + field + ' as
zzz from [Вхідні параметри - часові характеристики] ' +
'where [id_operation] = ' + IntToStr(MenuForm.Tag) +
' order by [Код]';

```

```

try
  ADOQuery1.Open;
except
  ShowMessage('Помилка в запиті: ' +
ADOQuery1.SQL.Text);
exit;
end;

DataSource1.DataSet.First;
Series1.Clear;
x := 0.5;

while not DataSource1.DataSet.Eof do
begin
  y := y + 0.5;
DataSource1.DataSet.FieldByName('zzz').AsFloat;
  Series1.AddXY(x, y, ", clr);

  x := x + 0.5;
  DataSource1.DataSet.Next;
end;

end;

//-----
end.

unit IPRCalc;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
  System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
  VclTee.TeeGDIPPlus,
  Data.DB, Vcl.Grids, Vcl.DBGrids, VCLTee.TeEngine,
  VCLTee.TeeProcs,
  VCLTee.Chart, Vcl.ExtCtrls, Data.Win.ADODB,
  VCLTee.Series, SelChart, Math, IPR_ReCalc;

type
  TIPRCalcForm = class(TForm)
    Panel2: TPanel;
    DBGrid1: TDBGrid;
    Memo1: TMemo;
    Panel3: TPanel;
    Memo2: TMemo;
    dynamicButton: TButton;
    ADOQuery1: TADOQuery;
    DSGrid: TDataSource;
    Splitter1: TSplitter;
    Panel1: TPanel;
    Splitter3: TSplitter;
    Splitter2: TSplitter;
    Chart1: TChart;
    Chart2: TChart;
    Chart3: TChart;
    Series1: TBarSeries;
    Series2: TBarSeries;
    Series3: TBarSeries;
  procedure FormClose(Sender: TObject; var Action:
  TCloseAction);
  procedure FormShow(Sender: TObject);
  procedure Memo1Enter(Sender: TObject);
  procedure Memo2Enter(Sender: TObject);
  procedure dynamicButtonClick(Sender: TObject);
  procedure FormResize(Sender: TObject);

  private
    { Private declarations }
    function Calc : boolean;
    function InitGrid : boolean;
    function Compare : boolean;
  public
    { Public declarations }
  end;

end;

var
  IPRCalcForm: TIPRCalcForm;

implementation
  uses Menu;
  {$R *.dfm}

//-----
procedure TIPRCalcForm.dynamicButtonClick(Sender:
TObject);
begin
  Hide;
  SelDates4ChartForm.Show;
  SelDates4ChartForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TIPRCalcForm.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
  MenuForm.Show;
  MenuForm.FormStyle := fsStayOnTop;
end;

//-----
procedure TIPRCalcForm.FormResize(Sender: TObject);
begin
  Chart1.Height := Round(Self.Height / 3);
  Chart1.Top := 0;

  Chart2.Height := Round(Self.Height / 3);
  Chart2.Top := Chart1.Top + Chart1.Height +
Splitter2.Height;
  Splitter2.Top := Chart1.Height;

  Chart3.Height := Round(Self.Height / 3);
  Chart3.Top := Chart2.Top + Chart2.Height +
Splitter3.Height;
  Splitter3.Top := Chart2.Top + Chart2.Height;
end;

//-----
procedure TIPRCalcForm.FormShow(Sender: TObject);
var q, k, qk, qq, kk, y : real;
begin
  FormatSettings.DecimalSeparator := ',';

  if not Calc then
    exit;

  if not InitGrid then
    exit;

  if not Compare then
    exit;

  {построение гистограмм}
  DSGrid.DataSet.First;
  Series1.Clear;
  Series2.Clear;
  Series3.Clear;

  q := 0.5;
  k := 0.5;
  qk := 0.5;

  while not DSGrid.DataSet.Eof do
    begin
      qq := DSGrid.DataSet.FieldByName('q').AsFloat *
1000;
      Series1.AddXY(q, qq, "");
      kk := DSGrid.DataSet.FieldByName('k').AsFloat;
      Series2.AddXY(k, kk, "");
      y := DSGrid.DataSet.FieldByName('qk').AsFloat *
1000;
    end;
  end;
end;

```

```

Series3.AddXY(qk, y, "");

q := q + 0.5;
k := k + 0.5;
qk := qk + 0.5;
DSGrid.DataSet.Next;
end;
end;

//-----
procedure TIPRCalcForm.Memo1Enter(Sender: TObject);
begin
  dynamicButton.SetFocus;
end;

