

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ

Навчально-науковий інститут денної освіти

Форма навчання денна

Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри

Олена ОЛЬХОВСЬКА

(підпис)

«___»_____202_р

КВАЛІФІКАЦІЙНА РОБОТА

на тему

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З ТЕМИ

«ВИЗНАЧНИКИ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ

«ПРИКЛАДНА МАТЕМАТИКА»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня «бакалавр»

Виконавець роботи Кадесніков Олександр Васильович

«___»_____2023 р.

(підпис)

Науковий керівник доцент, к.пед.н. Оксана КОШОВА

«___»_____2023 р.

(підпис)

Рецензент

ПОЛТАВА 2023

Затверджую
Завідувач кафедри _____ Олена Ольховська
“ _____ ” _____ 2023 р.

ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

**на тему «Розробка програмного забезпечення тренажеру з теми «Визначники»
дистанційного навчального курсу «Прикладна математика»**

Зі спеціальності 122 «Комп'ютерні науки»

Освітня програма «Комп'ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові _Кадесніков Олександр Васильович

Затверджена наказом ректора № ____ -Н від « ____ » _____ 202_ р.

Термін подання студентом курсового проекту « ____ » _____ 202_ р.

Вихідні дані до кваліфікаційно роботи: публікації з теми, навчальні
тренажери в дистанційних курсах із комп'ютерних наук.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1. ПОСТАНОВКА ЗАВДАННЯ

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд робіт з аналогічними завданнями і реалізацією

2.2. Переваги та труднощі у використанні оглянутих програм

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Алгоритм роботи першої сторінки тренажера

3.2. Алгоритм пошуку визначника першого порядку, мінора та алгебраїчного
доповнення матриці.

3.3. Обчислення визначника другого порядку

3.4. Обчислення визначника третього порядку

4. ПРАКТИЧНА ЧАСТИНА

4.1. Блок схеми алгоритмів

4.2. Головне вікно програми

4.3. Завдання 1

4.3.1. Завдання 1 Крок 1

4.3.2. Завдання 1 Крок 2

4.3.3. Завдання 1 Крок 3

4.3.4. Завдання 1 Крок 4

4.3.5. Завдання 1 Крок 5

4.4. Завдання 2

4.4.1. Завдання 2 Крок 1

4.4.2. Завдання 2 Крок 2

4.4.3. Завдання 2 Крок 3

4.5. Завдання 3

4.5.1. Завдання 3 Крок 1

4.5.2. Завдання 3 Крок 2

4.5.3. Завдання 3 Крок 3
ВИСНОВОК
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ
ДОДАТОК А

Консультанти розділів кваліфікаційної роботи

Розділ	ПІБ, посада консультанта	Підпис, дата	
		Завдання дав	Завдання прийняв
Постановка задачі			
Інформаційний огляд			
Теоретична частина			
Практична реалізація			

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій і стандартів та звіт керівнику		
3. Постановка завдання		
4. Інформаційний огляд джерел бібліотек та Інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доопрацювання (за необхідності), рецензування		

Дата видачі завдання «__» _____ 202_ р.

Здобувач вищої освіти _____

Науковий керівник доцент к. пед. н., Кошова О.П.

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____

(балів, оцінка за національною шкалою, оцінка за ЄКТС)

Протокол засідання ЕК № _____ від «__» _____ 2023 р.

Секретар ЕК _____

РЕФЕРАТ

Записка: 78 с., 55 рисунків, 1 таблиця, 9 джерел.

ДИСТАНЦІЙНА ОСВІТА, ТРЕНАЖЕР, ВИЗНАЧНИКИ, МАТРИЦІ,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Мета роботи – сформулювати вимоги, структуру та програмно реалізувати програмний тренажер. Приділити увагу зручності інтерфейсу, простоті експлуатації та мінімальному використанню ресурсів ПК.

Об'єкт розробки – програмне забезпечення тренажеру з теми «визначники» дистанційного навчального курсу «прикладна математика»

Методи, котрі були використані для вирішення задачі. Для написання програми було обрано мову C# та середовище для створення візуальних програм Microsoft Visual Studio.

Розроблено алгоритм елемента навчального тренажера з теми «Визначники», обрана методологія практичної реалізації тренажеру, створені блок-схеми компонентів тренажеру, виконана програмна реалізація готового рішення.

Результати роботи можуть бути використані у навчальному процесі ПУЕТ при вивченні курсу «Прикладна математика».

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП.....	7
1. ПОСТАНОВКА ЗАДАЧІ.....	9
2. ТЕОРЕТИЧНА ЧАСТИНА	11
2.1 Алгоритм роботи першої сторінки тренажера	11
2.2. Алгоритм пошуку визначника першого порядку, мінора та алгебраїчного доповнення матриці.	12
2.3. Обчислення визначника другого порядку	19
2.4. Обчислення визначника третього порядку	21
3. ПРАКТИЧНА ЧАСТИНА	24
3.1. Блок схеми алгоритмів.....	24
3.2 Головне вікно програми	25
3.3 Завдання 1	27
3.3.1 Завдання 1 Крок 1.....	27
3.3.2 Завдання 1 Крок 2.....	32
3.3.3 Завдання 1 Крок 3.....	36
3.3.4 Завдання 1 Крок 4.....	41
3.3.5 Завдання 1 Крок 5.....	49
3.4 Завдання 2	55
3.4.1 Завдання 2 Крок 1.....	55
3.4.2 Завдання 2 Крок 2.....	57
3.4.3 Завдання 2 Крок 3.....	59
3.5 Завдання 3	63
3.5.1 Завдання 3 Крок 1.....	63
3.5.2 Завдання 3 Крок 2.....	64
3.5.3 Завдання 3 Крок 3.....	66
Висновки	71
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	72
Додаток А.....	73

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ, ТЕРМІНІВ**

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, символів, скорочень, термінів
C#	Мова програмування загального призначення. Підтримує наступні парадигми програмування: • процедурне програмування; • об'єктно-орієнтоване програмування; • узагальнене програмування.
Microsoft Visual Studio	Багатофункціональне інтегроване середовище програмування; забезпечує швидкість візуальної розробки, продуктивність повторно використовуваних компонентів у сполученні з потужністю засобів мови C#.
Windows Forms .NET	Платформа користувацького інтерфейсу для створення різноманітних користувацьких застосунків робочого столу в рамках операційної системи Windows. Платформа розробки підтримує широкий комплект функцій для створення застосунків, включно з елементами керування, графікою, прив'язкою елементів та вводом даних користувачем
Мінор	Математична функція, котра обчислюється шляхом видалення з матриці стовпчика і рядка елемента, мінор котрого шукається.
ОЗП	Оперативний запам'ятовуючий пристрій. Також має назву RAM (read-access memory) – елемент комп'ютера та пристрій, що зберігає в собі код програми під час її роботи.
Помилковий ввід даних	Одрукування користувача при вводі відповіді чи свідомо спроба ввести недоступне значення у поле для вводу.
ПУЕТ	Полтавський Університет Економіки та Торгівлі.
Форма (Form, Windows Form)	Структурна одиниця програми. Котра дозволяє мінімізувати процедуру створення інтерфейсу і працює в середовищі операційної системи Windows, являючи собою її компонент.

ВСТУП

Актуальність. В умовах пандемії, а потім і війни важливу роль у світі почали відігравати процеси дистанційного характеру – робота, освіта, покупки та багато іншого. З розвитком технологій і бажання людей мати доступ до необхідної інформації незалежно від місця перебування актуальність програм, котрі використовуються для навчання та повторення тем, тільки зростатиме. Тому використання тренажерів з різних тем буде тільки зростати, оскільки вони дозволяють набувати нових знань за короткий час та з будь-якого місця.

Мета роботи – сформулювати вимоги, структуру та програмно реалізувати програмний тренажер. Приділити увагу зручності інтерфейсу, простоті експлуатації та мінімальному використанню ресурсів ПК.

Задача роботи - з'ясування структурних та програмних особливостей розробки навчальних тренажерів. Створення програми-тренажера для відпрацювання студентами практичних навичок. Додання до програми компонента оцінки засвоєння матеріалу, оскільки тренажер виконує навчальну функцію.

Об'єкт проекту - інформаційні технології у вищих навчальних закладах.

Предмет – створення програмного забезпечення для вивчення та повторення матеріалів з вказаної теми.

Методи, які використовувались – для написання програми було обрано мову C# та середовище для створення візуальних програм Visual Studio 2022. Методом реалізації програми обрано «Microsoft Forms» - універсальний засіб розробки інтерфейсу програми обраною мовою. Створена програма може бути використана у навчальному процесі ПУЕТ при вивченні курсу «Прикладна математика».

Дипломний проект містить наступні складові: вступ, постановка задачі, теоретична та практична частина, висновки та список літературних джерел. В теоретичній частині наведені алгоритми елементів тренажера. Практична частина присвячена програмній реалізації даних алгоритмів. Вона містить блок-схеми, опис програми, код програми, розбір його функціоналу.

Обсяг проекту складає 78 сторінок.

1. ПОСТАНОВКА ЗАДАЧІ

Під час написання курсового проекту необхідно звернути увагу на наступні питання:

- складання алгоритму елементів тренажера;
- розглянути літературні джерела, які досліджують питання теми;
- проаналізувати та порівняти результати отримані іншими авторами в ході своїх досліджень;
- написання програми-тренажера з теми.

Вимоги до тренажера:

1. На стартовій сторінці користувачу пропонується обрати задачу з наступного переліку задач:

- «Весь тест». Дозволяє користувачу пройти всі завдання у логічній послідовності вивчення теми;
- «Завдання 1». Обчислення мінора (визначника першого порядку) та алгебраїчного доповнення визначника матриці;
- «Завдання 2». Обчислення визначника другого порядку;
- «Завдання 3». Обчислення визначника третього порядку.

2. Умова завдання максимально довго перебуває на екрані. Оскільки, для повідомлення про помилку використовується переважно метод «MessageBox» - діалогове вікно може закривати частиною задачі собою, при використанні програми на екранах з низькою роздільною здатністю.

3. Користувач може викликати «Підказку», котра містить інформацію, що здатна полегшити виконання завдання і містить в собі додаткові або основні теоретичні відомості.

4. На кожному кроці алгоритму в разі неправильного виконання завдань з'являється повідомлення про помилку, у якому вказується правильна відповідь. Якщо це можливо – неправильні відповіді при цьому виділяються.

Практична складова дипломної роботи передбачає програмну реалізацію тренажера для відпрацювання навичок обчислення визначників першого, другого

та третього порядків.

Програмний комплекс повинен включати в себе теоретичну складову у вигляді додаткових запитань із запропонованими відповідями в разі помилки.

Тренажер для обчислення визначників матриці має бути орієнтований більшою мірою на самостійну роботу студентів. Він має бути зручним для користувача та містити завдання, які максимально охоплюють дану тему.

2. ТЕОРЕТИЧНА ЧАСТИНА

2.1 Алгоритм роботи першої сторінки тренажера

Перша сторінка повинна містити тему тренажера та мати основні навігаційні кнопки, котрі дозволять переключатись між завданнями. Було прийняти рішення виділити окрему кнопку для запуску всього тестування та окремо для кожного завдання. Також, це вікно містить кнопку, котра дозволить завершити виконання програми.

Користувач має обрати чи проходить він завдання тренажеру послідовно і всі підряд, чи його цікавить повторення конкретних завдань. Оскільки детальний опис може бути громіздким, при наведенні на кнопку з відповідним тестом під курсором з'являється підказка, котра дає опис тестування, що буде проходитись користувачем.

Підказки мають наступний опис:

- «Почати тестування» – «Запускає всі завдання по черзі, дозволяючи перевірити знання з теми «Визначники» курсу «Прикладна математика» або набути нових знань»;
- «Завдання 1» – «Запускає виключно перший блок тесту, котрий містить матеріали для повторення теми «пошук визначника першого порядку, мінора та алгебраїчного доповнення матриці»;
- «Завдання 2» – «Запускає виключно другий блок тесту, котрий містить матеріали для повторення теми «Обчислення визначника другого порядку»;
- «Завдання 3» – «Запускає виключно другий блок тесту, котрий містить матеріали для повторення теми «Обчислення визначника третього порядку».

Залежно, від вибору користувача буде запускатись виключно обране тестування або весь тест разом.

Кнопка «Закрити тренажер», як і закриття вікна слугує для закриття

тренажеру. Через деякі особливості Windows Forms – це єдиний спосіб закрити тренажер для користувача, використовуючи виключно засоби самого тренажеру. Для всіх подальших вікон кнопки «закрити завдання» та кнопка закриття (як і натискання Alt+F4) будуть виключно викликати перше вікно тренажеру.

2.2. Алгоритм пошуку визначника першого порядку, мінора та алгебраїчного доповнення матриці.

Завдання 1. Задано матрицю

$$\begin{matrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{matrix}$$

Визначити мінор M_{11} для елемента x_{11}

Визначити алгебраїчне доповнення для елемента x_{11}

Алгоритм для завдання 1

Крок 1. На екрані з'являється завдання: серед вказаних мінорів елементів матриці оберіть той, що відповідає елементу x_{11} . Користувач має відмітити клітинку, котру вважає правильною відповіддю.

$$\square M_{11} = \begin{matrix} x_{12} & x_{13} \\ x_{22} & x_{33} \end{matrix} \quad \square M_{11} = \begin{matrix} x_{11} & x_{12} \\ x_{21} & x_{32} \end{matrix} \quad \square M_{11} = \begin{matrix} x_{22} & x_{23} \\ x_{32} & x_{33} \end{matrix} \quad \square M_{11} = \begin{matrix} x_{21} & x_{22} \\ x_{31} & x_{32} \end{matrix}$$

Рис. 2.2.1 - Крок 1 Завдання 1

Правильна відповідь – перехід на крок 2. Якщо введено неправильні значення, то з'являється повідомлення про помилку:

- Якщо вибрана відповідь $\begin{matrix} x_{12} & x_{13} \\ x_{22} & x_{33} \end{matrix}$ текст повідомлення має вигляд: « $\begin{matrix} x_{12} & x_{13} \\ x_{22} & x_{33} \end{matrix}$ – це мінор для елемента матриці x_{31} .»
- Якщо вибрана відповідь $\begin{matrix} x_{11} & x_{12} \\ x_{21} & x_{32} \end{matrix}$ текст повідомлення має вигляд: «Вказана відповідь не є мінором для будь-якого елемента заданої матриці»
- Якщо вибрана відповідь $\begin{matrix} x_{21} & x_{22} \\ x_{31} & x_{32} \end{matrix}$ текст повідомлення має вигляд: « $\begin{matrix} x_{21} & x_{22} \\ x_{31} & x_{32} \end{matrix}$ – це мінор для елемента матриці x_{13} .»

Повідомлення про помилку повертає користувача на Крок 1 і дозволяє обрати відповідь наново.

Крок 2. З'являється завдання «Вкажіть якому елементу матриці відповідають наступні мінори».

Користувачу надається можливість обрати відповіді з випадючих списків для кожного прикладу:

- $\begin{matrix} x_{11} \\ x_{31} \end{matrix}$
- $\begin{matrix} x_{11} \\ x_{31} \end{matrix}$
- $\begin{matrix} x_{21} \\ x_{31} \end{matrix}$
- $\begin{matrix} x_{12} \\ x_{32} \end{matrix}$

Рис. 2.2.2 - Крок 2 Завдання 1

M_{12}
 M_{31}
 M_{11}
 M_{23}
 M_{22}
 M_{13}
 M_{21}

Рис. 2.2.3 - Крок 2 Завдання 1 -- випадючий список мінорів для елементів матриці

Після вибору всіх назв можливі 2 подальші варіанти:

1. Всі відповіді правильні – перехід на Крок 3.
2. Якщо допущено хоча б одну помилку з'являється вікно з повідомленням.

Текст повідомлення:

«При виборі допущені помилки. $\begin{matrix} x_{11} & x_{12} \\ x_{31} & x_{32} \end{matrix}$ – це мінор M_{23} , $\begin{matrix} x_{11} & x_{13} \\ x_{31} & x_{33} \end{matrix}$ – це мінор M_{22} , $\begin{matrix} x_{21} & x_{22} \\ x_{31} & x_{32} \end{matrix}$ – це мінор M_{13} , $\begin{matrix} x_{12} & x_{13} \\ x_{32} & x_{33} \end{matrix}$ – це мінор M_{21} .»

Примечание [a1]: Може додати наступну опцію: виводити під час виконання завдання теоретичну частину? Наприклад «що таке мінор елемента»

Неправильні відповіді варто виділити у відповідних випадаяючих списках червоним кольором, щоб користувач міг виправити відповіді і перейти до наступного кроку.

Крок 3. З'являється завдання «розташуйте в коректному порядку елементи формули для визначення алгебраїчного доповнення елемента x_{11} ».

Користувачу необхідно заповнити клітинки (значеннями 1, 2 та 3) (рис. 2.2.4)

$$\square A_{11} = \square \begin{bmatrix} x_{22} & x_{23} \\ x_{32} & x_{33} \end{bmatrix} \square ((-1)^{1-1})^*$$

Рис. 2.2.4 - Крок 3 Завдання 1 - Порядок обчислення алгебраїчного доповнення елемента x_{11}

Після заповнення всіх комірок можливі наступні наслідки:

1. Заповнення коректне (можливі 2 правильні відповіді – рис. 2.2.5) і відбувається перехід на крок 4.

$\square 1. A_{11} =$	$\square 1. A_{11} =$
$\square 2. (-1)^{1-1}^*$	$\square 2. \begin{bmatrix} x_{22} & x_{23} \\ x_{32} & x_{33} \end{bmatrix}$
$\square 3. \begin{bmatrix} x_{22} & x_{23} \\ x_{32} & x_{33} \end{bmatrix}$	$\square 3. (-1)^{1-1}^*$

Рис. 2.2.5 - Крок 3 Завдання 1 - Можливі два варіанти правильного вибору на кроці 3 завдання 1

2. Заповнення виконане невірно. Користувач отримує повідомлення про допущену помилку.

Текст повідомлення:

«Формула алгебраїчного доповнення означає, що алгебраїчне доповнення дорівнює результату множення мінора відповідного елемента на $(-1)^{i-j}$ ».

