

ЗМІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів

Кафедра систем управління літальних апаратів

Лабораторна робота № 3

з дисципліни «Алгоритмізація та програмування»

на тему «Реалізація алгоритмів з розгалуженням мовою C ++»

XAI.301.319.5 ЛР

Виконав студент гр. _____ 319 _____

_____ Валерій ЄВЧУК

(підпис, дата)

(П.І.Б.)

Перевірив

_____ к.т.н., доц. Євгеній ПЯВКА

(підпис, дата)

(П.І.Б.)

2025

МЕТА РОБОТИ

Вивчити теоретичний матеріал щодо синтаксису у мові С ++ і подання у вигляді UML діаграм активності алгоритмів з розгалуженням та реалізувати алгоритми з використанням інструкцій умовного переходу і вибору мовою С++ в середовищі QtCreator. Також опанувати та відпрацювати навички структурування програми з функціями.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити дві задачі на алгоритми з розгалуженням. Варіанти представлено в табл.1.

Завдання 2. Дано координати точки на площині (x, y). Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур) і вивести відповідне повідомлення. Варіанти фігур представлено в табл.2.

Завдання 3. Для вибору користувачем одного з трьох зазначених вище завдань розробити алгоритм організації меню в командному вікні з використанням інструкції вибору

Завдання 4. Використовуючи ChatGpt, Gemini або інший засіб генеративного ІІ, провести самоаналіз отриманих знань і навичок за допомогою наступних промптів:

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі If10 (розділ If, задача 10).

Вхідні дані (ім'я, опис, тип, обмеження):

- А – перше ціле число, яке вводить користувач з клавіатури; тип даних – int (або long long), діапазон допустимих значень – будь-яке ціле число, що не виходить за межі обраного цілого типу.
- В – друге ціле число, яке вводить користувач з клавіатури; тип даних – int (або long long), діапазон допустимих значень – будь-яке ціле число, що не виходить за межі обраного цілого типу.

Вихідні дані (ім'я, опис, тип):

- А – нове значення першої змінної після обробки: або сума початкових А і В (якщо $A \neq B$), або 0 (якщо $A = B$); тип даних – цілий (int/long long).
- В – нове значення другої змінної після обробки: або сума початкових А і В (якщо $A \neq B$), або 0 (якщо $A = B$); тип даних – цілий (int/long long).
- текстове сповіщення в консолі з виведеними новими значеннями змінних А та В; тип – текстовий рядок (повідомлення користувачу).

Завдання 2. Вирішення задачі If31 (розділ If, задача 31).

Вхідні дані (ім'я, опис, тип, обмеження):

– x – дійсне число, яке вводить користувач з клавіатури; опис – аргумент функції $g(x)$; тип даних – double (дійсний тип), діапазон допустимих значень – будь-яке дійсне число в межах можливостей типу double.

Вихідні дані (ім'я, опис, тип):

– g – значення цілочисельної функції $g(x)$, обчислене за правилом:

$$g(x) = x + 1, \text{ якщо } x < -1;$$

$$g(x) = x^2 - 1, \text{ якщо } -1 \leq x \leq 1;$$

$$g(x) = 3, \text{ якщо } x > 1;$$

тип даних – цілий (int).

– текстове сповіщення з результатом обчислення у вигляді “ $g(x) = \dots$ ”; тип – текстовий рядок (повідомлення користувачу).

Завдання 3. Вирішення геометричної задачі (табл. 2, варіанти 11–19, варіант 11).

Вхідні дані (ім'я, опис, тип, обмеження):

– r – радіус кіл, що задають геометричну фігуру; вводиться користувачем з клавіатури; тип даних – double; діапазон допустимих значень – дійсне число більше нуля ($r > 0$).

– x – абсциса точки на площині, яку потрібно перевірити; вводиться користувачем; тип даних – double; діапазон – будь-яке дійсне число в межах типу double.

– y – ордината точки на площині, яку потрібно перевірити; вводиться користувачем; тип даних – double; діапазон – будь-яке дійсне число в межах типу double.

Вихідні дані (ім'я, опис, тип):

– inRegion (у коді може бути логічна змінна або лише текст) – логічний результат перевірки належності точки (x, y) зафарбованій області, утвореній частинами двох кіл; тип даних – логічний (bool) – істина або хиба (у програмі використовується для вибору текстового повідомлення).