//-----
procedure TIPRCalcForm.Memo2Enter(Sender: TObject);
begin
  dynamicButton.SetFocus;
end;

//-----
function TIPRCalcForm.Calc : boolean;
var
  n : integer;
  Qpi, Kap, Pp: real;
begin
  {
  расчёт:
  - Оценка пожарного риска, Qв.
  Qv = Qpi * (1 - Kap) * Pp * (1 - Pe) * (1 - Kpz)

  - Коэффициент интенсивности развития
  пожара, Кр.
  Kp = 1/ (R1+R2 + ... + Rn) / 5

  - Произведение Qв*Кр.
  }

  MenuForm.ADOQuery1.SQL.Text := 'select count(*) as
n from [Оцінка пожежного ризику]' +
  'Where [id_operation] = ' + IntToStr(MenuForm.Tag);

  try
    MenuForm.ADOQuery1.Open;
    n :=
MenuForm.ADOQuery1.FieldByName('n').AsInteger;
  except
    ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
    Result := false;
    exit;
  end;

  MenuForm.ADOQuery1.SQL.Text := 'select Top 1
[d].[Qpi] as q, [d].[Kan] as k, [d].[Pnp] as p from [Дані] as d';

  try
    MenuForm.ADOQuery1.Open;
    Qpi :=
MenuForm.ADOQuery1.FieldByName('q').AsFloat;
    Kap :=
MenuForm.ADOQuery1.FieldByName('k').AsFloat;
    Pp :=
MenuForm.ADOQuery1.FieldByName('p').AsFloat;
  except
    ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
    Result := false;
    exit;
  end;

  {если есть строки в таблице - update}
  if (n > 0) then
    MenuForm.ADOQuery1.SQL.Text := 'UPDATE
[Оцінка пожежного ризику] as [opr]' +
    'INNER JOIN (' +

```

```

'SELECT ('+ FloatToStr(Qpi) + ' * (1 - '+
FloatToStr(Kap) + ') * '+ FloatToStr(Pp) +
' * (1 - vh.[Pe]) * (1 - vh.[Kpz]) ) as [Q], '+
' ( 1 / (([place].[Експертна оцінка, бал] + '+
[ex].[Експертна оцінка, бал] + [sr].[Експертна
оцінка, бал] + '+
[par].[Експертна оцінка, бал] + [spar].[Експертная
оцінка, бал] ) / 5)) as [K], '+
'((+ FloatToStr(Qpi) + ' * (1 - '+ FloatToStr(Kap) + ')
* '+ FloatToStr(Pp) +
' * (1 - vh.[Pe]) * (1 - vh.[Kpz]) ) * '+
' ( 1 / (([place].[Експертна оцінка, бал] + '+
[ex].[Експертна оцінка, бал] + [sr].[Експертна
оцінка, бал] + '+
[par].[Експертна оцінка, бал] + [spar].[Експертная
оцінка, бал] ) / 5))) as [QK], '+
[vh].[id_operation], [vh].[scen_num]' +
'FROM (((([Вхідні параметри - часові
характеристики] AS vh' +
'LEFT JOIN [Вхідні параметри - умови розвитку
пожежі] AS vu ON ' +
'([vu].[scen_num] = [vh].[scen_num]) AND
([vu].[id_operation] = [vh].[id_operation]))' +
'LEFT JOIN [Місця виникнення пожежі] AS place
ON place.[№] = vu.[id_place])' +
'LEFT JOIN [Закономірності розповсюдження
пожежі] AS ex ON ex.[№] = vu.[id_expansion])' +
'LEFT JOIN [Початкова область пожежі] AS sr ON
sr.[№] = vu.[id_start_region])' +
'LEFT JOIN [Параметри навколишнього
середовища] AS par ON par.[№] = vu.[id_params])' +
'LEFT JOIN [Початкові параметри приміщення] AS
spar ON spar.[№] = vu.[id_start_params])' +
'WHERE [vh].[id_operation] = ' +
IntToStr(MenuForm.Tag) + ' ORDER BY [vh].[Код]' +

') as [tab] ON [opr].[id_operation] =
[tab].[id_operation]' +
'SET [Qв] = [tab].[Q], [Кр] = [tab].[K], [Qв*Кр] =
[tab].[QK]' +
'Where [opr].[scen_num] = [tab].[scen_num];'
else
  {insert}
  MenuForm.ADOQuery1.SQL.Text := 'Insert Into
[Оцінка пожежного ризику]' +
'([Qв], [Кр], [Qв*Кр], [id_operation], [scen_num])
select * from (' +