Перехід на крок 4.

Крок 4. З'являється завдання «Дано наступну матрицю:

$$\begin{bmatrix} 4 & 7 \\ 1 & 3 \end{bmatrix}$$

Заповніть таблицю для обчислення алгебраїчного доповнення наступних елементів: x_{11} , x_{22} та x_{12} ».

Користувач має ввести у таблицю (рис. 2.6) правильні значення, котрі може обчислити за відповідною формулою.

Елемент	$(-1)^{i-j}$	Мінор елемента	Алгебраїчне доповнення
X_{11}			
X_{22}			
X_{12}			

Рис. 2.2.6 - Крок 4 Завдання 1 - Таблиця значень для обчислення алгебраїчного доповнення відповідних елементів

Правильна відповідь – перехід до наступної чарунки. В іншому випадку з'являється повідомлення про помилку.

Якщо користувач вводить в чарунку не число, а текстову відповідь, з'являється наступне повідомлення «Вказано недопустиме значення. Введіть додатне чи від'ємне число».

Тексти про помилку для різних випадків:

Стовпець $(-1)^{i-j}$

Елемент x_{11} Значення крім «1» – «Помилка. Ступінь «i-j» для елемента x_{11} обчислюється як $(1-1)$ »

Елемент x_{22} Значення крім «1» - «Помилка. Ступінь «i-j» для елемента x_{22} обчислюється як $(2-2)$ »

Елемент x_{12} Значення крім «-1» - «Помилка. Ступінь «i-j» для елемента x_{22} обчислюється як $(1-2)$. Від'ємна ступінь вказує, що треба виконати ділення 1 на число у вказаному ступені».

Стовпець «Мінор елемента»

Елемент x_{11} . Значення крім «3» - «Помилка. Мінор елемента дорівнює значенню матриці, котра не включає в себе строку та стовпчик, що дорівнюють позиції елемента. Тобто треба прибрати стовпчик 1 і строку 1. Залишається значення «3», елемент x_{22} початкової матриці»

Елемент x_{22} . Значення крім «4» - «Помилка. Мінор елемента дорівнює значенню матриці, котра не включає в себе строку та стовпчик, що дорівнюють позиції елемента. Тобто треба прибрати стовпчик 2 і строку 2. Залишається значення «4», елемент x_{11} початкової матриці».

Елемент x_{12} . Значення крім «1» - «Помилка. Мінор елемента дорівнює значенню матриці, котра не включає в себе строку та стовпчик, що дорівнюють позиції елемента. Тобто треба прибрати стовпчик 2 і строку 1. Залишається значення «1», елемент x_{21} початкової матриці»

Стовпець «Алгебраїчне доповнення»

Елемент x_{11} Значення крім «3» – «Помилка. Алгебраїчне доповнення є добутком від множників у відповідних елементу стовпцях $(-1)^{i-j}$ та мінор елемента. Тобто $1*3 = 3$ ».

Елемент x_{22} Значення крім «4» – «Помилка. Алгебраїчне доповнення є добутком від множників у відповідних елементу стовпцях $(-1)^{i-j}$ та мінор елемента. Тобто $1*4 = 4$ ».

Елемент x_{12} Значення крім «-1» – «Помилка. Алгебраїчне доповнення є добутком від множників у відповідних елементу стовпцях $(-1)^{i-j}$ та мінор елемента. Тобто $-1*1 = -1$ ».

Коли всі чарунки у таблиці заповнені правильно відбувається перехід до кроку 5.

Крок 5. На екрані з'являється завдання «Оберіть «Так» або «Ні» під кожним з наступних тверджень»

Чи вірне твердження: «Мінором M_{ij} елемента a_{ij} називається визначник $(n-1)$ -го порядку отриманий з визначника n -го порядку викресленням i -го рядка і j -го стовпчика».

☐ Так

☐ Ні

Чи є вираз вірним? Для елемента a_{12} у матриці $\begin{bmatrix} 12 & 6 \\ 3 & 9 \end{bmatrix}$ мінором, або визначником першого порядку є «6».

☐ Так

☐ Ні

Чи вірно наведена формула? Алгебраїчне доповнення A_{ih} для елемента a_{ij} визначається за формулою: $A_{ij} = (-1)^{i-j} * M_{ij}$

☐ Так

☐ Ні

Чи вірним є вираз: Для елемента A_{22} у матриці $\begin{bmatrix} 12 & 6 \\ 3 & 9 \end{bmatrix}$ алгебраїчне доповнення дорівнює «12».

☐ Так

☐ Ні

Рис. 2.2.7 - Крок 5 Завдання 1 – Коротке опитування на тему

Правильна відповідь – перехід до наступного питання. В іншому випадку з'являється відповідне до питання повідомлення.

Текст повідомлення для різних випадків:

Питання: «Міномор M_{ij} елемента a_{ij} називається визначник $(n-1)$ -го порядку отриманий з визначника n -го порядку викресленням i -го рядка і j -го стовпчика». Користувач надає відповідь «Ні». Повідомлення: «Міномор M_{ij} елемента a_{ij} називається визначник $(n-1)$ -го порядку отриманий з визначника n -го порядку викресленням i -го рядка і j -го стовпчика. Вказане твердження є вірним»

Питання: «Чи є вираз вірним? Для елемента a_{12} у матриці $\begin{bmatrix} 12 & 6 \\ 3 & 9 \end{bmatrix}$ міномор, або визначником першого порядку є 6». Користувач відповідає «Так». Повідомлення: «Міномор M_{ij} елемента a_{ij} називається визначник $(n-1)$ -го порядку отриманий з визначника n -го порядку викресленням i -го рядка і j -го стовпчика. Значення 6 відповідає міному елемента a_{21} . Міномор для елемента a_{12} є 9».

Питання: «Чи вірно наведена формула? Алгебраїчне доповнення A_{ij} для елемента a_{ij} визначається за формулою: $A_{ij} = (-1)^{i-j} * M_{ij}$ ». Відповідь користувача «Ні». Повідомлення: «Вказана формула : $A_{ij} = (-1)^{i-j} * M_{ij}$ є правильною».

Питання: Чи вірним є вираз: Для елемента a_{22} у матриці $\begin{bmatrix} 12 & 6 \\ 3 & 9 \end{bmatrix}$ алгебраїчне доповнення дорівнює «12». Користувач відповідає «ні». Повідомлення: «Алгебраїчне доповнення для елемента a_{22} це добуток $(-1)^{2-2} = (-1)^0 = 1$ та мінора елемента a_{22} , що дорівнює 12. Наведене твердження було вірним».

Після заповнення всіх чарунок правильними відповідями можливі 2 сценарії:

1. Якщо на початку роботи тренажеру, користувач обирає «Почати

тестування», то відбувається перехід до наступного завдання.

2. Якщо на початку роботи тренажеру користувач обрав проходження конкретного завдання – його поверне на початкове вікно тренажеру.

2.3. Обчислення визначника другого порядку

Завдання 2. Задано матрицю

$$\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}$$

Знайти визначник другого порядку для вказаної у завданні матриці.

Алгоритм для завдання 1

Крок 1. На екрані з'являється завдання: «Серед вказаних формул оберіть ту, що відповідає обчисленню визначника другого порядку для вказаної матриці». Користувач має відмітити правильну відповідь (рис 2.3.1)

- ☐ $\Delta = x_{11} * x_{12} + x_{11}/x_{12}$
- ☐ $\Delta = x_{22} * x_{12} + x_{11} * x_{21}$
- ☐ $\Delta = x_{11} * x_{22} + x_{12} * x_{21}$
- ☐ $\Delta = x_{11} * x_{22} - x_{12} * x_{21}$

Рис. 2.3.1 - Крок 1 Завдання 2 - завдання на вибір правильної формули для обчислення визначника другого порядку

звірно

користувач отримує повідомлення: «Визначник другого порядку обчислюється як різницю добутків основної діагоналі та зворотної. У наведеній матриці це $\Delta = x_{11} * x_{22} - x_{12} * x_{21}$ ».

Крок 2. На екрані з'являється 4 приклади матриць та текст: «Для кожної з наведених матриць оберіть правильне значення визначника серед доступних для заповнення значень».

У випадуючих списках можливо обрати наступні значення: 1, 2, 3, 4

$\begin{pmatrix} 1 & -1 \\ 2 & 2 \end{pmatrix}$	
$\begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}$	
$\begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix}$	
$\begin{pmatrix} -1 & -2 \\ 2 & 1 \end{pmatrix}$	

Рис. 2.3.2 - Крок 2 Завдання 2 - поле для виконання завдання

В залежності від правильності відповідей:

1. Всі відповіді правильні – перехід на крок 3.
2. Якщо здійснено хоча б один неправильний вибір – повідомлення про помилку.

Текст повідомлення:

«При виборі допущено помилки. Визначник другого порядку першої матриці – 4, другої – 1, третьої – 2, четвертої – 3».

При цьому неправильні відповіді у комірках виділити іншим кольором (наприклад, червоним) та надати можливість користувачу повторно виконати вибір. Перехід на крок 3.

Крок 3. На екрані з'являється 4 приклади матриць та текст завдання: «Серед наведених матриць відмітьте ті, що мають визначник другого порядку, котрий дорівнює 1».

$$\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 2 & 2 \\ 3 & 1 \end{pmatrix} \quad \begin{pmatrix} 2 & 1,5 \\ 2 & 2 \end{pmatrix} \quad \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix}$$

В залежності від правильності відповідей:

1. Всі відповіді правильні – завдання завершено.
2. Якщо здійснено хоча б один неправильний вибір – повідомлення про помилку.

Текст повідомлення:

«При виборі допущено помилки. Визначник другого порядку є різницею між добутками основної та зворотної діагоналей. Він дорівнює 1 для третьої та четвертої матриці.».

При цьому неправильні відповіді у комірках виділити іншим кольором (наприклад, червоним) та надати можливість користувачу повторно виконати вибір.

Після заповнення всіх чарунок правильними відповідями можливі 2 сценарії:

1. Якщо на початку роботи тренажеру, користувач обирає «Почати тестування», то відбувається перехід до наступного завдання.

2. Якщо на початку роботи тренажеру користувач обрав проходження конкретного завдання – його поверне на початкове вікно тренажеру.

2.4. Обчислення визначника третього порядку

Завдання 2. Задано матрицю

$$\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{32} & a_{33} \end{matrix}$$

Знайти визначник третього порядку для вказаної у завданні матриці.

Алгоритм для завдання 1

Крок 1. На екрані з'являється завдання: «Серед вказаних формул оберіть ту, що відповідає обчисленню визначника третього порядку для вказаної матриці». Користувач має відмітити правильну відповідь (рис .

- ☐ $\Delta = a_{11} * a_{22} * a_{33} + a_{13} * a_{21} * a_{32} + a_{31} * a_{12} * a_{23} - a_{13} * a_{22} * a_{13} - a_{33} * a_{12} * a_{21} - a_{11} * a_{23} * a_{32}$
- ☐ $\Delta = a_{11} * a_{22} * a_{33} + a_{13} * a_{21} * a_{32} + a_{31} * a_{12} * a_{23} + a_{13} * a_{22} * a_{13} + a_{33} * a_{12} * a_{21} + a_{11} * a_{23} * a_{32}$
- ☐ $\Delta = a_{11} * a_{22} * a_{33} + a_{13} * a_{21} * a_{32} + a_{31} * a_{12} * a_{23} - a_{13}/a_{22}/a_{13} - a_{33}/a_{12}/a_{21} - a_{11}/a_{23}/a_{32}$
- ☐ $\Delta = x_{11} * x_{22} - x_{12} * x_{21}$

Рис. 2.4.1 - Крок 1 Завдання 3 - завдання на вибір правильної формули для обчислення визначника другого порядку

Правильна відповідь – перехід на крок 2. Якщо відповідь надана невірно користувач отримує повідомлення: «Визначник третього порядку для заданої матриці визначається за формулою 1».

Крок 2. На екрані з'являється зображення «Поставте правильні алгебраїчні знаки у наведеному схематичному зображенні формули обчислення визначника третього порядку».

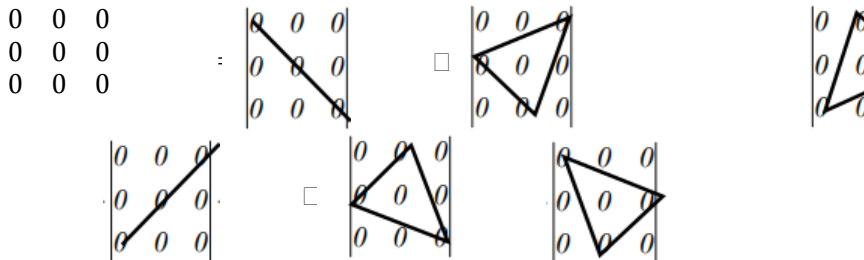


Рис. 2.4.2 - Крок 2 Завдання 3 - поле для виконання завдання

Користувач може помилково ставити символи крім необхідних математичних або цифри. У цьому випадку з'являється повідомлення про помилку. Текст повідомлення: «У даному завданні треба проставити символи «+» або «-». Числа та букву вказувати не треба».

В залежності від правильності відповідей:

1. Всі відповіді правильні – перехід на крок 3.
2. Якщо здійснено хоча б один неправильний вибір – повідомлення про помилку.

Текст повідомлення:

«При виборі допущено помилки. Правильний порядок обчислення це додавання перших трьох переліків елементів і віднімання останніх трьох».

При цьому неправильні відповіді у комірках виділити іншим кольором (наприклад, червоним) та надати можливість користувачу повторно виконати вибір. Перехід на крок 3.

Крок 3. На екрані з'являється приклад матриці і завдання. Текст завдання: «Для наведеної матриці

$$\begin{pmatrix} 2 & 3 & 1 \\ 4 & 2 & 3 \\ 2 & 4 & 1 \end{pmatrix}$$

обчисліть визначник третього порядку

$\Delta =$ ».

Користувач має обчислити та ввести значення визначника третього порядку.

Користувач може помилково ставити текст замість цифри. У цьому випадку

з'являється повідомлення про помилку. Текст повідомлення: «У даному завданні треба вписати число».

В залежності від правильності відповідей:

1. Всі відповіді правильні – завдання завершено.
2. Відповідь неправильна – повідомлення про помилку.

Текст повідомлення:

«Правильне обчислення виконується за формулою: $\Delta = a_{11} * a_{22} * a_{33} + a_{13} * a_{21} * a_{32} + a_{31} * a_{12} * a_{23} - a_{13} * a_{22} * a_{13} - a_{33} * a_{12} * a_{21} - a_{11} * a_{23} * a_{32}$ ».

Після заповнення всіх чарунок правильними відповідями, оскільки це завдання є останнім у тренажері, незалежно від вибору користувача, відбувається заміна деяких елементів тренажеру повідомлення про успішне завершення завдання і залишається тільки кнопка «Закрити завдання». Натискання на цю кнопку повертає користувача на перше вікно з котрого вже можна закрити сам тренажер.

Оцінка знань не проводиться, оскільки програма є тренажером для повторення чи вивчення матеріалу, а не тестом з теми.