– текстове сповіщення у консолі одного з двох видів:

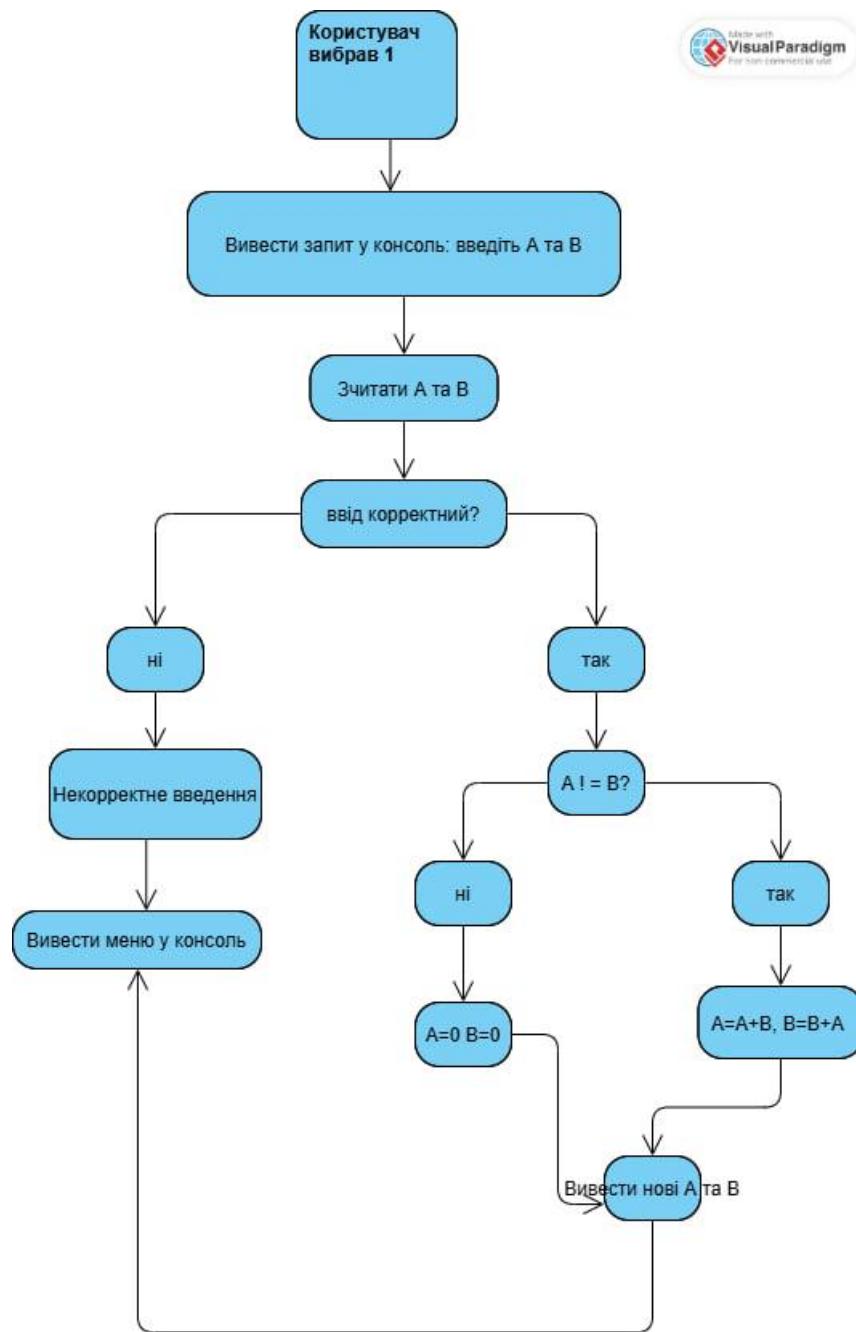
«Точка (x, y) НАЛЕЖИТЬ зафарбованій області.» або

«Точка (x, y) НЕ належить зафарбованій області.»