'SELECT ('+ FloatToStr(Qpi) + ' * (1 - '+
FloatToStr(Kap) + ') * '+ FloatToStr(Pp) +
' * (1 - vh.[Pe]) * (1 - vh.[Kpz]) ) as [Qв], '+
' ( 1 / (([place].[Експертна оцінка, бал] + '+
[ex].[Експертна оцінка, бал] + [sr].[Експертна
оцінка, бал] + '+
[par].[Експертна оцінка, бал] + [spar].[Експертная
оцінка, бал] ) / 5)) as [Кр], '+
'((+ FloatToStr(Qpi) + ' * (1 - '+ FloatToStr(Kap) + ')
* '+ FloatToStr(Pp) +
' * (1 - vh.[Pe]) * (1 - vh.[Kpz]) ) * '+
' ( 1 / (([place].[Експертна оцінка, бал] + '+
[ex].[Експертна оцінка, бал] + [sr].[Експертна
оцінка, бал] + '+
[par].[Експертна оцінка, бал] + [spar].[Експертная
оцінка, бал] ) / 5))) as [Qв*Кр], '+
[vh].[id_operation], [vh].[scen_num]' +
'FROM (((([Вхідні параметри - часові
характеристики] AS vh' +
'LEFT JOIN [Вхідні параметри - умови розвитку
пожежі] AS vu ON ' +
'([vu].[scen_num] = [vh].[scen_num]) AND
([vu].[id_operation] = [vh].[id_operation]))' +
'LEFT JOIN [Місця виникнення пожежі] AS place
ON place.[№] = vu.[id_place])' +
'LEFT JOIN [Закономірності розповсюдження
пожежі] AS ex ON ex.[№] = vu.[id_expansion])' +
'LEFT JOIN [Початкова область пожежі] AS sr ON
sr.[№] = vu.[id_start_region])' +
'LEFT JOIN [Параметри навколишнього
середовища] AS par ON par.[№] = vu.[id_params])' +

```



```

LEFT JOIN [Початкові параметри приміщення] AS
spar ON spar.[№] = vu.[id_start_params]) ' +
WHERE [vh].[id_operation] = ' +
IntToStr(MenuForm.Tag) + ' ORDER BY [vh].[Код]' +
) as tab;

try
MenuForm.ADOQuery1.ExecSQL;
except
ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
Result := false;
exit;
end;

Result := true;
end;

//-----
function TIPRCalcForm.InitGrid : boolean;
begin
ADOQuery1.CursorLocation := clUseServer;
ADOQuery1.Connection :=
MenuForm.ADOConnection1;

DBGrid1.DataSource := DSGrid;

ADOQuery1.SQL.Text := 'Select [scen_num], [Qв] as q,
[Kp] as k, [Qв*Kp] as qk, [dt]' +
' from [Оцінка пожежного ризику]' +
'Where [id_operation] = ' + IntToStr(MenuForm.Tag) + '
Order by [scen_num]';

try
ADOQuery1.Open;
except
ShowMessage('Помилка в запиті: ' +
ADOQuery1.SQL.Text);
Result := false;
exit;
end

DSGrid.DataSet := ADOQuery1;

TBCDField(DSGrid.DataSet.FieldByName('q')).DisplayFormat :=
'0.0000000';

TBCDField(DSGrid.DataSet.FieldByName('k')).DisplayFormat :=
'0.0000000';

TBCDField(DSGrid.DataSet.FieldByName('qk')).DisplayFormat :=
'0.0000000';

//TBCDField(DSGrid.DataSet.FieldByName('dt')).DisplayFormat :=
'dd.mm.yyyy HH:mm:ss';

DBGrid1.Columns[0].Title.Caption := 'Сценарій';
DBGrid1.Columns[1].Title.Caption := 'Qв';
DBGrid1.Columns[2].Title.Caption := 'Kp';
DBGrid1.Columns[3].Title.Caption := 'Qв*Kp';
DBGrid1.Columns[4].Title.Caption := 'Дата та час';

(DBGrid1.Columns[0] as TColumn).Width := 50;
(DBGrid1.Columns[1] as TColumn).Width := 70;
(DBGrid1.Columns[2] as TColumn).Width := 70;
(DBGrid1.Columns[3] as TColumn).Width := 70;
(DBGrid1.Columns[4] as TColumn).Width := 100;