3. ПРАКТИЧНА ЧАСТИНА

3.1. Блок схеми алгоритмів

На рис. 3.1. представлена блок-схема алгоритму тренажера в загальному випадку

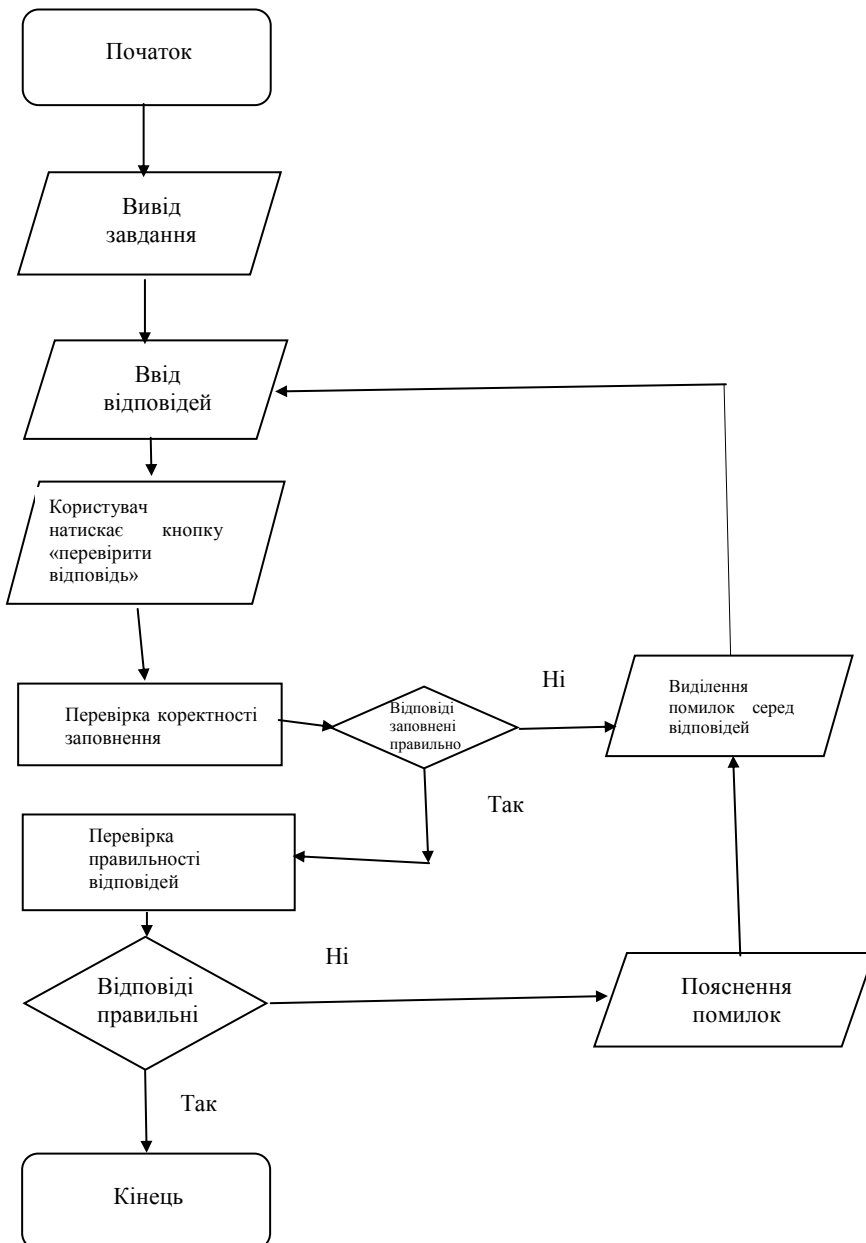
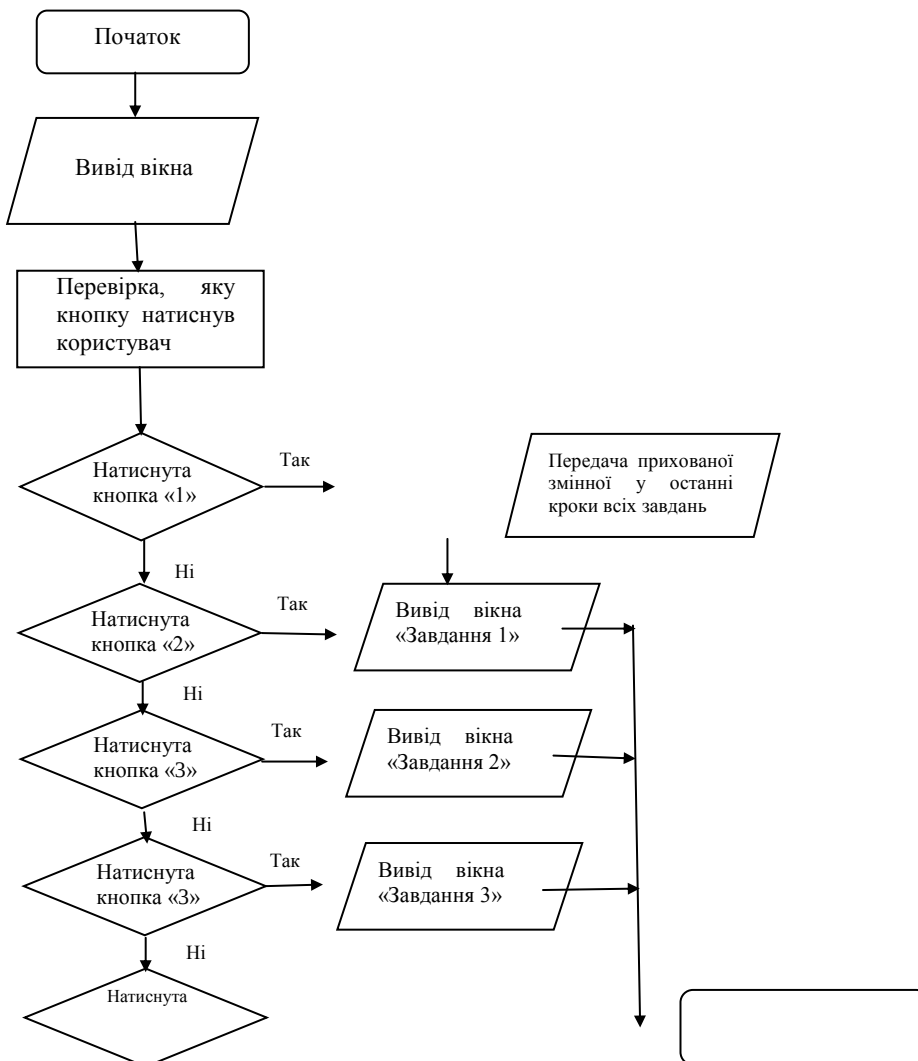


Рис. 3.1 – Блок-схема алгоритму тренажера в загальному

Оскільки деякі з задач потребують від користувача самостійно вписати відповідь, а не обрати одну з доступних, можуть виникнути ситуації, коли механізми Windows Forms бачать відповідь у якості системного винятку. Щоб подібні ситуації не призводили до аварійного завершення роботи програми було реалізовано механізм перевірки відповідей перед тим, як дати програмі їх обробити. Детальніше, можна буде побачити на прикладі «Завдання 1 Крок 3», де подібний механізм задіяно вперше.

3.2 Головне вікно програми

Розглянемо основне вікно програми. Воно має містити в собі, як кнопку для початку всього тесту, так і кнопки, котрі дозволять викликати завдання окремо. Алгоритм роботи цієї сторінки виглядає наступним чином:



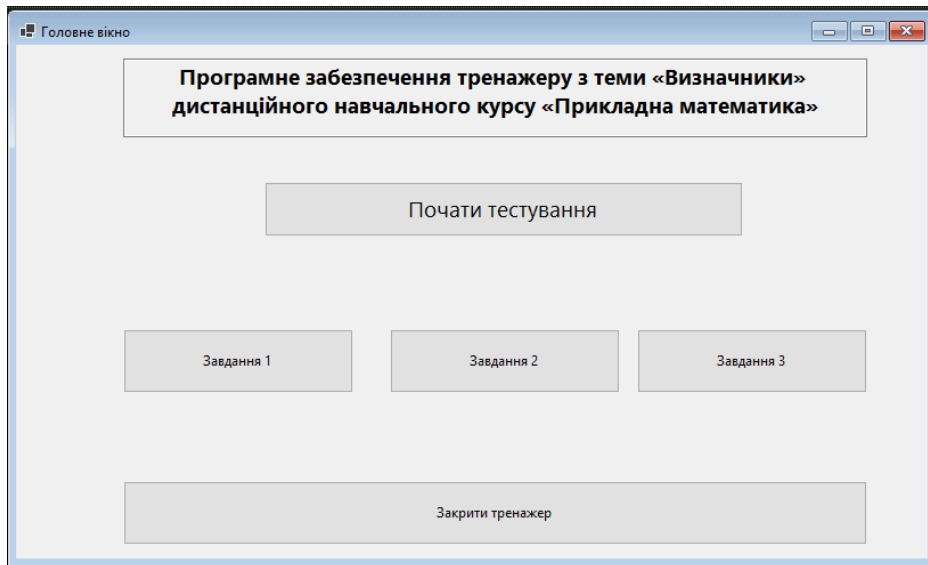


Рис. 3.2.2 – Інтерфейс головного вікна програми

Програмний код для вікна має наступний вигляд:

```
using System.Runtime.CompilerServices;
using System.Security.Cryptography.X509Certificates;

namespace Definers_test
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public static bool a_t; //Змінна для визначення чи запущено весь тест, чи окреме завдання
        private void button1_Click(object sender, EventArgs e)
        {
            var form2 = new Form2();
            form2.Show();
            this.Hide();
            a_t = true;
        }
        private void button2_Click(object sender, EventArgs e)
        {
            var form2 = new Form2();
            form2.Show();
            this.Hide();
            a_t = false;
        }
        private void button3_Click(object sender, EventArgs e)
        {
            var Test2 = new Test2_1();
            Test2.Show();
            this.Hide();
            a_t = false;
        }
        private void button4_Click(object sender, EventArgs e)
        {

```

```

    {
        var Test3 = new Test3_1();
        Test3.Show();
        this.Hide();
        a_t = false;
    }
    private void button5_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}

```

Особливої уваги заслуговує змінна «a_t». Її призначення – дати програмі зрозуміти, що користувач запустив весь тест, а не одне конкретне завдання. Вона має тип «bool» (логічна операція) і приймає значення «true» (твердження правдиве) або «false» (твердження не вірне). Також тип цієї змінної «public» (загальнодоступний), що дозволяє використовувати її в інших частинах програми. Ця змінна використовується у останніх кроках кожного завдання тесту, щоб завдання передало команду на відкриття наступного завдання або на відображення користувачу, що він впорався з завданням.

Метод, котрий дозволяє викликати цю змінну у інших частинах застосунку має наступний вигляд:

```
bool a_t = Form1.a_t;
```

де «Form1» - внутрішня назва головного вікна у самій програмі.

Метод «button1_Click» - це внутрішній метод застосунку побудови Microsoft Forms, котрий ініціалізується у додатковому вікні дизайну і дозволяє викликати певні події після натискання кнопки. Даний метод в подальшому буде служити у якості основного методу перевірки, блокуючи подальший прогрес користувача у використанні застосунку, поки не буде надано правильних відповідей.

3.3 Завдання 1

Є найбільшим завданням тренажеру і містить 5 форм з різними завданнями. Кожна форма має різну реалізацію питань та способів вводу відповіді і робить акцент саме на програмній реалізації тренажеру, подекуди додаючи необов'язкові для закріплення матеріалу питання (див. опитування у 3.3.5).

3.3.1 Завдання 1 Крок 1

Блок-схема цього завдання збігається з основною блок-схемою роботи

програми (рис 3.1).

Завдання 1

Підказка

Дано матрицю

x_{11}	x_{12}	x_{13}
x_{21}	x_{22}	x_{23}
x_{31}	x_{32}	x_{33}

серед вказаних мінорів елементів матриці оберіть той, що відповідає елементу x_{11} . Відмітьте клітинку, котру вважаєте правильною відповіддю.

☐ $M_{11} = \begin{matrix} x_{12} & x_{13} \\ x_{22} & x_{33} \end{matrix}$
☐ $M_{11} = \begin{matrix} x_{11} & x_{12} \\ x_{21} & x_{32} \end{matrix}$

☐ $M_{11} = \begin{matrix} x_{22} & x_{23} \\ x_{32} & x_{33} \end{matrix}$
☐ $M_{11} = \begin{matrix} x_{21} & x_{22} \\ x_{31} & x_{32} \end{matrix}$

Перевірити відповідь

Закрити завдання

Рис. 3.3.1.1 – Інтерфейс вікна, що відповідає «Завдання 1. Крок 1».

В даній формі (вікні застосунку Microsoft Forms) використовується клас «RadioButton», котрий дозволяє відмітити лише одну кнопку одночасно. Це інтегрований у середовище MF клас і його використання автоматично створює всі необхідні компоненти опису інтерфейсу.

Код з логікою вікна має наступний вигляд:

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

namespace Definers_test
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_FormClosed(object sender, FormClosedEventArgs e)
        {
            var form1 = new Form1();
            form1.Show();
        }
    }
}
```

Особливу увагу хочеться приділити використанню конструкції «this.Hide()» закриття сторінок. Проблема полягає у тому, що кожна наступна форма, котра викликається у MF є спадковою від тієї, звідки викликається. Через це її закриття автоматично закриває і наступне завдання. Подія «button_close_Click» - закриває

діючу форму і вона прив'язана до кнопки «закрити завдання». Також вона викликає «Основне вікно» програми, оскільки це прописано у події «Form2_FormClosed» (використовується у кожній формі застосунку, крім першої). Варто відмітити, що проблемою є виклик подією нової копії вікна. Через це у коді попередньої форми і використовується перезапис змінної «a_t», щоб вона не зберігала свій стан, якщо користувач відкриє повний тест, закриває його, а потім відкриває окреме завдання.

Повідомлення про помилки реалізовано за допомогою класу MF «MessageBox», котрий дозволяє заблокувати активну форму, поки не буде натиснута кнопка «Зрозуміло».

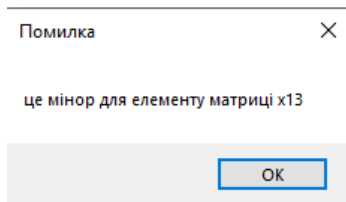


Рис. 3.3.1.2. – Повідомлення про помилку

В свою чергу кнопка «Підказка» реалізована за допомогою нової форми, котра має назву «Form2_help» і викликається не як нова форма з прихованням старої, а за допомогою конструкції:

```
private void Label1_Click(object sender, EventArgs e)
{
    var help12 = new Form2_help();
    help12.ShowDialog();
}
```

Це дозволяє викликати вікно з підказкою у якості додаткового вікна, котре блокує введення якихось даних чи використання батьківського вікна, поки не буде закрито «діалогове вікно».

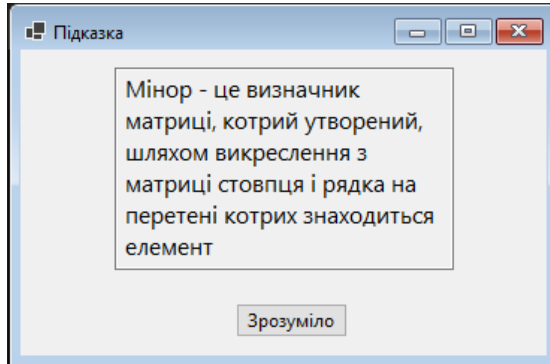


Рис. 3.3.1.3. – Діалогове вікно, що відкривається при натисканні на кнопку «підказка»

Код для цього вікна має наступний вигляд:

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

namespace Definers_test
{
    public partial class Form2_help : Form
    {
        public Form2_help()
        {
            InitializeComponent();
        }
        private void button_close_Click(object sender, EventArgs e)
        {
            this.Close();
            Close();
        }
    }
}
```

При цьому можна помітити відсутність у блоку події закриття форми. Воно однакове для діалогових вікон і повертає основну форму у «фокус» (дозволяє з нею працювати).

3.3.2 Завдання 1 Крок 2

Блок-схема для цього завдання відповідає основній блок-схемі роботи програми (рис.3.1).

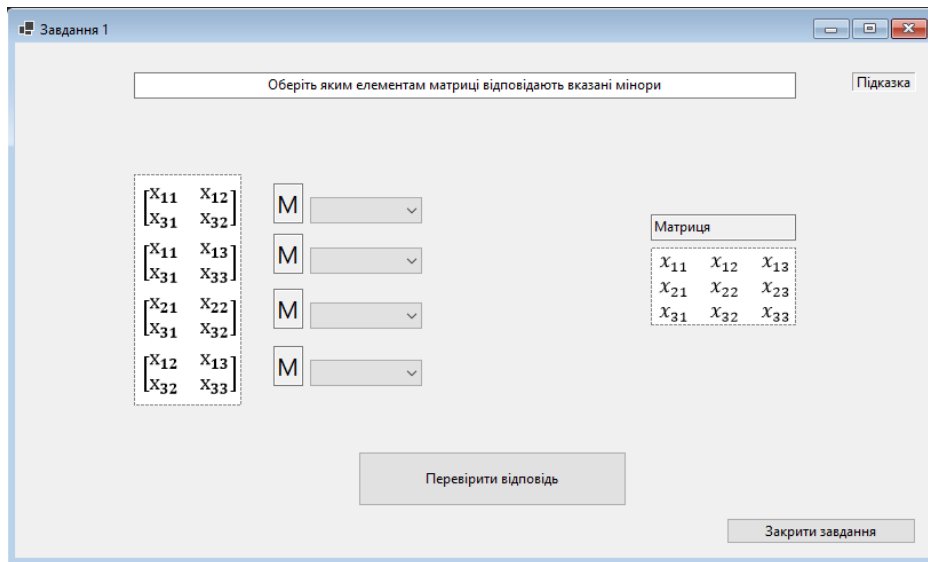


Рис. 3.3.2.1 – Вікно, котре відповідає компоненту «Завдання 1. Крок 2»

Для вводу відповідей у цьому завданні використовується клас «ComboBox», котрий дозволяє задати різні властивості обраному елементу. Конкретно у цьому випадку дане поле створене для вибору значень зі списку наперед визначених значень. Список містить як правильні, так і не правильні значення. Кнопка «підказка» викликає «MessageBox», що значно спрощує як написання коду, так і навантаження від роботи програми. Код має наступний вигляд:

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

namespace Definers_test
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }
    }
}
```



```

    }
    //Закриття форми
    private void Form3_FormClosed(object sender, FormClosedEventArgs e)
    {
        var form1 = new Form1();
        form1.Show();
    }
    //Виклик підказки
    private void Label1_Click(object sender, EventArgs e)
    {
        var help12 = new Form2_help();
        help12.ShowDialog();
    }
    //Закриття форми натисканням кнопки "завершити завдання".
    //Описує саму роботу кнопки, що закріє форму. Виклик першої форми
    //відбувається у відповідній події "форма.Закрита"
    private void button_close_Click(object sender, EventArgs e)
    {
        this.Close();
        Close();
    }
    //Кнопка перевірити відповіді
    private void button_next_Click(object sender, EventArgs e)
    {
        //Якщо користувач нічого не обрав, або не обрав одну з відповідей
        //Програма повідомляє про це відповідною помилкою
        if (comboBox1.SelectedIndex == -1 || comboBox2.SelectedIndex == -1 || comboBox3.SelectedIndex == -1 ||
comboBox4.SelectedIndex == -1)
        {
            string message2 = "Будь ласка, заповніть відповіді";
            string title = "Помилка";
            MessageBox.Show(message2, title);
        }
        else
        {
            //Оскільки значення в випадючих списках є текстом, треба перетворити їх на число для подальшого
            порівняння. Це трошки монструозне рішення, але краще за нестабільне порівняння тексту, що буде використовуватись
            далі

            //Насправді, дана дія є зайвою, через існування такого методу, як String.Compare
            //але для цього завдання було вирішено скористатись нестандартним методом
            //Порівняння тексту буде в інших завданнях
            int cur_answer_1 = Convert.ToInt32(comboBox1.Text);
            int cur_answer_2 = Convert.ToInt32(comboBox2.Text);
            int cur_answer_3 = Convert.ToInt32(comboBox3.Text);
            int cur_answer_4 = Convert.ToInt32(comboBox4.Text);
            //Задасмо значення для порівняння
            int valid_answer_1 = 23;
            int valid_answer_2 = 22;
            int valid_answer_3 = 13;
            int valid_answer_4 = 21;
            //Тепер виконуємо порівняння. Всі значення мають бути правильними одночасно
            //Уникаємо зайвого коду, як це, наприклад відбувається у Test3_2
            if (cur_answer_1 == valid_answer_1
                && cur_answer_2 == valid_answer_2
                && cur_answer_3 == valid_answer_3
                && cur_answer_4 == valid_answer_4)
            {
                //Якщо відповіді надані вірно, програма просто пропускає користувача далі
                {
                    var form2 = new Form4();
                    form2.Show();
                    this.Hide();
                }
            }
            else
            {
                //Якщо у відповідях є помилки, користувач бачить повідомлення, що містить правильні відповіді
            }
        }
    }

```

```

        string message2 = "Для першого елемента правильна відповідь 23. Для другого 22. Для третього 13.  