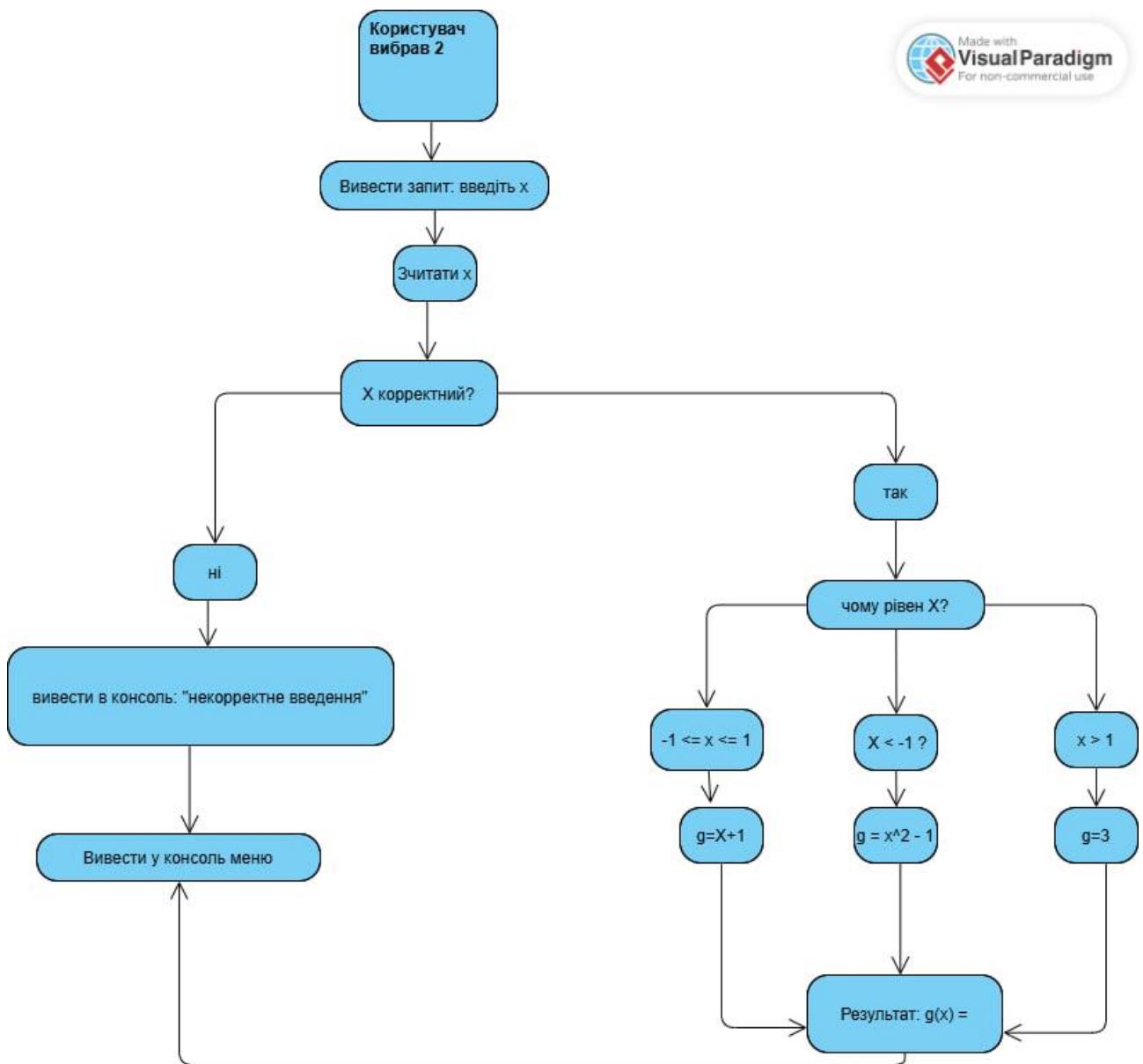
тип – текстовий рядок (повідомлення користувачу).