Result := true;
end;

//-----
function TIPRCalcForm.Compare : boolean;
var Qnorm, max1, max2 : real;
m1, m2 : String;
begin
MenuForm.ADOQuery1.SQL.Text := 'Select [Qнормат]
as q from [Дані]';
try
MenuForm.ADOQuery1.Open;
except
ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
Result := false;
exit;
end;

DSGrid.DataSet.First;
max1 := DSGrid.DataSet.FieldByName('q').AsFloat;
max2 := DSGrid.DataSet.FieldByName('qk').AsFloat;

while not DSGrid.DataSet.Eof do
begin
if DSGrid.DataSet.FieldByName('q').AsFloat > max1
then
max1 := DSGrid.DataSet.FieldByName('q').AsFloat;
if DSGrid.DataSet.FieldByName('qk').AsFloat > max2
then
max2 := DSGrid.DataSet.FieldByName('qk').AsFloat;

DSGrid.DataSet.Next;
end;

m1 := FloatToStr((RoundTo(max1 * 10000, -3))) +
'*10^-4';
m2 := FloatToStr((RoundTo(max2 * 10000, -3))) +
'*10^-4';

if max1 <= Qnorm then
begin
MenuForm.N8.Enabled := false;
Memo1.Font.Color := clGreen;
Memo1.Text := 'Розрахована величина
індивідуального пожежного ризику Qв = ' + m1 + ' не
перевищує нормативне значення.' +
'Отже реалізована на підприємстві система
протипожежної безпеки є ефективною на поточний момент.' +
'Однак, враховуючи вплив внутрішнього та
зовнішнього середовища (фізичний, моральний знос
обладнання, ' +
'зміна законодавства та т.ін.), необхідно наспаного з
встановленню періодичністю здійснювати оцінку пожежного
ризикy і надалі';
end
else
begin
MenuForm.N8.Enabled := true;
IPR_ReCalcForm.Qv_Edit.Text := FloatToStr(max1);
IPR_ReCalcForm.Qnorm_Edit.Text :=
FloatToStr(Qnorm);

Memo1.Font.Color := clRed;
Memo1.Text := 'Розрахована величина
індивідуального пожежного ризику Qв = ' + m1 + ' перевищує
нормативне значення.' +
'В цьому випадку необхідно розробити комплекс
заходів щодо його зниження з урахуванням фінансових
обмежень організації.' +
'Для переоцінки ризику можна повернутися на
початкову форму та обрати в меню «Обчислити» пункт ' +
«Перерахунок пожежного ризику з урахуванням
заходів»';
end;

if max2 <= Qnorm then
begin
MenuForm.N8.Enabled := false;
Memo2.Font.Color := clGreen;
Memo2.Text := 'Розрахована величина
індивідуального пожежного ризику Qв = ' + m2 + ' не
перевищує нормативне значення.' +
'Отже реалізована на підприємстві система
протипожежної безпеки є ефективною на поточний момент.' +

```

```

'Однак, враховуючи вплив внутрішнього та
зовнішнього середовища (фізичний, моральний знос
обладнання,' +
'зміна законодавства та т.ін.), необхідно наспаного з
встановленню періодичністю здійснювати оцінку пожежного
ризикy і надалі';
end
else
begin
MenuForm.N8.Enabled := true;
IPR_ReCalcForm.Qv_Edit.Text := FloatToStr(max1);
IPR_ReCalcForm.Qnorm_Edit.Text :=
FloatToStr(Qnorm);

Memo2.Font.Color := clRed;
Memo2.Text := 'Розрахована величина
індивідуального пожежного ризику Qv = ' + m2 + ' перевищує
нормативне значення.' +
'В цьому випадку необхідно розробити комплекс
заходів щодо його зниження з урахуванням фінансових
обмежень організації' +
'Для переоцінки ризику можна повернутися на
початкову форму та обрати в меню «Обчислити» пункт' +
'«Перерахунок пожежного ризику з урахуванням
заходів»';
end;

Memo1.Color := RGB(240, 240, 240);
Memo2.Color := RGB(240, 240, 240);
Result := true;
end;
//-----
end.