        Для четвертого 21";
        string title = "Помилка";
        MessageBox.Show(message2, title);
    }
}
}

```

Можна помітити, що повідомлення для виклику підказки і для помилки мають однакову назву змінної. Оскільки вони знаходяться у різних блоках, котрі є «private» (дані використовуються у самому блоці) – ці данні не змішуються між собою. Тобто, при натисканні «підказка» користувач бачить підказку, а при невірній відповіді повідомлення про помилку.

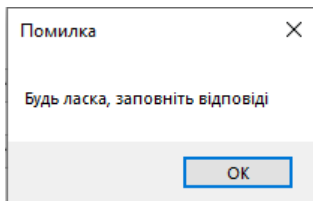


Рис. 3.3.2.2. – Вікно, що повідомляє про відсутні відповіді

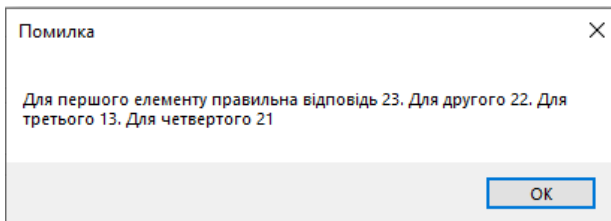


Рис. 3.3.2.3. – Вікно, що виникає, коли відповіді не

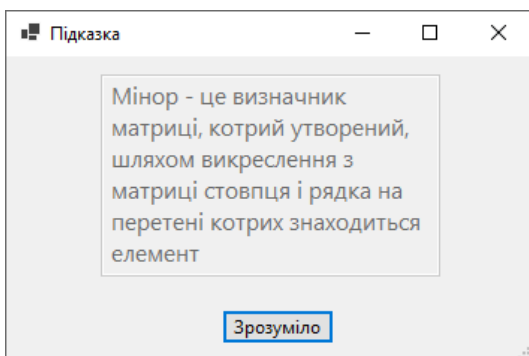


Рис. 3.3.2.3. – Вікно «Підказка»

«Матриця» слугує для наочності і не має жодного використання в рамках програми.

3.3.3 Завдання 1 Крок 3

Блок-схема для цього завдання відповідає основній блок-схемі алгоритму роботи програми. Інтерфейс вікна має наступний вигляд:

Рис. 3.3.3.1. – основне вікно «Завдання 1. Крок 3»

В цьому завданні вперше використовується клас «TextBox», котрий користувач може редагувати. Знаки «?» проставлені завчасно, щоб уникнути перевірки на пусту відповідь. Вікно «Підказка» реалізоване додатковою формою, котра має наступний вигляд:

Рис. 3.3.3.2. – вікно «Підказка» для «Завдання 1. Крок

Першою перевіркою при натисканні на кнопку «Перевірити відповідь» є перевірка на коректність символів у відповіді. Відповідь має містити тільки наперед визначені значення «1», «2» або «3». Повідомлення про помилку у цьому випадку має наступний вигляд:

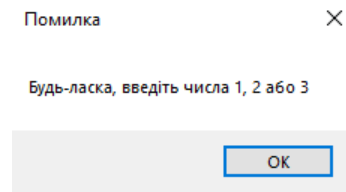


Рис. 3.3.3.3. – вікно «помилка», якщо відповідь користувача не знаходиться у визначеному діапазоні

Підказка, по аналогії з попереднім вікном викликається як діалогове вікно, котре заблоковано для вводу чи копіювання. Вона містить текстовий опис формули.

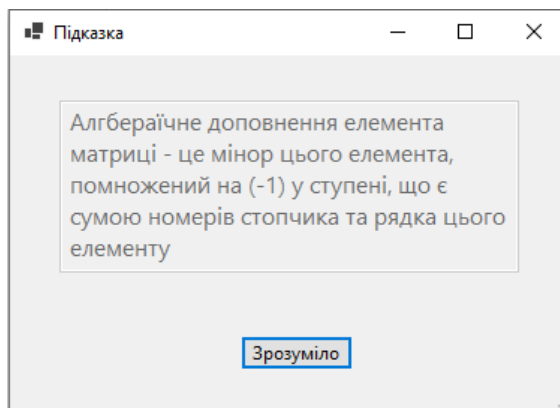


Рис. 3.3.3.3. – вікно «Підказка» для «Завдання 1. Крок 3»

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

namespace Definers_test
{
    public partial class Form4 : Form
    {
        public Form4()
    }
}
```

```

{
    InitializeComponent();
}
//Виклик підказки
private void Label1_Click(object sender, EventArgs e)
{
    var help12 = new Form4_help();
    help12.ShowDialog();
}
//Реакція на закриття форми
private void Form4_FormClosed(object sender, FormClosedEventArgs e)
{
    var form1 = new Form1();
    form1.Show();
}
//реакція на натискання кнопки "закрити завдання"
private void button_close_Click(object sender, EventArgs e)
{
    this.Close();
    Close();
}
//встановлення змінних. Було не обов'язким, але програма його вимагала
public object TextBox5 { get; private set; }
public object TextBox3 { get; private set; }
public object TextBox4 { get; private set; }
//подія при натисканні на кнопку "перевірити відповіді"
private void button_next_Click(object sender, EventArgs e)
{
    //скидаємо кольорову схему, на випадок повторних помилок
    textBox3.BackColor = SystemColors.Window;
    textBox4.BackColor = SystemColors.Window;
    textBox5.BackColor = SystemColors.Window;
    //оголошення змінних. В тому числі і "логічних", щоб виконати порівняння
    //тексту відповіді з завданням.
    string cur_ans_1 = textBox3.Text;
    string cur_ans_2 = textBox4.Text;
    string cur_ans_3 = textBox5.Text;
    bool ans_1_cor = false;
    bool ans_2_cor = false;
    bool ans_3_cor = false;
    //особлива змінна
    bool block_message = false;
    //Створюємо список "коректних значень"
    var cor_inp = new List<string> { "1", "2", "3" };
    //Перевіряємо символи на коректність
    bool comp1 = cor_inp.Contains(cur_ans_1, StringComparer.OrdinalIgnoreCase);
    bool comp2 = cor_inp.Contains(cur_ans_2, StringComparer.OrdinalIgnoreCase);
    bool comp3 = cor_inp.Contains(cur_ans_3, StringComparer.OrdinalIgnoreCase);
    //Якщо всі значення коректні, то йдемо далі. Якщо ні - помилка
    if (comp1 == true && comp2 == true && comp3 == true)
    {
        //виставляємо значення службових змінних, бо завдання містить нюанс у реалізації
        if (cur_ans_1 == "1")
        {
            ans_1_cor = true;
            //Чарунки з вірними відповідями відмічаються зеленим
            textBox3.BackColor = Color.Green;
        }
        else
        {
            ans_1_cor = false;
            //Чарунки з помилками - червоним
            textBox3.BackColor = Color.Red;
        }
        if (cur_ans_2 == "2" || cur_ans_2 == "3")
        {

```

```

        ans_2_cor = true;
        textBox4.BackColor = Color.Green;
    }
    else
    {
        ans_2_cor = false;
        textBox4.BackColor = Color.Red;
    }
    if (cur_ans_3 == "3" || cur_ans_3 == "2")
    {
        if (cur_ans_2 == cur_ans_3)
        {
            //Добуток дозволяє перестановку множників, але
            //якщо проігнорувати можливу їх рівність, завдання буде зараховано
            //з неправильною відповіддю. Задана перевірка блокує таку можливість
            ans_3_cor = false;
            block_message = true;
            textBox5.BackColor = Color.Red;
            string message1 = "Хоча добуток дозволяє перестановку елементів, це не означає, що вони мають
мати однаковий порядковий номер у цьому завданні";
            string title = "Помилка";
            MessageBox.Show(message1, title);
        }
        else
        {
            ans_3_cor = true;
            textBox5.BackColor = Color.Green;
        }
    }
    else
    {
        ans_3_cor = false;
        textBox5.BackColor = Color.Red;
    }
    if (ans_1_cor == true && ans_2_cor == true && ans_3_cor == true)
    {
        var form3 = new Form5();
        form3.Show();
        this.Hide();
    }
    else
    {
        //якщо не додати цю пусту перевірку, програма створюватиме два діалогових вікна поспіль
        if (block_message == true)
        {
        }
        else
        {
            string message1 = "Формула алгебраїчного доповнення означає, що алгебраїчне доповнення
дорівнює результату множення мінора відповідного елемента на (-1) в ступіні (i-j)";
            string title = "Помилка";
            MessageBox.Show(message1, title);
        }
    }
}
//Якщо користувач ввів значення не з визначених, як дозволені
else
{
    string message2 = "Будь-ласка, введіть числа 1, 2 або 3";
    string title = "Помилка";
    MessageBox.Show(message2, title);
}
}
}

```

Особливістю цього завдання є використання кольорових схем для вказання помилок, котрих допустився користувач. Приклад на зображенні «3.1.4.4». Це реалізовано за допомогою конструкції «клас.Властивість». У конкретному випадку, це «textBox1.BackColor», що дозволяє встановити фоновий колір для об'єкту.

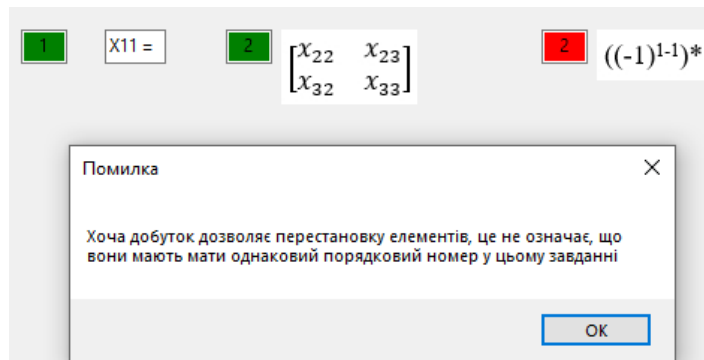


Рис. 3.3.3.4. – основне вікно завдання з відміченими полями де допущені помилки та підказкою, про те, що відповідь 2 та 3 не можуть бути однаковими

3.3.4 Завдання 1 Крок 4

Блок-схема для роботи цього компонента програми виглядає наступним чином:

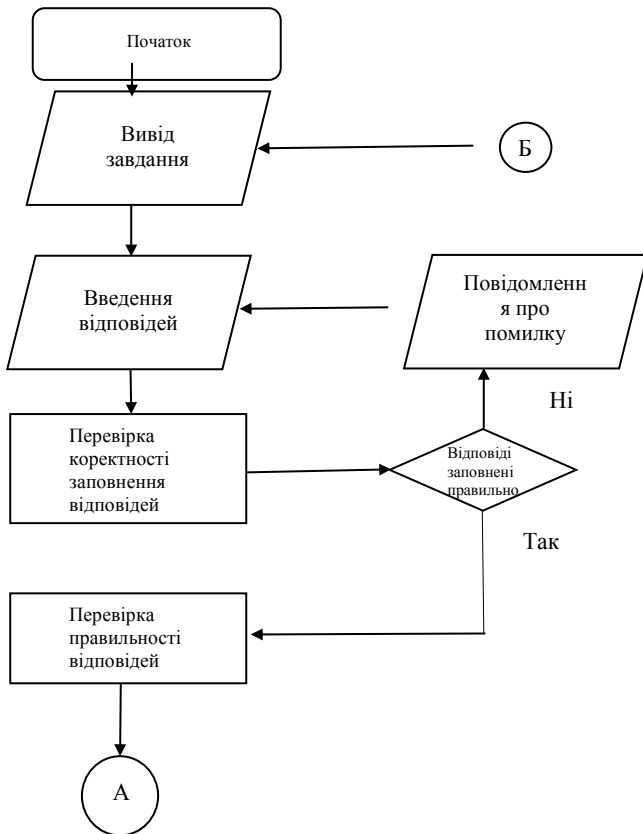


Рис. 3.3.4.1 – Блок-схема алгоритму для «Завдання 1 Крок 4»

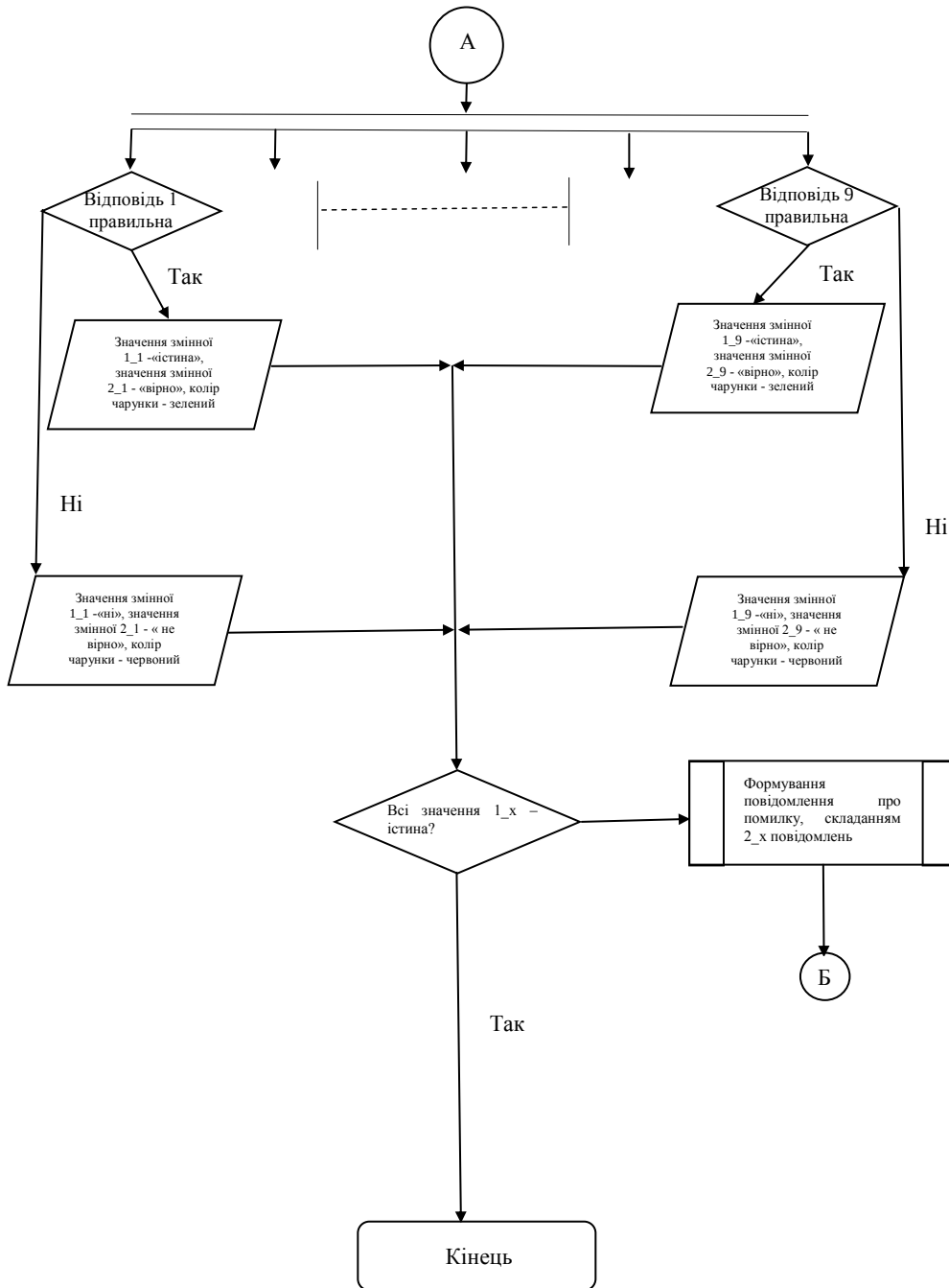


Рис. 3.3.4.2 – Блок-схема алгоритму для «Завдання 1 Крок 4» (кінець)

Вікно для цього завдання має наступний вигляд:

Дано наступну матрицю

$$\begin{bmatrix} 4 & 7 \\ 1 & 3 \end{bmatrix}$$

Підказка

заповніть таблицю для обчислення алгебраїчного доповнення наступних елементів: x_{11} , x_{22} та x_{12}

Елемент	$(-1)^{i+j}$	Міnor елемента	Алгебраїчне доповнення
X_{11}	?	?	?
X_{22}	?	?	?
X_{12}	?	?	?

Перевірити відповідь

Закрити завдання

Рис. 3.3.4.3 – Основне вікно «Завдання 1 Крок 4»

Кнопка «підказка» викликає нову форму, що являє собою діалогове вікно та має вигляд:

Перший стовбець таблиці треба заповнити значеннями "1" та "-1" відповідно до ступені, що є сумою індексів для елемента, котрий вказаний у рядку. Так, для елемента X_{21} це буде (-1) у ступені $(2+1)$, що дає "-1"

Другий стовбець треба заповнити значеннями мінорів, що відповідають елементам. Для цього треба прибрати строку і стовбець, що відповідають індексу елемента. Так, для елемента X_{21} треба викреслити з матриці другий рядок та перший стовбець. Залишиться значення 7.

Третій стовбець треба заповнити значеннями алгебраїчного доповнення. Воно дорівнює добутку "-1" у ступені, що є сумою індексів елемента та мінора цього елемента. Для елемента x_{21} це буде $(-1)^{(1+2)*7}$, що дорівнює -7

Зрозуміло

Рис. 3.3.4.4 – Вікно «Підказка» для «Завдання 1 Крок 4»

Візуальна складність блок-схеми викликана частими повтореннями однакових дій, що фактично складають послідовність порівнянь відповідей спочатку з діапазоном допустимих, а потім з коректними відповідями. Якщо

детально розглянути окремий цикл роботи, то для чарунки 2 діють наступні перевірки:

1. Перевірка чи всі поля заповнені допустимими значенням. Якщо – ні, то повідомлення про помилку. Якщо так – алгоритм працює далі.
2. Перевірка правильності значення у чарунці. Якщо значення правильне – змінна «answer_2_cog» набуває значення «істина», а чарунки зафарбовується зеленим кольором. Якщо ні – змінна набуває значення «не вірно», а чарунка зафарбовується зеленим.
3. Далі програма порівнює чи всі чарунки повертають значення «істина». Якщо це так – програма завершує завдання. Якщо, це не так – виводить повідомлення про помилку.