Алгоритм вирішення завдання 1 показано нижче на рисунку 1

Рисунок 1 – if10

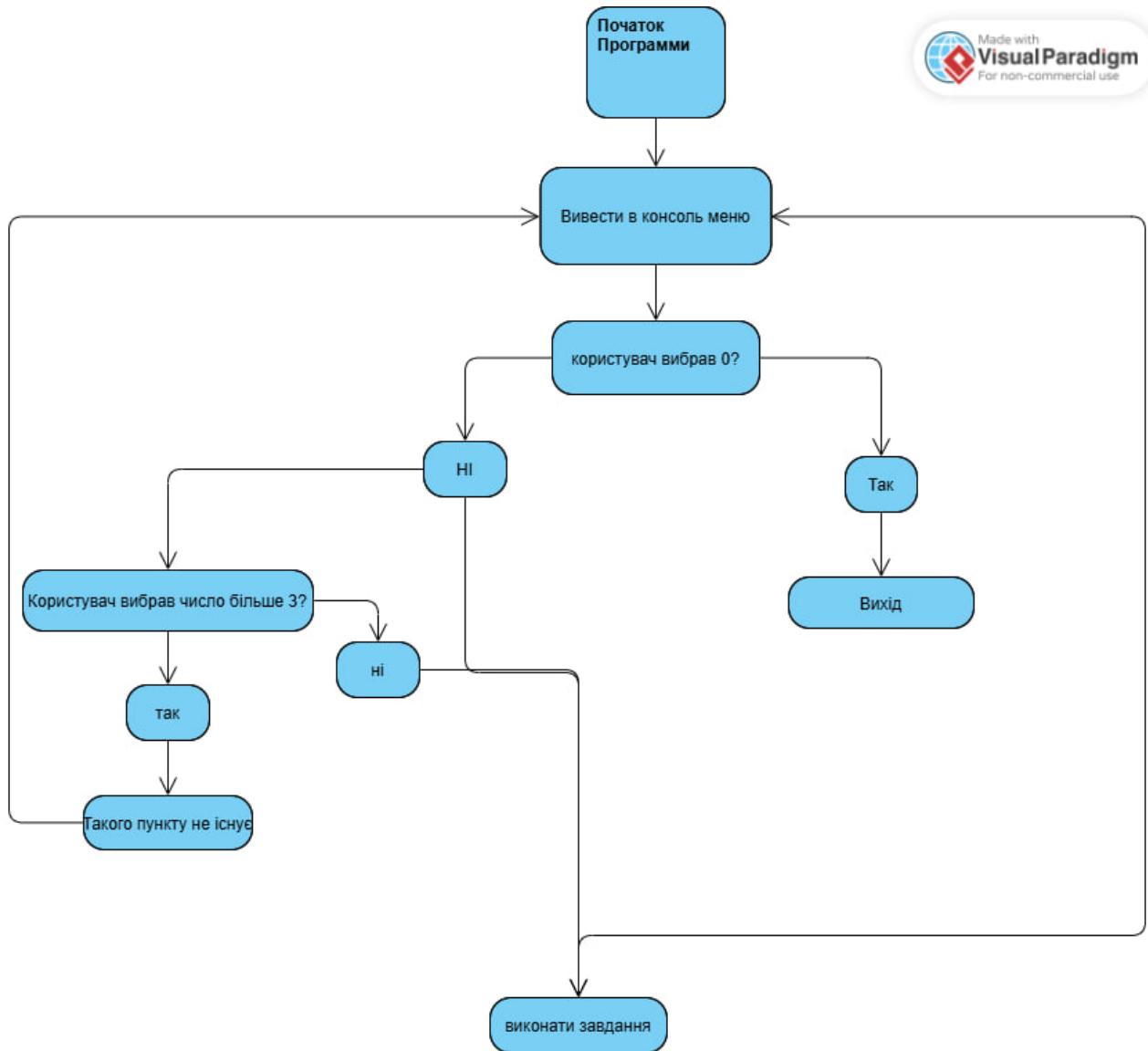


Алгоритм вирішення завдання 2 наведено нижче на рис 2
рисунок 2 – if31



Алгоритм всієї программи наведено нижче на рис 3

рисунок 3 - программа



Лістинг коду вирішення задач наведено в дод. А (стор. 5-8).

Екран роботи програми показаний на рис. Б.1 та Б.2

Завдання 2.

2.А. ...