unit IPR_ReCalc;

interface

uses
Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB,
Vcl.StdCtrls, Vcl.Grids, Math,
Vcl.DBGrids, Data.Win.ADODB, Laba;

type
TIPR_ReCalcForm = class(TForm)
ReCalcMemo: TMemo;
DBGrid1: TDBGrid;
LoadButton: TButton;
ReCalcButton: TButton;
Memo1: TMemo;
ADOQuery1: TADOQuery;
DataSource1: TDataSource;
Qv_Edit: TEdit;
Qnorm_Edit: TEdit;
ZEdit: TEdit;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
procedure FormClose(Sender: TObject; var Action:
TCloseAction);
procedure FormShow(Sender: TObject);
procedure ReCalcMemoEnter(Sender: TObject);
procedure Memo1Enter(Sender: TObject);
procedure LoadButtonClick(Sender: TObject);
procedure ReCalcButtonClick(Sender: TObject);

private
{ Private declarations }
public
{ Public declarations }
end;

var
IPR_ReCalcForm: TIPR_ReCalcForm;
implementation
uses Menu;
{$R *.dfm}

```

```

//-----
procedure TIPR_ReCalcForm.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
MenuForm.Show;
MenuForm.FormStyle := fsStayOnTop;
DBGrid1.DataSource := nil;
end;

//-----
procedure TIPR_ReCalcForm.FormShow(Sender:
TObject);
begin
ReCalcButton.Visible := false;
Memo1.Visible := false;
ReCalcMemo.Color := RGB(240, 240, 240);
ReCalcMemo.Font.Color := RGB(162, 73, 73);
Memo1.Color := RGB(240, 240, 240);

ADOQuery1.Connection :=
MenuForm.ADOConnection1;
DataSource1.DataSet := ADOQuery1;

ADOQuery1.SQL.Text := 'Select [Фінансові
вкладення] AS fin, [Захід №1] as z1, [Захід №2] as z2, '+
'[Захід №3] as z3, [Захід №4] as z4, [Захід №5] as z5
from [Матриця фінансових витрат];';

try
ADOQuery1.Open;
except
ShowMessage('Помилка в запиті ' +
ADOQuery1.SQL.Text);
exit;
end;

DBGrid1.DataSource := DataSource1;
DBGrid1.Columns[0].ReadOnly := true;
DBGrid1.Columns.Add;
DBGrid1.Columns.Add;
DBGrid1.Columns.Add;
DBGrid1.Columns.Add;
DBGrid1.Columns.Add;
DBGrid1.Columns.Add;

DBGrid1.Columns[0].FieldName := 'fin';
DBGrid1.Columns[1].FieldName := 'z1';
DBGrid1.Columns[2].FieldName := 'z2';
DBGrid1.Columns[3].FieldName := 'z3';
DBGrid1.Columns[4].FieldName := 'z4';
DBGrid1.Columns[5].FieldName := 'z5';

DBGrid1.Columns[0].Title.Caption := 'Фінансові
вкладення';
DBGrid1.Columns[1].Title.Caption := 'Захід №1';
DBGrid1.Columns[2].Title.Caption := 'Захід №2';
DBGrid1.Columns[3].Title.Caption := 'Захід №3';
DBGrid1.Columns[4].Title.Caption := 'Захід №4';
DBGrid1.Columns[5].Title.Caption := 'Захід №5';

(DBGrid1.Columns[0] as TColumn).Width := 120;
(DBGrid1.Columns[1] as TColumn).Width := 60;
(DBGrid1.Columns[2] as TColumn).Width := 60;
(DBGrid1.Columns[3] as TColumn).Width := 60;
(DBGrid1.Columns[4] as TColumn).Width := 60;
(DBGrid1.Columns[5] as TColumn).Width := 60;

ReCalcButton.Visible := true;
end;

//-----
procedure TIPR_ReCalcForm.Memo1Enter(Sender:
TObject);
begin
LoadButton.SetFocus;
end;

//-----

```

```

procedure
TIPR_ReCalcForm.ReCalcButton_Click(Sender: TObject);
var mes : String;
    Qv, Qnorm, Z : real;
begin
    DataSource1.DataSet.Edit;
    DataSource1.DataSet.Post;

    mes := 'Помилка при розрахунку пожежного ризику.';

    try
        // LabaForm.Show;
        // LabaForm.FormStyle := fsStayOnTop;

        LabaForm.beforeShow;
        LabaForm.Button2Click(LabaForm.Button2);
    except
        ShowMessage(mes);
        exit;
    end;

    if trim(ZEdit.Text) = '' then
    begin
        ShowMessage(mes);
        exit;
    end
    else
        ZEdit.Text := StringReplace(ZEdit.Text, ',', '');
[rfReplaceAll, rfIgnoreCase]);