Код має наступний вигляд:

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

namespace Definers_test
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();
        }
        private void Form5_FormClosed(object sender, FormClosedEventArgs e)
        {
            var form1 = new Form1();
            form1.Show();
        }
        private void button_close_Click(object sender, EventArgs e)
        {
            this.Close();
            Close();
        }
        private void Label1_Click(object sender, EventArgs e)
        {
            var help12 = new Form5_help();
            help12.ShowDialog();
        }
        private void button_next_Click(object sender, EventArgs e)
        {
            string cur_ans_1 = textBox3.Text;
            string cor_ans_1 = "1"; //x11 (-1)^ij
            string cur_ans_2 = textBox4.Text;
            string cor_ans_2 = "1"; //x22 (-1)^ij
        }
    }
}
```

```

string cur_ans_3 = textBox5.Text;
string cor_ans_3 = "-1"; //x12 (-1)^ij
string cur_ans_4 = textBox6.Text;
string cor_ans_4 = "3"; //x11 minor
string cur_ans_5 = textBox7.Text;
string cor_ans_5 = "4"; //x22 minor
string cur_ans_6 = textBox8.Text;
string cor_ans_6 = "1"; //x12 minor
string cur_ans_7 = textBox9.Text;
string cor_ans_7 = "3"; //x11 minor
string cur_ans_8 = textBox10.Text;
string cor_ans_8 = "4"; //x22 minor
string cur_ans_9 = textBox11.Text;
string cor_ans_9 = "-1"; //x12 minor
var cor_in = new List<string> { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "-1", "-2", "-3", "-4", "-5", "-6", "-7", "-8", "-9" };
bool compar1 = cor_in.Contains(cur_ans_1);
bool compar2 = cor_in.Contains(cur_ans_2);
bool compar3 = cor_in.Contains(cur_ans_3);
bool compar4 = cor_in.Contains(cur_ans_4);
bool compar5 = cor_in.Contains(cur_ans_5);
bool compar6 = cor_in.Contains(cur_ans_6);
bool compar7 = cor_in.Contains(cur_ans_7);
bool compar8 = cor_in.Contains(cur_ans_8);
bool compar9 = cor_in.Contains(cur_ans_9);

if (compar1 && compar2 && compar3 && compar4 && compar5 && compar6
    && compar7 && compar8 && compar9)
{ //зміни, щоб сформувані повідомлення про помилку
  string message1;
  string message2;
  string message3;
  string message4;
  string message5;
  string message6;
  string message7;
  string message8;
  string message9;

  //спочатку (вище) ми перевірили чи знаходяться значення в допустимому діапазоні
  //це дозволяє продовжити роботу без зайвих крашів програми
  //TextBox3
  bool answer_1_cor = (cur_ans_1 == cor_ans_1);
  if (answer_1_cor == true)
  {
    message1 = "Чарунка 1: Відповідь вірна";
    textBox3.BackColor = Color.Green;
  }
  else
  {
    message1 = "Чарунка 1! Ступінь «i-j» для елемента x11 обчислюється як (1-1)";
    textBox3.BackColor = Color.Red;
  }
  //TextBox4
  bool answer_2_cor = (cur_ans_2 == cor_ans_2);
  if (answer_2_cor == true)
  {
    message2 = "Чарунка 2: Відповідь вірна";
    textBox4.BackColor = Color.Green;
  }
  else
  {
    message2 = "Чарунка 2! Ступінь «i-j» для елемента x11 обчислюється як (2-2)";
    textBox4.BackColor = Color.Red;
  }
}

```

```

//TextBox5
bool answer_3_cor = (cur_ans_3 == cor_ans_3);
if (answer_3_cor == true)
{
    message3 = "Чарунка 3: Відповідь вірна";
    textBox5.BackColor = Color.Green;
}
else
{
    message3 = "Чарунка 3! Ступінь «i-j» для елемента x22 обчислюється як (1-2). Від'ємна ступінь
вказує, що треба виконати ділення 1 на число у вказаному ступені";
    textBox5.BackColor = Color.Red;
}
//TextBox6
bool answer_4_cor = (cur_ans_4 == cor_ans_4);
if (answer_4_cor == true)
{
    message4 = "Чарунка 4: Відповідь вірна";
    textBox6.BackColor = Color.Green;
}
else
{
    message4 = "Чарунка 4! Мінор елемента дорівнює значенню матриці, котра не включає в себе строку
та стовпчик, що дорівнюють позиції елемента. Тобто треба прибрати стовпчик 1 і строку 1. Залишається значення «3»,
елемент x22 початкової матриці";
    textBox6.BackColor = Color.Red;
}
//TextBox7
bool answer_5_cor = (cur_ans_5 == cor_ans_5);
if (answer_5_cor == true)
{
    message5 = "Чарунка 5: Відповідь вірна";
    textBox7.BackColor = Color.Green;
}
else
{
    message5 = "Чарунка 5! Мінор елемента дорівнює значенню матриці, котра не включає в себе строку
та стовпчик, що дорівнюють позиції елемента. Тобто треба прибрати стовпчик 2 і строку 2. Залишається значення «4»,
елемент x11 початкової матриці";
    textBox7.BackColor = Color.Red;
}
//TextBox8
bool answer_6_cor = (cur_ans_6 == cor_ans_6);
if (answer_6_cor == true)
{
    message6 = "Чарунка 6: Відповідь вірна";
    textBox8.BackColor = Color.Green;
}
else
{
    message6 = "Чарунка 6! Мінор елемента дорівнює значенню матриці, котра не включає в себе строку
та стовпчик, що дорівнюють позиції елемента. Тобто треба прибрати стовпчик 2 і строку 1. Залишається значення «1»,
елемент x21 початкової матриці";
    textBox8.BackColor = Color.Red;
}
//textBox9
bool answer_7_cor = (cur_ans_7 == cor_ans_7);
if (answer_7_cor == true)
{
    message7 = "Чарунка 7: Відповідь вірна";
    textBox9.BackColor = Color.Green;
}
else
{
    message7 = "Чарунка 7! Помилка. Алгебраїчне доповнення є добутком від множників у відповідних
елементу стовпця (-1)i-j та мінор елемента. Тобто 1*3 = 3";

```

```

        textBox9.BackColor = Color.Red;
    }
    //textBox10
    bool answer_8_cor = (cur_ans_8 == cor_ans_8);
    if (answer_8_cor == true)
    {
        message8 = "Чарунка 8: Відповідь вірна";
        textBox10.BackColor = Color.Green;
    }
    else
    {
        message8 = "Чарунка 8! Алгебраїчне доповнення є добутком від множників у відповідних елементу
стовпцях (-1)i-j та мінор елемента. Тобто 1*4 = 4";
        textBox10.BackColor = Color.Red;
    }
    //textBox11
    bool answer_9_cor = (cur_ans_9 == cor_ans_9);
    if (answer_9_cor == true)
    {
        message9 = "Чарунка 9: Відповідь вірна";
        textBox11.BackColor = Color.Green;
    }
    else
    {
        message9 = "Чарунка 9! Алгебраїчне доповнення є добутком від множників у відповідних елементу
стовпцях (-1)i-j та мінор елемента. Тобто -1*1 = -1";
        textBox11.BackColor = Color.Red;
    }

    if (answer_1_cor == true && answer_2_cor == true && answer_3_cor == true &&
        answer_4_cor == true && answer_5_cor == true && answer_6_cor == true &&
        answer_7_cor == true && answer_8_cor == true && answer_9_cor == true)
    {
        var form_new = new Form6();
        form_new.Show();
        this.Hide();
    }
    else
    {
        string title2 = "Помилка";
        MessageBox.Show("Чарунки, відмічені зеленим містять правильні відповіді. Чарунки відмічені
червоним - містять помилки" + "\n" + message1 + "\n" + message2 + "\n" + message3 + "\n" + message4 + "\n" + message5 +
"\n" + message6 + "\n" + message7 + "\n" + message8 + "\n" + message9, title2);
    }
}
else
{
    string message = "Допустими в цих полях є значення від -9 до 9, включно з нулем";
    MessageBox.Show(message);
}
}
}
}

```

Особливістю реалізації цього завдання є складене повідомлення про допущені помилки. Кожна перевірка правильності значення встановлює окрему текстову змінну, що залежно від правильності відповіді має різний текст. Потім всі ці повідомлення складаються в одне ціле. На практиці це має наступний вигляд:

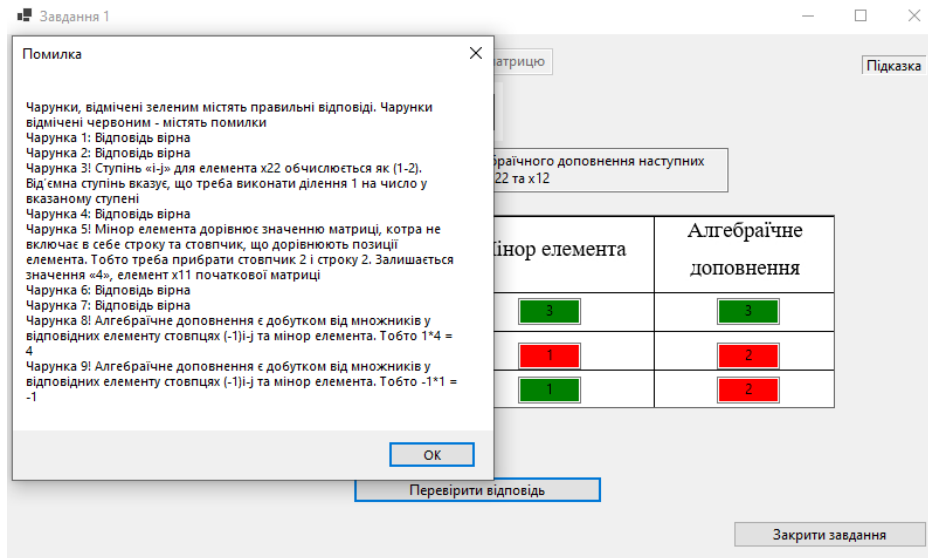


Рис. 3.3.4.5 – Вікно «Помилка» для «Завдання 1 Крок 4»

3.3.5 Завдання 1 Крок 5

Блок-схема для цього завдання має відмінний вигляд від основної, тому що дане вікно використовує змінну «a_t», що була створена на головному вікні програми:

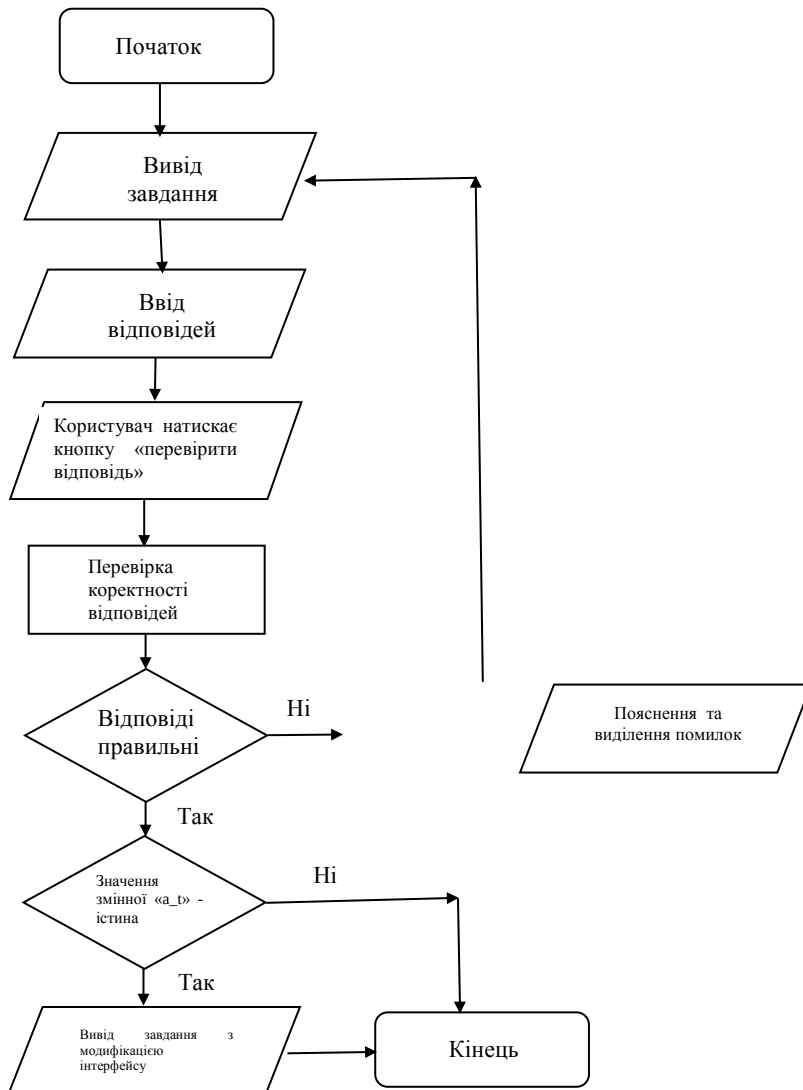


Рис. 3.3.5.1 – Блок-схема для «Завдання 1 Крок 5»

Основний інтерфейс вікна має наступний вигляд:

Завдання 1

Оберіть «Так» або «Ні» під кожним з наступних тверджень

Підказка

Мінором M_{ij} елемента A_{ij} називається визначник $(n-1)$ -го порядку отриманий з визначника n -го порядку викресленням i -го рядка і j -го стовпчика

☐ Так ☐ Ні

Для елемента A_{12} у матриці $\begin{bmatrix} 12 & 6 \\ 3 & 9 \end{bmatrix}$ мінором, або визначником першого порядку є «6».

☐ Так ☐ Ні

Алгебраїчне доповнення A_{ij} для елемента a_{ij} визначається за формулою $A_{ij} = (-1)^{i-j} * M_{ij}$

☐ Так ☐ Ні

Для елемента A_{22} у матриці $\begin{bmatrix} 12 & 6 \\ 3 & 9 \end{bmatrix}$ алгебраїчне доповнення дорівнює «12».

☐ Так ☐ Ні

Перевірити відповідь

Закрити завдання

Рис. 3.3.5.2 – Інтерфейс вікна для «Завдання 1 Крок 5»

Проте, якщо користувач обрав виконання тільки завдання «1» і надав правильні відповіді, вікно матиме наступний вигляд:

Завдання 1

Відповідь правильна

Підказка

Мінором M_{ij} елемента A_{ij} називається визначник $(n-1)$ -го порядку отриманий з визначника n -го порядку викресленням i -го рядка і j -го стовпчика

☒ Так ☐ Ні

Для елемента A_{12} у матриці $\begin{bmatrix} 12 & 6 \\ 3 & 9 \end{bmatrix}$ мінором, або визначником першого порядку є «6».

☐ Так ☒ Ні

Алгебраїчне доповнення A_{ij} для елемента a_{ij} визначається за формулою $A_{ij} = (-1)^{i-j} * M_{ij}$

☒ Так ☐ Ні

Для елемента A_{22} у матриці $\begin{bmatrix} 12 & 6 \\ 3 & 9 \end{bmatrix}$ алгебраїчне доповнення дорівнює «12».

☒ Так ☐ Ні

Закрити Завдання

Рис. 3.3.5.3 – Інтерфейс вікна для «Завдання 1 Крок 5» при «a_t» не вірно

Код програми має наступний вигляд:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Definers_test
{
    public partial class Form6 : Form
    {
        public Form6()
        {
            InitializeComponent();
        }
        private void Form6_FormClosed(object sender, FormClosedEventArgs e)
        {
            var form1 = new Form1();
            form1.Show();
        }
        private void button_close_Click(object sender, EventArgs e)
        {
            this.Close();
            Close();
        }
        private void button_1_Click(object sender, EventArgs e)
        {
            this.Close();
            Close();
        }
        private void Label1_Click(object sender, EventArgs e)
        {
            var help12 = new Form6_help();
            help12.ShowDialog();
        }
        private void button_next_Click(object sender, EventArgs e)
        {
            bool answer_1 = false;
            bool answer_2 = false;
            bool answer_3 = false;
            bool answer_4 = false;
            string message1;
            string message2;
            string message3;
            string message4;
            //змінна з основного вікна програми
            bool a_t = Form1.a_t;
            //скидання кольорів, щоб уникнути зайвого розфарбування при повторних відповідях
            radioButton1.BackColor = Color.White;
            radioButton9.BackColor = Color.White;
            radioButton8.BackColor = Color.White;
            radioButton10.BackColor = Color.White;
            radioButton4.BackColor = Color.White;
            radioButton3.BackColor = Color.White;
            radioButton5.BackColor = Color.White;
            radioButton6.BackColor = Color.White;
            //перевіряємо чи надана хоча б одна відповідь
            if ((radioButton1.Checked || radioButton9.Checked) && (radioButton8.Checked
                || radioButton10.Checked) && (radioButton4.Checked || radioButton3.Checked)
                && (radioButton5.Checked || radioButton6.Checked))
```

```

{
    //GroupBox1 - це перший блок, що ізолює 2 RadioButton в одну групу
    if (radioButton1.Checked)
    {
        answer_1 = true;
        radioButton1.BackColor = Color.Green;
        message1 = "Відповідь відносно першого твердження вірна";
    }
    else
    {
        answer_1 = false;
        radioButton9.BackColor = Color.Red;
        message1 = "Мінором Мij елемента aij називається визначник  $\begin{vmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{i1} & \dots & a_{in} \end{vmatrix}$ -го порядку отриманий з
визначника n -го порядку викресленням i -го рядка і j -го стовпчика. Вказане твердження є вірним";
    }

    //GroupBox2 - те саме, що і GroupBox1
    if (radioButton8.Checked)
    {
        answer_2 = true;
        radioButton8.BackColor = Color.Green;
        message2 = "Відповідь відносно другого твердження вірна";
    }
    else
    {
        answer_2 = false;
        radioButton10.BackColor = Color.Red;
        message2 = "Мінором Мij елемента aij називається визначник  $\begin{vmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{i1} & \dots & a_{in} \end{vmatrix}$ -го порядку отриманий з
визначника n -го порядку викресленням i -го рядка і j -го стовпчика. Значення б відповідає мінору елемента a21. Мінором
для елемента a12 є 9";
    }

    //GroupBox3 - те саме, що і GroupBox1
    if (radioButton4.Checked)
    {
        answer_3 = true;
        radioButton4.BackColor = Color.Green;
        message3 = "Відповідь відносно третього твердження вірна";
    }
    else
    {
        answer_3 = false;
        radioButton3.BackColor = Color.Red;
        message3 = "Вказана формула :  $A_{ij}=(-1)^{(i-j)} * M_{ij}$  є правильною";
    }

    //GroupBox4 - те саме, що і GroupBox1
    if (radioButton5.Checked)
    {
        answer_4 = true;
        radioButton5.BackColor = Color.Green;
        message4 = "Відповідь відносно четвертого твердження вірна";
    }
    else
    {
        answer_4 = false;
        radioButton6.BackColor = Color.Red;
        message4 = "Алгебраїчне доповнення для елемента a22 це добуток  $(-1)^{(2-2)} = (-1)^0 = 1$  та мінора
елемента a22, що дорівнює 12. Наведене твердження було вірним";
    }
    if (answer_1 == true && answer_2 == true && answer_3 == true && answer_4 == true)
    {
        if (a_t == true)
        {
            var form = new Test2_1();