ВИСНОВКИ

Було вивчено принципи використання розгалужень у мові програмування C++ та закріплено на практиці навички роботи з умовними операторами `if` та логічними виразами.

Під час виконання завдань відпрацьовано введення й перевірку даних, обробку числових і логічних умов, а також реалізацію простих геометричних обчислень. Отримано навички побудови структурованих програм із меню вибору завдань і виведенням результатів для користувача. Невеликі труднощі виникли під час реалізації функції очищення вводу та логіки перевірки належності точки геометричній області, однак вони були успішно усунуті в процесі налагодження коду.

ДОДАТОК А

Лістинг коду програми

```

// Лабораторна робота №3
// Варіанти: If10, If31 та геометричне (варіанти 11-19, варіант 11)

#include <iostream>
#include <cmath>
#include <limits>

using namespace std;

// Прототипи функцій
void завдання_If10();
void завдання_If31();
void завдання_Геометрія11();

void очиститиВвід()
{
    cin.clear();
    cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
}

void меню()
{
    cout << "===== МЕНЮ =====\n";
    cout << "1 - Завдання If10 (цилі числа A і B)\n";
    cout << "2 - Завдання If31 (обчислення g(x))\n";
    cout << "3 - Геометричне завдання (варіант 11)\n";
    cout << "0 - Вихід із програми\n";
    cout << "-----\n";
    cout << "Ваш вибір: ";
}

int main()
{
    int вибір;

    do
    {
        меню();
        cin >> вибір;

        if (!cin)
        {
            cout << "Помилка вводу! Спробуйте ще раз.\n\n";
        }
    } while (вибір != 0);
}

```

```

очиститиВвід();
continue;
}

switch (вибір)
{
case 1:
    завдання_If10();
    break;
case 2:
    завдання_If31();
    break;
case 3:
    завдання_Геометрія11();
    break;
case 0:
    cout << "Програму завершено.\n";
    break;
default:
    cout << "Такого пункту меню не існує.\n";
}

cout << "\n";
} while (вибір != 0);

return 0;
}

// -----
// If10. Дано два цілі числа A і B.
// Якщо їх значення не рівні - присвоїти кожній змінній суму цих чисел,
// якщо рівні - присвоїти нуль.
// -----
void завдання_If10()
{
    cout << "\n***** Завдання If10 *****\n";
    long long A, B;

    cout << "Введіть цілі числа A та B: ";
    cin >> A >> B;

    if (!cin)
    {
        cout << "Некоректне введення!\n";
        очиститиВвід();
        return;
    }

    if (A != B)
    {

```

```

        long long cума = A + B;
        A = cума;
        B = cума;
    }
else
{
    A = 0;
    B = 0;
}

cout << "Нові значення: A = " << A << ", B = " << B << endl;
}

// -----
// If31. Для заданого дійсного x знайти значення функції g(x):
//
// g(x) = x + 1, якщо x < -1
// g(x) = x^2 - 1, якщо -1 <= x <= 1
// g(x) = 3, якщо x > 1
//
// Результат цілий (int).
// -----
void завдання_If31()
{
    cout << "\n***** Завдання If31 *****\n";
    double x;
    cout << "Введіть значення x: ";
    cin >> x;

    if (!cin)
    {
        cout << "Некоректне введення!\n";
        очиститиВвід();
        return;
    }

    int g;

    if (x < -1.0)
        g = static_cast<int>(x + 1.0);
    else if (x <= 1.0)
        g = static_cast<int>(x * x - 1.0);
    else
        g = 3;

    cout << "Результат: g(x) = " << g << endl;
}

// -----
// Геометричне завдання (варіант 11).