    FormatSettings.DecimalSeparator := '.';

    try
        Qv := StrToFloat(Qv_Edit.Text);
        Z := StrToFloat(ZEdit.Text);
        Qnorm := StrToFloat(Qnorm_Edit.Text);

        if (Qv * Z) <= Qnorm then
        begin
            Memo1.Text := 'Розрахована величина
індивідуального пожежного ризику ' + FloatToStr(RoundTo((Qv *
Z * 1000), -3)) + '*10^-4 не перевищує нормативне значення.' +
            'Отже реалізована на підприємстві система
протипожежної безпеки є ефективною на поточний момент.' +
            'Однак, враховуючи вплив внутрішнього та
зовнішнього середовища (фізичний, моральний знос
обладнання,' +
            'зміна законодавства та т.ін.), необхідно наспаного
з встановленню періодичністю здійснювати оцінку ' +
            'пожежного ризику і надалі';

            Memo1.Font.Color := clGreen;
            Memo1.Font.Size := 8;
        end
        else
        begin
            Memo1.Text := 'Розрахована величина
індивідуального пожежного ризику ' + FloatToStr(RoundTo((Qv *
Z * 1000), -3)) + '*10^-4 перевищує нормативне значення.' +
            'Змініть коефіцієнти для перерахунку величини
індивідуального пожежного ризику';

            Memo1.Font.Color := clRed;
            Memo1.Font.Size := 10;
        end;

        Memo1.Visible := true;
    except
        ShowMessage(mes);
        exit;
    end;
end;

//-----
procedure TIPR_ReCalcForm.ReCalcMemoEnter(Sender:
TObject);
begin
    LoadButton.SetFocus;
end;

```

```

//-----
end.

unit Spr;
interface
uses
    Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB,
Vcl.StdCtrls, Data.Win.ADODB,
    Vcl.ExtCtrls, Vcl.DBGrids, Vcl.DBGrids;

type
    TSprForm = class(TForm)
        DataSource1: TDataSource;
        DBGrid1: TDBGrid;
        DBNavigator1: TDBNavigator;
        ADOConnection1: TADOConnection;
        ADOTable1: TADOTable;
        procedure FormShow(Sender: TObject);
        procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    private
        { Private declarations }
        procedure OpenTable;
    public
        { Public declarations }
    end;

var
    SprForm: TSprForm;

implementation
uses Menu;
{$R *.dfm}

//-----
procedure TSprForm.OpenTable;
var
    i : integer;
    DB_file : String;
    DB_table : String;
begin
    Case SprForm.Tag of
        9 : DB_table := 'Місця виникнення пожежі';
        10 : DB_table := 'Закономірності розповсюдження
пожежі';
        11 : DB_table := 'Початкова область пожежі';
        12 : DB_table := 'Параметри навколишнього
середовища';
        13 : DB_table := 'Початкові параметри приміщення';
        14 : DB_table := 'Список заходів';
    else exit;
    end;

    DB_file := ExtractFileDir(Application.ExeName) +
'db.accdb';

    ADOConnection1.ConnectionString :=
'Provider=Microsoft.ACE.OLEDB.12.0;Password=''; Data Source='
+ DB_file;
    ADOConnection1.LoginPrompt := False;
    DataSource1.DataSet := ADOTable1;
    ADOTable1.Connection := ADOConnection1;
    ADOTable1.TableName := DB_table;
    ADOTable1.Active := true;

    DBGrid1.DataSource := DataSource1;
    DBNavigator1.DataSource := DataSource1;

    {ширина 2й колонки для справочників, кроме
последнего}

    if SprForm.Tag <> 14 then
        (DBGrid1.Columns[1] as TColumn).Width := 350
    else
        begin

```

```

(DBGrid1.Columns[1] as TColumn).Width := 90;
(DBGrid1.Columns[2] as TColumn).Width := 350;
end;
end;

//-----
procedure TSprForm.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
  ADOConnection1.Connected := false;
  ADOTable1.Active := False;
  MenuForm.Visible := true;
end;