```

```

        form.Show();
        this.Hide();
    }
    else
    {
        button_next.Visible = false;
        button_close.Visible = false;
        button1.Visible = true;
        button2.Visible = true;
        radioButton1.BackColor = Color.White;
        radioButton9.BackColor = Color.White;
        radioButton8.BackColor = Color.White;
        radioButton10.BackColor = Color.White;
        radioButton4.BackColor = Color.White;
        radioButton3.BackColor = Color.White;
        radioButton5.BackColor = Color.White;
        radioButton6.BackColor = Color.White;

    }
}
else
{
    string title = "Помилка";
    MessageBox.Show("При заповненні допущені наступні помилки:" + "\n" + message1 + "\n" + message2
        + "\n" + message3 + "\n" + message4, title);
}
}
//коли відповіді не надані
else
{
    string title = "Помилка";
    string message = "Зробіть вибір!";
    MessageBox.Show(message, title);
}
}
}
}

```

Головною особливістю цього завдання є об'єднання окремих `RadioBox` у групи (`GroupBox`). Ці групи не використовуються у коді і служать виключно для ізоляції класу `RadioBox` від самого себе. Якщо не виконати подібне об'єднання, програма вважатиме, що всі кнопки належать до однієї групи і при виборі однієї будуть блокуватись інші.

Повідомлення про помилку реалізовано аналогічно з попереднім пунктом і має наступний вигляд:

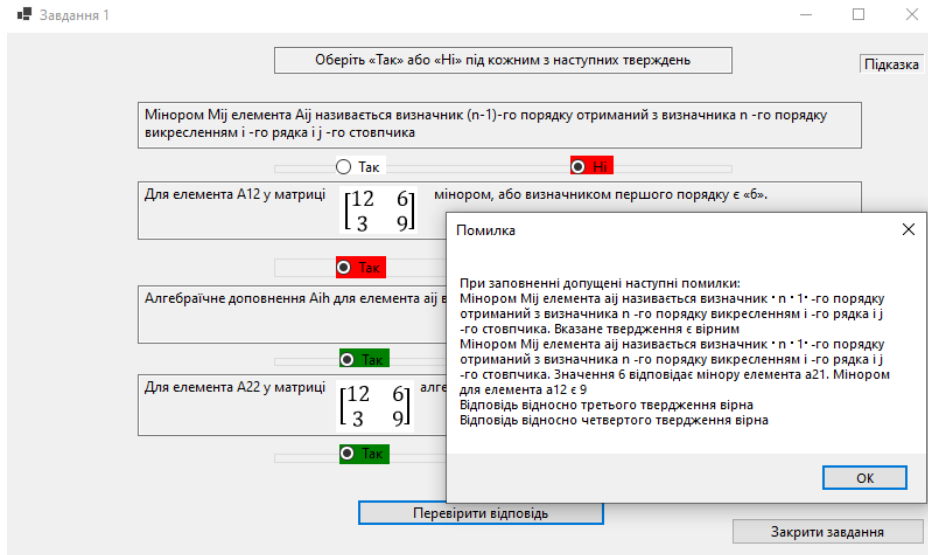


Рис. 3.3.5.4 – Повідомлення про помилку та виділення кольором відповідей

Функція підказка у цьому завданні недоступна, тому містить повідомлення, що «Підказка» не доступна. Реалізовано це новою діалоговою формою.

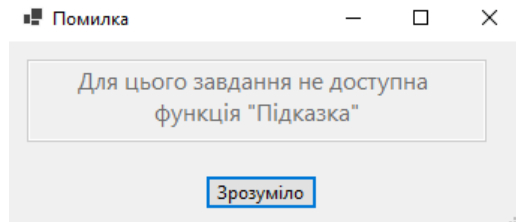


Рис. 3.3.5.5 – текст «Підказки»

3.4 Завдання 2

Хоча і відрізняється за математичними принципами від Завдання 1, але має аналогічні програмні реалізації, що майже не мають відмінностей. Для повідомлень про помилки та для підказок використовується клас `messagebox`, оскільки його використання є простішим за створення діалогових форм.

3.4.1 Завдання 2 Крок 1

Блок-схема цього завдання збігається з основною блок-схемою роботи програми. Інтерфейс першого вікна має наступний вигляд:

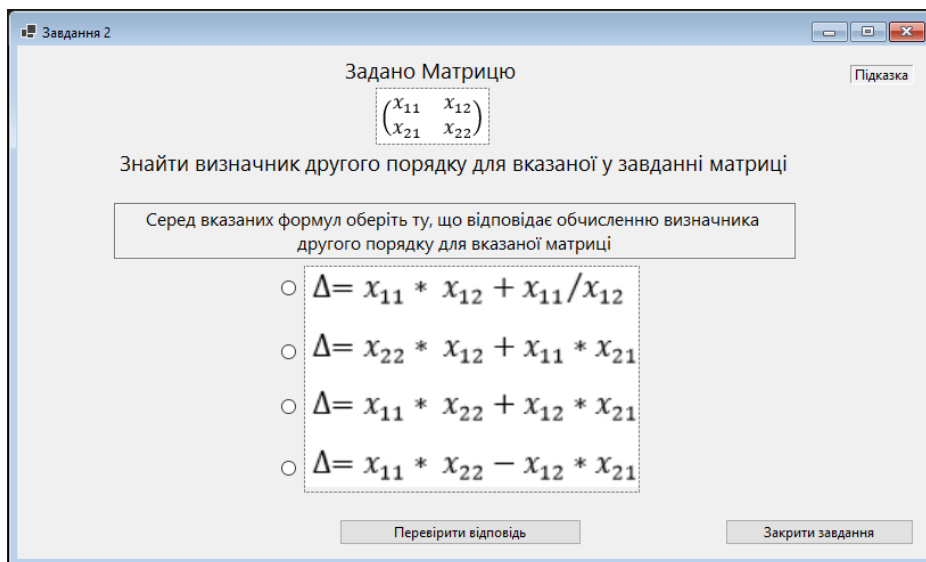


Рис. 3.4.1.1 – Інтерфейс вікна для «Завдання 2 Крок 1»

Тут, по аналогії з «Завдання 1 Крок 1» використовується клас «`RadioButton`», котрий дозволяє обрати лише одну відповідь з запропонованих. Так само використовується зміна кольору елементів, щоб відмітити помилки.

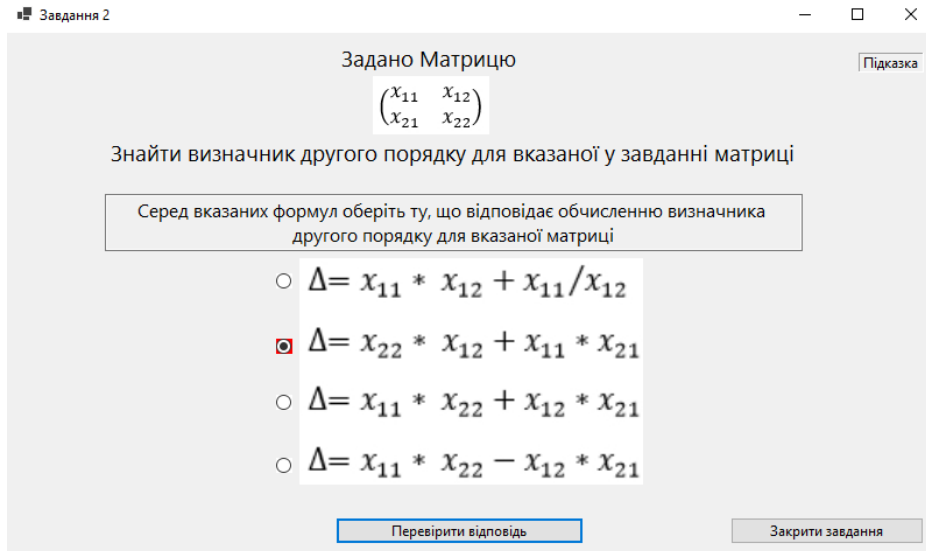


Рис. 3.4.1.2 – Відмічена помилка у відповіді

Код форми наведений у додатку А.

Підказку реалізовано за допомогою класу `messageBox`, що створює діалогове вікно:

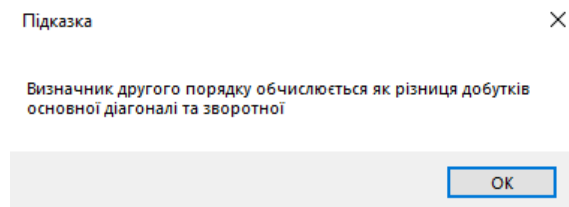


Рис. 3.4.1.3 – Вікно «Підказка»

Помилка реалізована таким самим чином:

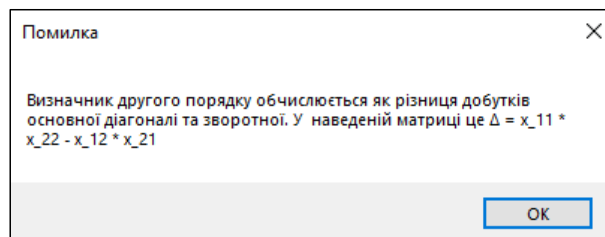


Рис. 3.4.1.3 – Вікно «Помилка»

3.4.2 Завдання 2 Крок 2

Дане завдання реалізоване на основі тих же принципів, що і «Завдання 1 Крок 2». Блок схема є аналогічною. Різницю складає використання класу MessageBox для виведення «Підказка» та «Помилка». Інтерфейс основного вікна містить комбіновані контейнери (comboBox), що дозволяють створювати випадаючі списки.

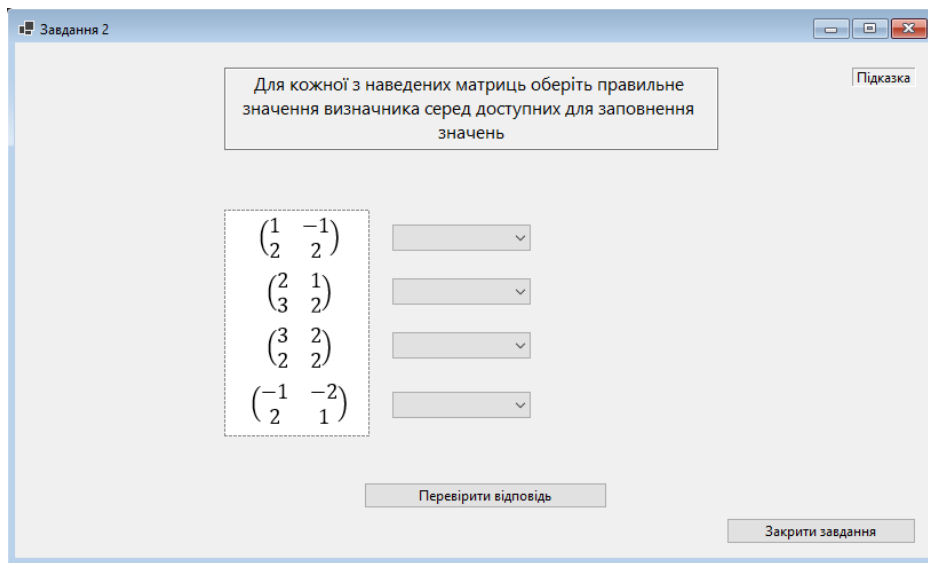


Рис. 3.4.2.1 – Вікно застосунку для «Завдання 2 Крок 2»

Якщо, користувач випадково чи свідомо натискає на кнопку «перевірити відповіді» не вказавши жодної відповіді, з'являється наступне повідомлення:

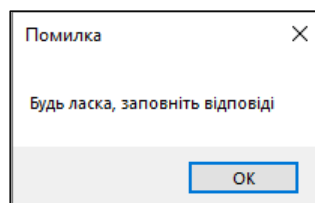


Рис. 3.4.2.2 – Помилка, якщо користувач не надав відповіді

Якщо користувач надає неправильну відповідь, то застосунок виводить повідомлення з правильною відповіддю. Оскільки в цьому завданні застосовується послідовне порівняння значень – змусити застосунок прийняти неправильну

відповідь не вдається. Вікно з помилкою містить правильну відповідь і має наступний вигляд:

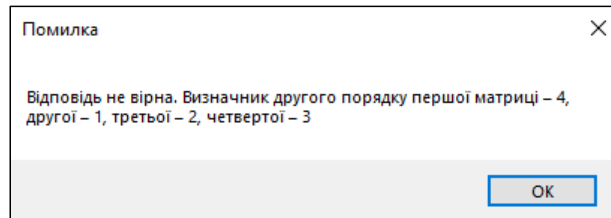


Рис. 3.4.2.3 – Вікно «Помилка»

Дане завдання не заявляло виділення невірних відповідей кольором, тому цього не відбувається. Підказка до цього завдання не є вичерпною і містить тільки основну інформацію, бо у разі неправильної відповіді користувач побачить правильну.

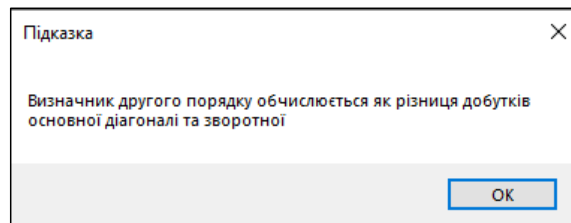


Рис. 3.4.2.4 – Вікно «Підказка»

Введення значень з клавіатури для цього завдання заблокована. Користувач має обирати відповіді за допомогою випадających списків. Це також робить перевірку коректності заповнення зайвою. Код цієї форми має наступний вигляд:

У цьому завданні використовується метод «Convert.ToInt32», котрий слугує для перетворення і подальшого порівняння тексту з дійсними числами. Використання цього методу не є обов'язковим, тому при подальшій оптимізації програми можна замінити його на порівняння текстових значень між собою, що зменшить об'єм програми на кілька строк.

3.4.3 Завдання 2 Крок 3

Дане завдання аналогічно посилається до змінної «a_t», заявленої у основному вікні тренажеру. Його блок-схема аналогічна Завданню 1 Крок 5 з тією різницею, що використовується клас checkbox, а не клас radiobutton. Це дозволяє зробити кілька виборів у рамках одного вікна не блокуючи жодне з полів.

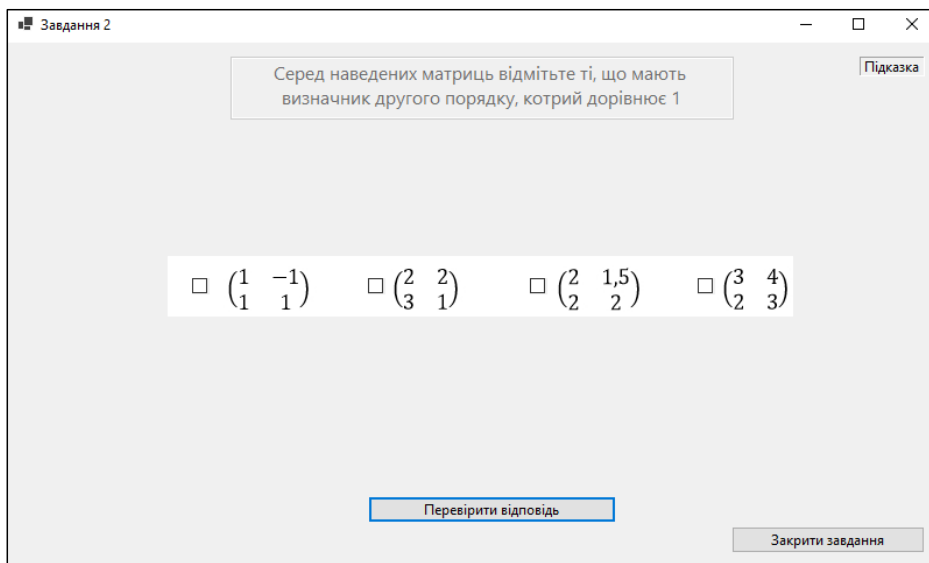


Рис. 3.4.3.1 – Вікно застосунку для «Завдання 2 Крок 3»

Кнопка «Підказка» викликає messagebox, котрий має наступний вигляд:

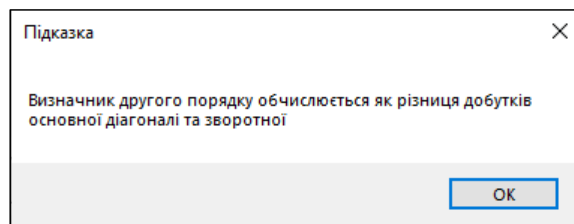


Рис. 3.4.3.2 – Вікно «Підказка»

При відсутності відповіді викликається додаткове вікно, що вказує на те, що відповідь відсутня:

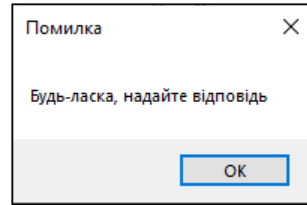


Рис. 3.4.3.3 – «Помилка» відповідь відсутня

Якщо відповідь надана не вірно відбувається обведення червоним кольором невірних відповідей і зеленим вірних, та додається повідомлення про допущену помилку:

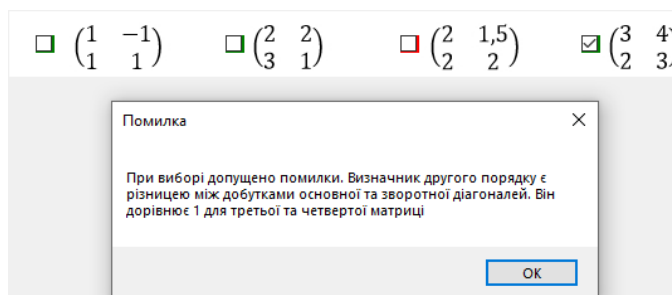


Рис. 3.4.3.4 – Фрагмент вікна, що містить виділення правильних та неправильних відповідей та повідомлення про помилку

Якщо всі відповіді вірні перевіряється значення змінної «a_t». Якщо воно є істиною відбувається модифікація вікна, по аналогії з Завдання 1 Крок 5.