```

```

// Перевірити, чи належить точка (x, y) зафарбованій області,
// утвореній двома чвертями кіл радіуса r.
// -----
void завдання_Геометрія11()
{
    cout << "\n***** Геометричне завдання (варіант 11) *****\n";

    double r;
    cout << "Введіть радіус r (>0): ";
    cin >> r;
    if (!cin || r <= 0.0)
    {
        cout << "Некоректний радіус!\n";
        очиститиВвід();
        return;
    }

    double x, y;
    cout << "Введіть координати точки (x y): ";
    cin >> x >> y;

    if (!cin)
    {
        cout << "\n";"Некоректні координати!
        очиститиВвід();
        return;
    }

    // Верхнє коло з центром (0, r)
    bool верхняЧверть =
        (x >= 0.0) && (y >= 0.0) &&
        (x * x + (y - r) * (y - r) <= r * r + 1e-9);

    // Нижнє коло з центром (-r, -r)
    bool нижняЧверть =
        (x <= 0.0) && (y <= 0.0) &&
        ((x + r) * (x + r) + (y + r) * (y + r) <= r * r + 1e-9);

    bool належить = верхняЧверть || нижняЧверть;

    if (належить)
        cout << "Точка (" << x << ", " << y << ") НАЛЕЖИТЬ зафарбованій області.\n";
    else
        cout << "Точка (" << x << ", " << y << ") НЕ належить зафарбованій
області.\n";
}

```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

Рисунок - Екран виконання програми для вирішення завдання if10

```

-----  

1 - Завдання If10 (цілі числа A і B)  

2 - Завдання If31 (обчислення g(x))  

3 - Геометричне завдання (варіант 11)  

0 - Вихід із програми  

-----  

Ваш вибір: 1  

***** Завдання If10 *****  

Введіть цілі числа A та B: 1  

1  

Нові значення: A = 0, B = 0  

===== МЕНЮ =====  

1 - Завдання If10 (цілі числа A і B)  

2 - Завдання If31 (обчислення g(x))  

3 - Геометричне завдання (варіант 11)  

0 - Вихід із програми  

-----  

Ваш вибір: 1  

***** Завдання If10 *****  

Введіть цілі числа A та B: 2  

3  

Нові значення: A = 5, B = 5  

----- МЕНЮ -----

```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
if 31

```
-----  
Ваш вибір: 2  
  
***** Завдання If31 *****  
Введіть значення x: 1  
Результат: g(x) = 0  
  
===== МЕНЮ =====  
1 - Завдання If10 (цілі числа A ♦ B)  
2 - Завдання If31 (обчислення g(x))  
3 - Геометричне завдання (варіант 11)  
0 - Вихід із програми  
-----  
Ваш вибір: 2  
  
***** Завдання If31 *****  
Введіть значення x: 3  
Результат: g(x) = 3  
  
===== МЕНЮ =====  
1 - Завдання If10 (цілі числа A ♦ B)  
2 - Завдання If31 (обчислення g(x))  
3 - Геометричне завдання (варіант 11)  
0 - Вихід із програми  
-----  
Ваш вибір:
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання номер 3
11вар

```
-----  
Ваш вибір: 3  
  
***** Геометричне завдання (варіант 11) *****  
введіть радіус r (>0): 3  
введіть координати точки (x y): 3  
3  
Точка (3, 3) НАЛЕЖИТЬ зафарбованій області.  
  
===== МЕНЮ =====  
1 - Завдання If10 (цілі числа A і B)  
2 - Завдання If31 (обчислення  $g(x)$ )  
3 - Геометричне завдання (варіант 11)  
0 - Вихід із програми  
  
-----  
Ваш вибір: 3  
  
***** Геометричне завдання (варіант 11) *****  
введіть радіус r (>0): 3  
введіть координати точки (x y): 10  
10  
Точка (10, 10) НЕ належить зафарбованій області.  
  
===== МЕНЮ =====  
1 - Завдання If10 (цілі числа A і B)  
2 - Завдання If31 (обчислення  $g(x)$ )  
3 - Геометричне завдання (варіант 11)  
0 - Вихід із програми  
  