//-----
procedure TSprForm.FormShow(Sender: TObject);
begin
  Case SprForm.Tag of
    9 : SprForm.Caption := 'Довідник - Місяця
виникнення пожежі [Оцінка пожежного ризику]';
    10 : SprForm.Caption := 'Довідник - Закономірність
розповсюдження пожежі [Оцінка пожежного ризику]';
    11 : SprForm.Caption := 'Довідник - Початкова
область пожежі [Оцінка пожежного ризику]';
    12 : SprForm.Caption := 'Довідник - Параметри
навколишнього середовища [Оцінка пожежного ризику]';
    13 : SprForm.Caption := 'Довідник - Початкові
параметри приміщення [Оцінка пожежного ризику]';
    14 : SprForm.Caption := 'Довідник - Список заходів
щодо зниження пожежного ризику [Оцінка пожежного ризику]';
  else exit;
  end;

  OpenTable;
end;

//-----
end.

unit SelChart;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants, System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB,
Vcl.StdCtrls, Vcl.ComCtrls, DateUtils,
Vcl.Grids, Vcl.DBGrids, Data.Win.ADODB,
Vcl.Tee.TeeGDIPlus, VCL.Tee.TeEngine,
VCL.Tee.Series, Vcl.ExtCtrls, VCL.Tee.TeeProcs,
VCL.Tee.Chart;

type
  TSelDates4ChartForm = class(TForm)
    DBGrid1: TDBGrid;
    dtPicker1: TDateTimePicker;
    dtPicker2: TDateTimePicker;
    Label2: TLabel;
    SelDateMemo: TMemo;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    Chart1: TChart;
    Series1: TBarSeries;
  procedure FormClose(Sender: TObject; var Action:
TCloseAction);
  procedure FormShow(Sender: TObject);
  procedure SelDateMemoEnter(Sender: TObject);
  procedure DBGrid1 CellClick(Column: TColumn);
  procedure dtPicker1Change(Sender: TObject);
  procedure dtPicker2Change(Sender: TObject);
  private
    { Private declarations }
  procedure SelectQuery(dt1, dt2 :String);
  public
    { Public declarations }
  end;

var
  SelDates4ChartForm: TSelDates4ChartForm;
implementation
uses IPRCalc, Menu;
{$R *.dfm}

//-----
procedure
TSelDates4ChartForm.DBGrid1 CellClick(Column: TColumn);
var kod : String;
x, y : real;
begin
  if DataSource1.DataSet.RecordCount > 0 then
  begin
    kod :=
DBGrid1.DataSource.DataSet.Fields.Fields[0].AsString;

    MenuForm.ADOQuery1.SQL.Text := 'Select [Qв] AS a
' +
'from [Оцінка пожежного ризику] ' +
'WHERE [id_operation] = ' + kod;

    try
      MenuForm.ADOQuery1.Open;
    except
      ShowMessage('Помилка в запиті: ' +
MenuForm.ADOQuery1.SQL.Text);
    end;

    MenuForm.ADOQuery1.First;
    Series1.Clear;

    x := 0.5;

    while not MenuForm.ADOQuery1.Eof do
    begin
      y :=
MenuForm.ADOQuery1.FieldByName('a').AsFloat * 1000;
      Series1.AddXY(x, y, '');

      x := x + 0.5;
      MenuForm.ADOQuery1.Next;
    end;

    Chart1.Visible := true;
    DBGrid1.Visible := false;
    Chart1.BringToFront;
    Chart1.Align := alClient;
    Self.Constraints.MaxHeight := 0;
    Self.Constraints.MinHeight := 0;
    Self.Constraints.MaxWidth := 0;
    Self.Constraints.MinWidth := 0;
    Self.BorderIcons := Self.BorderIcons + [biMaximize];
  end;
end;

//-----
procedure
TSelDates4ChartForm.dtPicker1Change(Sender: TObject);
begin
  SelectQuery(DateToStr(dtPicker1.Date),
DateToStr(dtPicker2.Date));
end;

//-----
procedure
TSelDates4ChartForm.dtPicker2Change(Sender: TObject);
begin
  SelectQuery(DateToStr(dtPicker1.Date),
DateToStr(dtPicker2.Date));
end;

//-----
procedure
TSelDates4ChartForm.FormClose(Sender:
TObject;
var Action: TCloseAction);
begin
  Hide;
  IPRCalcForm.Show;

```

```

        IPRCalcForm.FormStyle := fsStayOnTop;
    end;
    //-----
    procedure TSelDates4ChartForm.FormShow(Sender:
TObject);
    begin
        Chart1.Visible := false;
        Chart1.BringToFront;
        Chart1.Align := alClient;
        DBGrid1.Visible := true;
        PostMessage(Self.Handle, WM_SYSCOMMAND,
SC_RESTORE, 0);