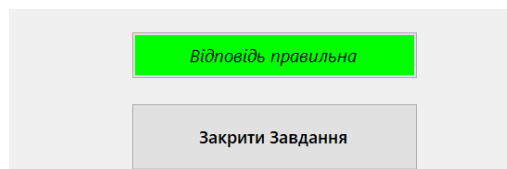


Рис. 3.4.3.4 – Фрагмент вікна, якщо змінна «a_t» має значення «не

В іншому випадку відбувається перехід до наступного завдання.

Код цієї форми має наступний вигляд:

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

namespace Definers_test
{
```

```

public partial class Test2_3 : Form
{
    public Test2_3()
    {
        InitializeComponent();
    }
    private void Test2_3_FormClosed(object sender, FormClosedEventArgs e)
    {
        var form1 = new Form1();
        form1.Show();
    }
    private void button_1_Click(object sender, EventArgs e)
    {
        this.Close();
        Close();
    }
    private void button_next_Click(object sender, EventArgs e)
    {
        bool answer_1;
        bool answer_2;
        bool answer_3;
        bool answer_4;
        bool a_t = Form1.a_t;
        if (checkBox1.Checked || checkBox2.Checked || checkBox3.Checked || checkBox4.Checked)
        {
            if (checkBox1.Checked)
            {
                answer_1 = false;
                checkBox1.BackColor = Color.Red;
            }
            else
            {
                answer_1 = true;
                checkBox1.BackColor = Color.Green;
            }
            if (checkBox2.Checked)
            {
                answer_2 = false;
                checkBox2.BackColor = Color.Red;
            }
            else
            {
                answer_2 = true;
                checkBox2.BackColor = Color.Green;
            }
            if (checkBox3.Checked)
            {
                answer_3 = true;
                checkBox3.BackColor = Color.Green;
            }
            else
            {
                answer_3 = false;
                checkBox3.BackColor = Color.Red;
            }
            if (checkBox4.Checked)
            {
                answer_4 = true;
                checkBox4.BackColor = Color.Green;
            }
            else
            {
                answer_4 = false;
                checkBox4.BackColor = Color.Red;
            }
            if (answer_1 == true && answer_2 == true

```

```

        && answer_3 == true && answer_4 == true)
    {
        if (a_t == true)
        {
            var form2 = new Test3_1();
            form2.Show();
            this.Hide();
        }
        else
        {
            button_close.Visible = false;
            button_next.Visible = false;
            button1.Visible = true;
            button2.Visible = true;
            checkBox1.BackColor = SystemColors.Control;
            checkBox2.BackColor = SystemColors.Control;
            checkBox3.BackColor = SystemColors.Control;
            checkBox4.BackColor = SystemColors.Control;
        }
    }
    else
    {
        string message1 = "При виборі допущено помилки. Визначник другого порядку є різницею між
добутками основної та зворотної діагоналей. Він дорівнює 1 для третьої та четвертої матриці";
        string title = "Помилка";
        MessageBox.Show(message1, title);
    }
}
else
{
    string message1 = "Будь-ласка, надайте відповідь";
    string title = "Помилка";
    MessageBox.Show(message1, title);
    checkBox1.BackColor = SystemColors.Control;
    checkBox2.BackColor = SystemColors.Control;
    checkBox3.BackColor = SystemColors.Control;
    checkBox4.BackColor = SystemColors.Control;
}
}
private void button_close_Click(object sender, EventArgs e)
{
    this.Close();
    Close();
}
private void label3_Click(object sender, EventArgs e)
{
    string message = "Визначник другого порядку обчислюється як різниця добутків основної діагоналі та
зворотної";
    string title = "Підказка";
    MessageBox.Show(message, title);
}
}
}

```

3.5 Завдання 3

Нових прийомів, блок-схем чи рішень у собі не містить. Від завдання 2 відрізняється виключно математичною складовою та самими питаннями.

3.5.1 Завдання 3 Крок 1

Використовує ту ж блок-схему, що і Завдання 1 Крок 1. Значення за замовчанням встановлено, щоб уникнути перевірки на наявність відповіді.

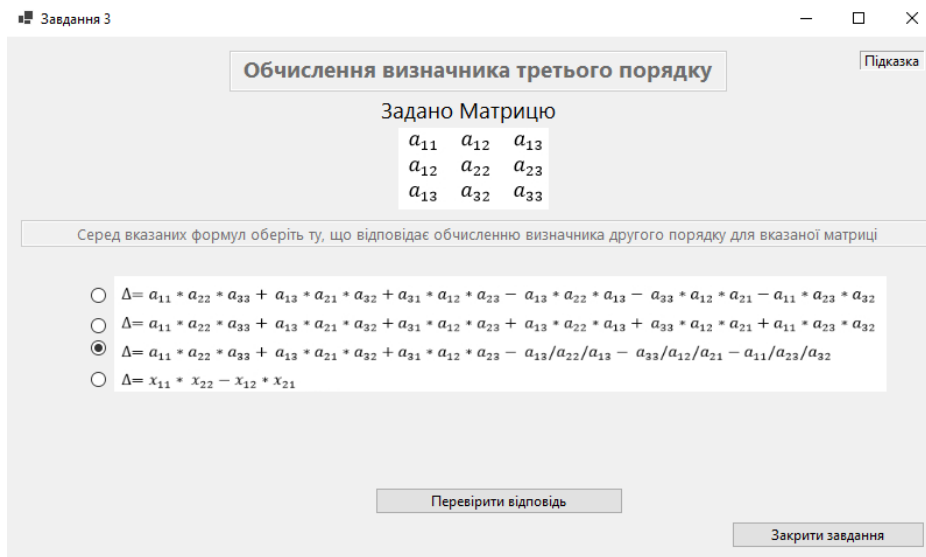


Рис. 3.5.1.1 – Вікно застосунку для «Завдання 1 Крок 1»

Повідомлення у «Підказка» та при неправильних відповідях реалізовані за допомогою MessageBox.

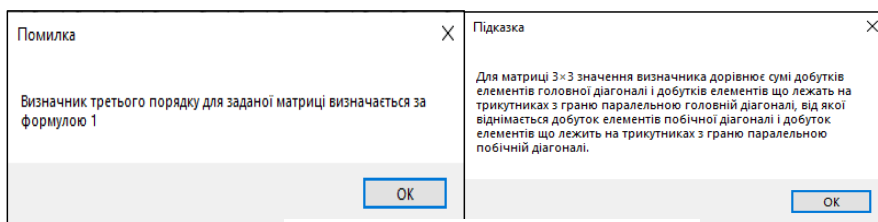


Рис. 3.5.1.2 – Вікна «Помилка» та «Підказка»

Особливістю Завдання, через наявність правильної відповіді у повідомленні про помилку є одночасне відмічення кольором всіх питань – правильних, та неправильних.

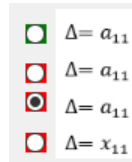


Рис. 3.5.1.3 – Фрагмент вікна застосунку, котрий кольорами підказує відповідь

Код форми поданий у додатку А.

3.5.2 Завдання 3 Крок 2

Є розширеною версією Завдання 1 Крок 3. Тут використовується порівняння тексту з текстом. Блок-схема має такий самий вигляд, як і блок-схема алгоритму тренажера в загальному випадку. Різницю складають тільки код та складові інтерфейсу.

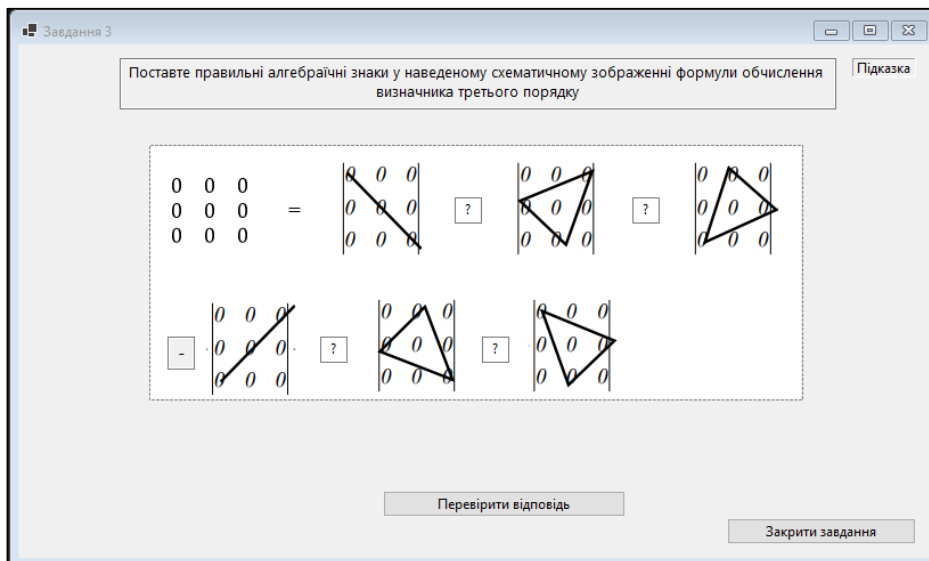


Рис. 3.5.2.1 – Вікна «Помилка» та «Підказка»

Даний крок містить 2 повідомлення про помилки:

1. Коли введені недопустимі знаки (всім крім знаків «+» та «-») або не зроблено жодного вводу.

2. Коли знаки проставлені у невідповідному порядку. Дане повідомлення також супроводжується відміткою невірних відповідей за допомогою кольору.

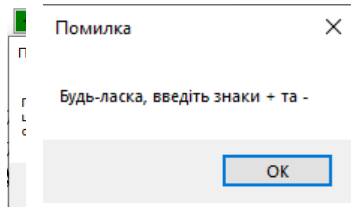


Рис. 3.5.2.2 – Повідомлення про помилки з фрагментом виділення помилок кольором

Підказка в свою чергу містить опис формули, котру користувач має заповнити та реалізована через MessageBox.

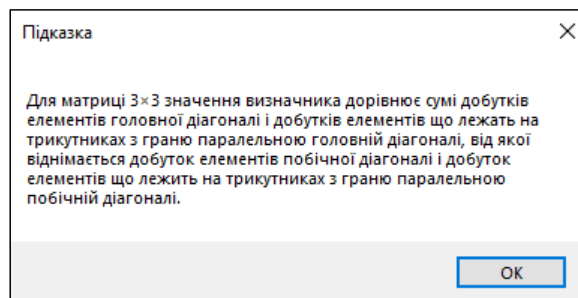


Рис. 3.5.2.3 – Повідомлення, що містить текстовий опис формули у якості

Код цієї форми дивись у додатку А.

3.5.3 Завдання 3 Крок 3

Має блок-схему, що являє собою суміш загальної блок-схеми та блок-схеми для останніх кроків кожного завдання:

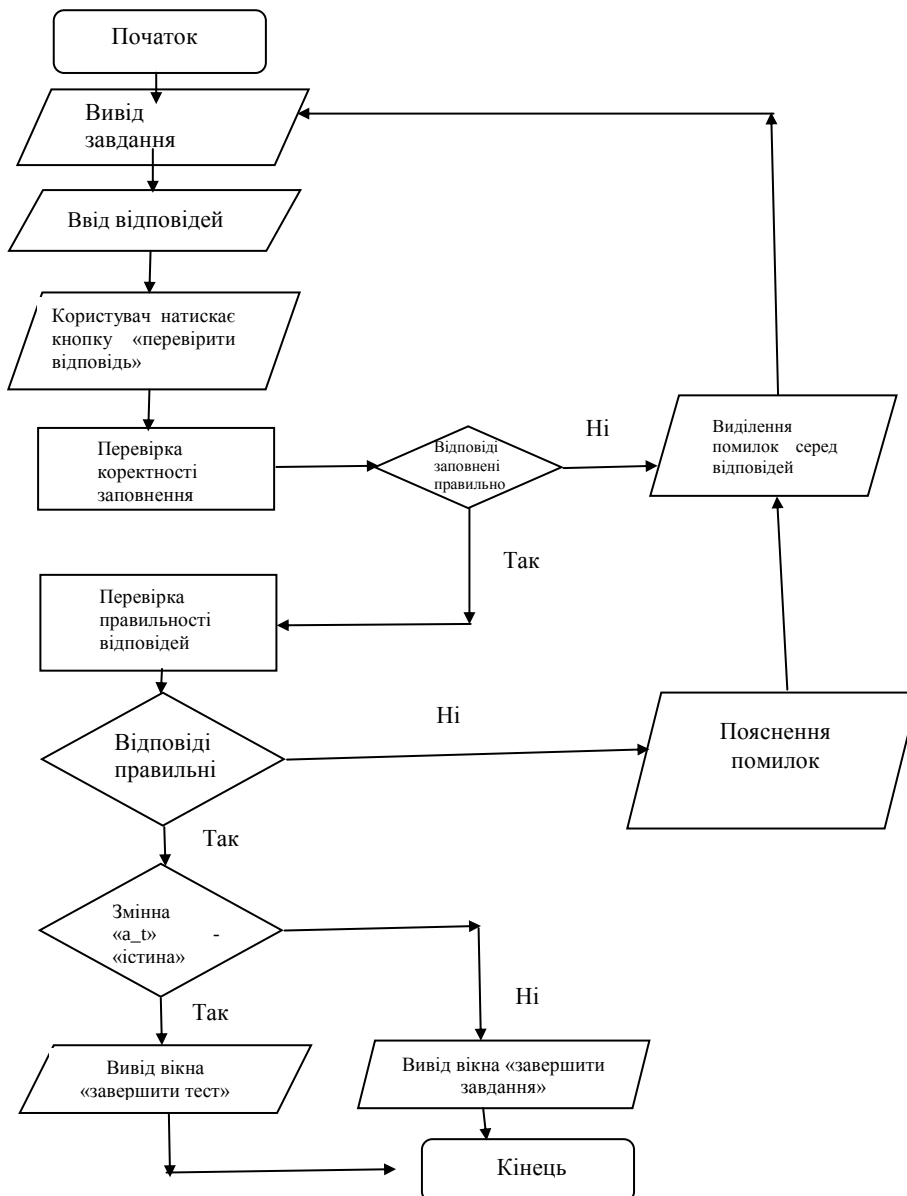


Рис. 3.5.3.1 – Блок-схема для Форми, що відповідає Завданню 3 Крок 3

Вікно завдання не виглядає складним:

Обчисліть визначник третього порядку для запропонованої матриці

Підказка

$$\begin{pmatrix} 2 & 3 & 1 \\ 4 & 2 & 3 \\ 2 & 4 & 1 \end{pmatrix}$$

$\Delta =$?

Відповідь вірна

Закрити завдання

Закрити завдання

Рис. 3.5.3.2 – Вікно Форми, що відповідає Завданню 3 Крок 3

Користувач має ввести значення, замінивши ним символ «?». В залежності від відповіді змінюється повідомлення про помилку:

Помилка

Введіть значення від -5 до 5

ОК

Помилка

Правильне обчислення виконується за формулою:
 $\Delta = a_{11} \cdot a_{22} \cdot a_{33} + a_{13} \cdot a_{21} \cdot a_{32} + a_{31} \cdot a_{12} \cdot a_{23} - a_{13} \cdot a_{22} \cdot a_{13} - a_{33} \cdot a_{12} \cdot a_{21} - a_{11} \cdot a_{23} \cdot a_{32}$

ОК

Рис. 3.5.3.3 – Вікно з передбаченими помилками

У якості «Підказки» слугує наступне вікно:

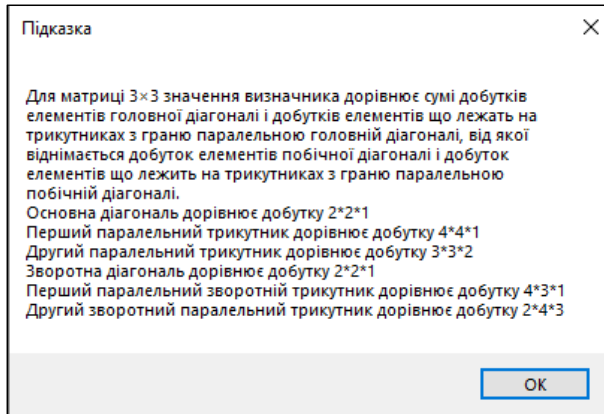


Рис. 3.5.3.4 – Вікно «Підказка»

Коли користувач надає правильну відповідь, перевіряється стан змінної «a_t». Якщо вона має значення «не вірно» - кнопка «перевірити відповідь» зникає з екрану, а кнопка «закрити завдання» переміщується на середину (рис. 3.5.3.5).

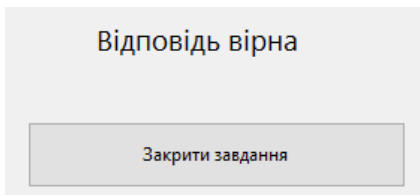


Рис. 3.5.3.5 – Фрагмент модифікованого вікна

У іншому випадку, якщо користувач проходив весь тест, з'явиться наступне повідомлення:

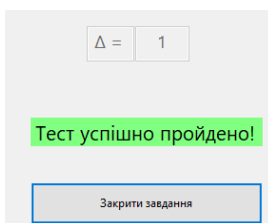


Рис. 3.5.3.6 – Інша модифікація вікна

При цьому текстове поле блокується для вводу. Код для даної форми:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

namespace Definers_test
{
    public partial class Test3_3 : Form
    {
        public Test3_3()
        {
            InitializeComponent();
        }

        private void button_next_Click(object sender, EventArgs e)
        {
            string correct = "1";
            string answer = textBox3.Text;
            var cor_in = new List<string> { "-5", "-4", "-3", "-2", "-1", "0", "1", "2", "3", "4", "5" };
            bool a_t = Form1.a_t;
            bool compare = cor_in.Contains(answer);
            if (compare == true)
            {
                if (correct == answer)
                {
                    if (a_t == true)
                    {
                        button_next.Visible = false;
                        button_close.Visible = false;
                        button_close_all.Visible = true;
                        label2.Visible = true;
                        textBox3.Enabled = false;
                    }
                    else
                    {
                        button_next.Visible = false;
                        button_close.Visible = false;
                        button_close_all.Visible = true;
                        label1.Visible = true;
                        textBox3.Enabled = false;
                    }
                }
                else
                {
                    string message = "Правильне обчислення виконується за формулою:  $\Delta = a_{11} \cdot a_{22} \cdot a_{33} +$   

 $a_{13} \cdot a_{21} \cdot a_{32} + a_{31} \cdot a_{12} \cdot a_{23} - a_{13} \cdot a_{22} \cdot a_{13} - a_{33} \cdot a_{12} \cdot a_{21} - a_{11} \cdot a_{23} \cdot a_{32}$ ";
                    string title = "Помилка";
                    MessageBox.Show(message, title);
                }
            }
            else
            {
                string message = "Введіть значення від -5 до 5";
                string title = "Помилка";
                MessageBox.Show(message, title);
            }
        }

        private void Test3__FormClosed(object sender, FormClosedEventArgs e)
        {
            var form1 = new Form1();
            form1.Show();
        }

        private void label3_Click(object sender, EventArgs e)
        {
            string message = "Для матриці 3×3 значення визначника дорівнює сумі добутків елементів головної  

діагоналі і добутків елементів що лежать на трикутниках з граню паралельною головної діагоналі, від якої віднімається