        Self.Constraints.MaxHeight := 450;
        Self.Constraints.MinHeight := 450;
        Self.Constraints.MaxWidth := 600;
        Self.Constraints.MinWidth := 600;

        Self.BorderIcons := Self.BorderIcons - [biMaximize];

        SelDateMemo.Color := RGB (240, 240, 240);
        SelDateMemo.Font.Color := RGB (133, 10, 10);

        {ограничиваю выбор дат}
        dtPicker1.MinDate := StrToDate('01.05.2018');
        dtPicker2.MinDate := StrToDate('01.05.2018');
        dtPicker1.MaxDate := Date; //сегодня
        dtPicker2.MaxDate := IncDay(Date, +1); //завтра
        dtPicker1.Date := IncDay(Date, -1); //вчера
        dtPicker2.Date := Date;

        {соединяюсь с БД}
        ADOQuery1.Connection :=
MenuForm.ADOConnection1;
        DataSource1.DataSet := ADOQuery1;
        DBGrid1.DataSource := DataSource1;
        DBGrid1.ReadOnly := true;

        SelectQuery(DateToStr(dtPicker1.Date),
DateToStr(dtPicker2.Date));
    end;
    //-----
    procedure
TSelDates4ChartForm.SelDateMemoEnter(Sender: TObject);
    begin
        DBGrid1.SetFocus;
    end;
    //-----
    procedure TSelDates4ChartForm.SelectQuery(dt1, dt2
:String);
    begin
        ADOQuery1.SQL.Text := 'select [id_operation],
count(*) as [Кількість сценаріїв], [dt] as [Дата та час] from' +
'[Оцінка пожежного ризику]' +
' where [dt] >= Format('' + dt1 + '
00:00:00'', ''dd.mm.yyyy hh:nn:ss'') and [dt] <= Format('' + dt2 + '
23:59:59'', ''dd.mm.yyyy hh:nn:ss'') +
' and [dt] >= Format('' + DateToStr(dtPicker1.MinDate)
+ ' 00:00:00'', ''dd.mm.yyyy hh:nn:ss'') +
' and [dt] <= Format('' + DateToStr(dtPicker1.MaxDate)
+ ' 23:59:59'', ''dd.mm.yyyy hh:nn:ss'') +
//' and [id_operation] <> ' + IntToStr(MenuForm.Tag) +
' group by [id_operation], [dt] ' +
' order by [dt] desc;';
        try
            ADOQuery1.Open;
        except
            Label2.Visible := false;
            ShowMessage('Помилка в запиті: ' +
ADOQuery1.SQL.Text);
            exit;
        end;

        Label2.Visible := true;
        Label2.Caption := 'Знайдено варіантів: ' +
IntToStr(DataSource1.DataSet.RecordCount) + ' ';

        DBGrid1.Columns[0].Visible := false;
        (DBGrid1.Columns[1] as TColumn).Alignment :=
taLeftJustify;
        (DBGrid1.Columns[1] as TColumn).Width := 100;
        (DBGrid1.Columns[2] as TColumn).Width := 200;
    end;
    //-----
    end.
end.

```

ДОДАТОК Г. Перелік наукових праць автора

1. Розробка програмного забезпечення для оцінки пожежного ризику на АЗС / **«Пожежний ризик»** // Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: матеріали Всеукраїнської науково-практичної Internet-конференції. – Черкаси, 2019. – С. – 79–81.
2. Оцінка пожежного ризику: методика та інструментарій / **«Пожежний ризик»** // Актуальні задачі сучасних технологій: зб. тез доповідей міжнар. наук.-техн. конф. Молодих учених та студентів, (Тернопіль, 28–29 листоп. 2018.) / М-во освіти і науки України, Терн. націон. техн. ун-т ім. І. Пулюя [та ін]. – Тернопіль : ТНТУ, 2018. – С. 196–197.
3. Автоматизація процесу оцінки пожежного ризику на АЗС / **«Пожежний ризик»** // Технологія-2019 : XXII матеріали міжнар.наук.-техн. конф., 26-27 квіт. 2019 р., м. Сєверодонецьк. Ч. II / [укл. : Тарасов В.Ю.]. – Сєверодонецьк : [Східноукр. нац. ун-т ім. В. Даля], 2019. – С. 52–53.
4. Оцінка пожежного ризику на АЗС: моделювання, аналіз, прийняття рішень / **«Пожежний ризик»** // Вісник економічної науки України. Науковий журнал. – № 2(35), 2018. – С.167–170.