```

добуток елементів побічної діагоналі і добуток елементів що лежить на трикутниках з граню паралельною побічній діагоналі."

```

        + "\n" + "Основна діагональ дорівнює добутку 2*2*1" + "\n" + "Перший паралельний трикутник
дорівнює добутку 4*4*1" + "\n" + "Другий паралельний трикутник дорівнює добутку 3*3*2"
        + "\n" + "Зворотна діагональ дорівнює добутку 2*2*1" + "\n" + "Перший паралельний зворотній
трикутник дорівнює добутку 4*3*1" + "\n" +
        "Другий зворотний паралельний трикутник дорівнює добутку 2*4*3";
        string title = "Підказка";
        MessageBox.Show(message, title);
    }
    private void button_close_Click(object sender, EventArgs e)
    {
        this.Close();
        Close();
    }
    private void button_close_all_Click(object sender, EventArgs e)
    {
        this.Close();
        Close();
    }
}
}

```

Робота тренажеру завершується поверненням на першу форму (див. 3.2). Закрити програму можливу тільки закривши це вікно, або натисканням на кнопку «закрити тренажер». Будь-які переривання роботи програми іншим чином автоматично викликають вказане вікно.

Висновки

Дистанційне навчання чи навчання в аудиторіях довгий час не має практичного повторення, котре буде зручним для використання і дозволить повторювати матеріал. Саме тому тренажери з тем навчального матеріалу дозволяють у максимально простій формі засвоювати складні формули та прийоми роботи з матеріалом. В результаті роботи мною було розроблено вимоги та структуру програмного тренажеру. Завдання з програмної реалізації виконано у повному обсязі, з урахуванням можливих винятків при роботі програми. Інтерфейс програмного рішення мінімізований та має необхідні підказки, котрі легко знайти. Питання мінімізації використання ресурсів розв'язане максимально, завдяки обраним методикам реалізації програми.

Програмне забезпечення тренажеру з теми «Визначники» дистанційного навчального курсу «Прикладна математика» реалізоване у повному обсязі і може бути використане студентами, як у дистанційному курсі навчання, так і в аудиторіях.

Основні результати роботи:

- Сформульовані основні вимоги до програмного забезпечення тренажеру;
- Виконано опис проектних рішень, інструментів та підходів до розробки програмного забезпечення тренажеру;
- Розроблено інтерфейс тренажеру;
- Програмно реалізовано тренажер з теми «визначники» дистанційного навчального курсу «прикладна математика»;
- Створено блок-схеми для компонентів тренажеру, що пояснюють його роботу;
- Проведено деталізований огляд програмних рішень, що використані при створенні тренажеру.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Положення про дистанційне навчання // Офіційний сайт Українського інституту інформаційних технологій в освіті [Електронний ресурс]. – Режим доступу: <http://udec.ntu-kpi.kiev.ua>.
2. Економічні тренажери // ПрАТ "Литер" – офіційний дистрибутор [SMART Technologies в Україні](http://www.smartboard.com.ua/about/) [Електронний ресурс]. – Режим доступу: <http://www.smartboard.com.ua/about/>
3. Кухаренко В. Практикум дистанционного обучения / В.Кухаренко. – Изд. 2-е. – К.: Милленниум, 2003. – 196 с. – ISBN 966-8063-30-21
4. Вища математика в прикладах і задачах: Навч. посібник / В.Ю. Клепко; В.Л. Голець. – К.: ЦУЛ, 2006. – 600 с.
5. Дистанційна освіта: забезпечення доступності та неперервної освіти впродовж життя (E-Learning and University Education 2017) : матеріали XLII Міжнародної науково-методичної конференції (м. Полтава, 9–10 лютого 2017 року). – Полтава : ПУЕТ, 2017. – 365 с
6. Програмування мовою C // З. Я. Шпак – Вид. 2-е – К: Львівська політехніка, 2011 – 436 с. – ISBN 978-617-607-104-4
7. Create a Windows Forms app in Visual Studio with C# // Офіційний сайт довідки «Microsoft» [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>
8. Офіційна колекція уроків по роботі з Visual Studio // Офіційний сайт довідки «Microsoft» [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/visualstudio/get-started/csharp/?view=vs-2022>
9. Andrew Troelsen, Pro C# 2010 and the .NET 4 Platform / A.Troelsen – Ed.4 – An.Apress, 2010 – 1762p. – ISBN 978-1430225492

Додаток А

Завдання 3 Крок 1. Код компонента програми:

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

namespace Definers_test
{
    public partial class Test3_1 : Form
    {
        public Test3_1()
        {
            InitializeComponent();
        }
        private void Test3_1_FormClosed(object sender, FormClosedEventArgs e)
        {
            var form1 = new Form1();
            form1.Show();
        }
        private void button_next_Click(object sender, EventArgs e)
        {
            if (radioButton1.Checked)
            {
                var form3 = new Test3_2();
                form3.Show();
                this.Hide();
            }
            else
            {
                string message1 = "Визначник третього порядку для заданої матриці визначається за формулою 1";
                string title = "Помилка";
                MessageBox.Show(message1, title);
                radioButton1.BackColor = Color.Green;
                radioButton2.BackColor = Color.Red;
                radioButton3.BackColor = Color.Red;
                radioButton4.BackColor = Color.Red;
            }
        }
        private void button_close_Click(object sender, EventArgs e)
        {
            this.Close();
            Close();
        }
        private void label3_Click(object sender, EventArgs e)
        {
            string message = "Для матриці 3×3 значення визначника дорівнює сумі добутоків елементів головної діагоналі і добутоків елементів що лежать на трикутниках з граню паралельною головної діагоналі, від якої віднімається добуток елементів побічної діагоналі і добуток елементів що лежить на трикутниках з граню паралельною побічній діагоналі.";
            string title = "Підказка";
            MessageBox.Show(message, title);
        }
    }
}
```

Завдання 3 Крок 2 код форми

```
using System;
using System.Collections.Generic;
```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Definers_test
{
    public partial class Test3_2 : Form
    {
        public Test3_2()
        {
            InitializeComponent();
        }
        private void Test3__FormClosed(object sender, FormClosedEventArgs e)
        {
            var form1 = new Form1();
            form1.Show();
        }

        public object TextBox1 { get; private set; }
        public object TextBox5 { get; private set; }
        public object TextBox3 { get; private set; }
        public object TextBox4 { get; private set; }
        private void button_next_Click(object sender, EventArgs e)
        {
            string cur_answer_1 = textBox1.Text;
            string valid_answer_1 = "+";
            string cur_answer_2 = textBox3.Text;
            string valid_answer_2 = "+";
            string cur_answer_3 = textBox4.Text;
            string valid_answer_3 = "-";
            string cur_answer_4 = textBox5.Text;
            string valid_answer_4 = "-";
            var cor_inp = new List<string> { "+", "-" };
            //Перевіряємо символи у першій чарунці
            bool comp1 = cor_inp.Contains(cur_answer_1, StringComparer.OrdinalIgnoreCase);
            bool comp2 = cor_inp.Contains(cur_answer_2, StringComparer.OrdinalIgnoreCase);
            bool comp3 = cor_inp.Contains(cur_answer_3, StringComparer.OrdinalIgnoreCase);
            bool comp4 = cor_inp.Contains(cur_answer_4, StringComparer.OrdinalIgnoreCase);
            //Перевіряємо чи коректними символами заповнені чарунки
            if (comp1 == true && comp2 == true && comp3 == true && comp4 == true)
            {
                //Цей рядок буде скидати колір, щоб він не транзитивався при повторних помилках користувача
                textBox1.BackColor = SystemColors.Window;
                textBox3.BackColor = SystemColors.Window;
                textBox4.BackColor = SystemColors.Window;
                textBox5.BackColor = SystemColors.Window;
                //Кілька змінних, щоб порівнювати значення відповідей і маніпулювати далі
                bool answe1;
                bool answe2;
                bool answe3;
                bool answe4;
                //порівнюємо першу відповідь. Якщо вона правильна, відмічаємо чарунку зеленим. Якщо ні - червоним
                if (cur_answer_1 == valid_answer_1) // "==" використовуємо тому що маємо справу з порівнянням, а не
                прирівнюємо значення а) до значення б)
                {
                    answe1 = true;
                    textBox1.BackColor = Color.Green;
                }
                else
                {

```

```

        answe1 = false;
        textBox1.BackColor = Color.Red;
    }
    //порівнюємо другу відповідь. Якщо вона правильна, відмічаємо чарунку зеленим. Якщо ні - червоним
    if (cur_answer_2 == valid_answer_2)
    {
        answe2 = true;
        textBox3.BackColor = Color.Green;
    }
    else
    {
        answe2 = false;
        textBox3.BackColor = Color.Red;
    }
    //порівнюємо третю відповідь. Якщо вона правильна, відмічаємо чарунку зеленим. Якщо ні - червоним
    if (cur_answer_3 == valid_answer_3)
    {
        answe3 = true;
        textBox4.BackColor = Color.Green;
    }
    else
    {
        answe3 = false;
        textBox4.BackColor = Color.Red;
    }
    //порівнюємо четверту відповідь. Якщо вона правильна, відмічаємо чарунку зеленим. Якщо ні -
    червоним
    if (cur_answer_4 == valid_answer_4)
    {
        answe4 = true;
        textBox5.BackColor = Color.Green;
    }
    else
    {
        answe4 = false;
        textBox5.BackColor = Color.Red;
    }
    //Тепер скористаємось змінною answe"x", щоб виявити чи надав користувач правильну відповідь
    if (answe1 == true && answe2 == true
        && answe3 == true && answe4 == true)
    //Якщо виконуються всі 4 умови - переходимо до наступної форми
    {
        var form3 = new Test3_3();
        form3.Show();
        this.Hide();
    }
    //Якщо умови не виконуються - обійдемося без конструкції з Form7. Тут нам достатньо повідомлення
    про помилку
    else
    {
        string message1 = "При виборі допущено помилки. Правильний порядок обчислення це додавання
        перших трьох переліків елементів і віднімання останніх трьох";
        string title = "Помилка";
        MessageBox.Show(message1, title);
    }
    else
    {
        string message2 = "Будь-ласка, введіть знаки + та -";
        string title = "Помилка";
        MessageBox.Show(message2, title);
    }
}
private void button_close_Click(object sender, EventArgs e)
{
    this.Close();
}

```

```
        Close();
    }
    private void label3_Click(object sender, EventArgs e)
    {
        string message = "Для матриці 3×3 значення визначника дорівнює сумі добутків елементів головної  
діагоналі і добутків елементів що лежать на трикутниках з граню паралельною головної діагоналі, від якої віднімається  
добуток елементів побічної діагоналі і добуток елементів що лежить на трикутниках з граню паралельною побічній  
діагоналі.";
        string title = "Підказка";
        MessageBox.Show(message, title);
    }
}
```

Завдання 2 Крок 1, код форми:

```

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

namespace Definers_test
{
    public partial class Test2_1 : Form
    {
        public Test2_1()
        {
            InitializeComponent();
        }
        private void Test2_1_FormClosed(object sender, FormClosedEventArgs e)
        {
            var form1 = new Form1();
            form1.Show();
        }
        private void button_next_Click(object sender, EventArgs e)
        {
            radioButton1.BackColor = Color.White;
            radioButton2.BackColor = Color.White;
            radioButton3.BackColor = Color.White;
            radioButton4.BackColor = Color.White;

            if (radioButton4.Checked)
            {
                var form2 = new Test2_2();
                form2.Show();
                this.Hide();
            }
            else if (radioButton1.Checked)
            {
                string message1 = "Визначник другого порядку обчислюється як різниця добутків основної діагоналі та зворотної. У наведений матриці це  $\Delta = x_{11} * x_{22} - x_{12} * x_{21}$ ";
                string title = "Помилка";
                MessageBox.Show(message1, title);
                radioButton1.BackColor = Color.Red;
            }
            else if (radioButton2.Checked)
            {
                string message1 = "Визначник другого порядку обчислюється як різниця добутків основної діагоналі та зворотної. У наведений матриці це  $\Delta = x_{11} * x_{22} - x_{12} * x_{21}$ ";
                string title = "Помилка";
                MessageBox.Show(message1, title);
                radioButton2.BackColor = Color.Red;
            }
            else
            {
                string message1 = "Визначник другого порядку обчислюється як різниця добутків основної діагоналі та зворотної. У наведений матриці це  $\Delta = x_{11} * x_{22} - x_{12} * x_{21}$ ";
                string title = "Помилка";
                MessageBox.Show(message1, title);
                radioButton3.BackColor = Color.Red;
            }
        }
        private void button_close_Click(object sender, EventArgs e)
        {
            this.Close();
            Close();
        }
    }
}

```

```
    }  
    private void label3_Click(object sender, EventArgs e)  
    {  
        string message = "Визначник другого порядку обчислюється як різниця добутків основної діагоналі та  
зворотної";  
        string title = "Підказка";  
        MessageBox.Show(message, title);  
    }  
}
```

Завдання 2 Крок 2 код форми

```

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

namespace Definers_test
{
    public partial class Test2_2 : Form
    {
        public Test2_2()
        {
            InitializeComponent();
        }
        private void Test2_2_FormClosed(object sender, FormClosedEventArgs e)
        {
            var form1 = new Form1();
            form1.Show();
        }
        private void button_next_Click(object sender, EventArgs e)
        {
            //Для комбобоксів не існує можливості задати значення за замовчанням. Через це вони повертають нуль
            //при помилковому натисканні
            //або якщо користувач не заповнив ніяких значень, що аварійно завершує програму. Дана конструкція
            //дозволяє уникнути цієї помилки
            if (comboBox1.SelectedIndex == -1 || comboBox2.SelectedIndex == -1 || comboBox3.SelectedIndex == -1 ||
                comboBox4.SelectedIndex == -1)
            {
                string message2 = "Будь ласка, заповніть відповіді";
                string title = "Помилка";
                MessageBox.Show(message2, title);
            }
            else
            {
                //Оскільки значення в випадючих списках є текстом, треба перетворити їх на число для подальшого
                //порівняння
                int cur_answer_1 = Convert.ToInt32(comboBox1.Text);
                int cur_answer_2 = Convert.ToInt32(comboBox2.Text);
                int cur_answer_3 = Convert.ToInt32(comboBox3.Text);
                int cur_answer_4 = Convert.ToInt32(comboBox4.Text);
                //Задасмо значення для порівняння
                int valid_answer_1 = 4;
                int valid_answer_2 = 1;
                int valid_answer_3 = 2;
                int valid_answer_4 = 3;
                if (cur_answer_1 == valid_answer_1
                    && cur_answer_2 == valid_answer_2
                    && cur_answer_3 == valid_answer_3
                    && cur_answer_4 == valid_answer_4)
                {
                    var form2 = new Test2_3();
                    form2.Show();
                    this.Hide();
                }
                else
                {
                    string message = "Відповідь не вірна. Визначник другого порядку першої матриці – 4, другої – 1,
                    третьої – 2, четвертої – 3";
                    string title = "Помилка";
                    MessageBox.Show(message, title);
                }
            }
        }
    }
}

```

```
    }  
}  
private void button_close_Click(object sender, EventArgs e)  
{  
    this.Close();  
    Close();  
}  
private void label3_Click(object sender, EventArgs e)  
{  
    string message = "Визначник другого порядку обчислюється як різниця добутків основної діагоналі та  
зворотної";  
    string title = "Підказка";  
    MessageBox.Show(message, title);  
}  
}
```