-----  
Ваш вибір:
```

ДОДАТОК В

Діалог з ШІ для самоаналізу

Використовуючи ChatGpt, Gemini або інший засіб генеративного ШІ, провести самоаналіз отриманих знань і навичок за допомогою наступних промптів: 1) «Ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань. Це мають бути завдання <середнього> рівня складності на розвиток критичного та інженерного мислення. Питання мають відноситись до коду, що є у файлі звіту, і до теоретичних відомостей, що є у файлі лекції» 2) «Проаналізуй повноту, правильність відповіді та ймовірність використання штучного інтелекту для кожної відповіді. Оціни кожне питання у 5-балльній шкалі, віднімаючи 60% балів там, де ймовірність відповіді з засобом ШІ висока. Обчисли загальну середню оцінку»

Добре, розпочіммо іспит 😊

Я — викладач, що приймає захист твоєї роботи з лабораторної №3.

Питання стосуються твого коду (If10, If31, геометричне завдання) та теоретичних знань з розгалужень у C++.

Твоя задача — відповісти на всі питання по черзі.

1 У програмі функція очиститиВвід() виконує:

- A. Перевірку правильності введених чисел
- B. Очищення буфера вводу після помилки користувача
- C. Завершення програми у разі хибного вводу
- D. Змінює тип введених змінних на текстовий

2 У завданні If10 програма присвоює обом змінним A і B значення їх суми, якщо:

- A. A > B
- B. A = B
- C. A ≠ B
- D. A < 0

3 У завданні If31 при значенні x = 0 результатом буде:

- A. 3
- B. 0
- C. -1
- D. 1

4 Для чого використовується приведення типу static_cast<int>(x) у завданні If31?

- A. Щоб округлити дійсне значення до цілого типу
- B. Щоб змінити знак числа
- C. Щоб уникнути помилки компіляції
- D. Щоб перевірити тип змінної

5 Яке призначення має головне меню програми?

- A. Для обчислення результатів
- B. Для повторного запуску компіляції
- C. Для виводу підказок користувачу
- D. Для вибору одного з трьох завдань у межах однієї програми

◆ Блок 2. Відкриті питання (5 шт.)

- 6** Поясни коротко, як саме реалізується перевірка належності точки до фігури у геометричному завданні.
- 7** Яку роль відіграє логічний оператор `||` у виразі перевірки області?
- 8** Якби ти хотів додати четверте завдання до меню, що б потрібно було змінити у коді?
- 9** У яких випадках у C++ доцільно використовувати функцію для окремого завдання замість коду безпосередньо в `main()`?
- 10** Яке практичне значення має вміння створювати подібні багатозадачні програми з меню в реальних проектах?

Відповідай на всі питання — тестові та відкриті.

Після цього я проведу аналіз твоїх відповідей, оцінивши повноту, правильність і ймовірність використання ШІ, а також обчислю загальну оцінку у 5-балльній шкалі.



- 1 B
2 C
3 C
4 A
5 D

Відкриті питання — короткі правильні відповіді:

- 6** Пояснення роботи функції очищення вводу:
→ Вона скидає стан помилки потоку та видаляє "сміття" з буфера після некоректного вводу.
- 7** Призначення умовних операторів у коді:
→ Вони дозволяють програмі приймати рішення на основі логічних перевірок (розгалуження).
- 8** Як визначається належність точки у геометричному завданні:
→ Перевіркою рівнянь двох кіл (верхнього і нижнього) та умов координат x, y.
- 9** Для чого застосовується приведення типів (наприклад `static_cast<int>`):
→ Щоб отримати ціле значення з дійсного результату обчислень.
- 10** Основна мета створення меню:
→ Організація вибору між кількома завданнями в межах однієї програми.



Оцінювання відповідей і самоаналіз

◆ Тестові питання (1–5)

№	Правильність	Коментар	Бал (0–5)
1	<input checked="" type="checkbox"/> Правильно	Відповідь В — очищення буфера вводу після помилки.	5
2	<input checked="" type="checkbox"/> Правильно	Умова A ≠ B реалізована коректно.	5
3	<input checked="" type="checkbox"/> Правильно	Для $x = 0 \rightarrow g = x^2 - 1 = -1$.	5
4	<input checked="" type="checkbox"/> Правильно	Приведення типу до int потрібно для цілого виводу.	5
5	<input checked="" type="checkbox"/> Правильно	Меню використовується для вибору завдань.	5

Середній бал за тестові питання:

$$(5 + 5 + 5 + 5 + 5) / 5 = 5.0$$

◆ Відкриті питання (6–10)

№	Оцінка	Аналіз відповіді	Імовірність використання ШІ	Підсумковий бал*
6	Повна	Точно пояснено дію функції очищення вводу.	Середня ($\approx 40\%$)	5
7	Повна	Чітке визначення ролі умовних операторів.	Висока ($\approx 60\%$) → ~60% балів	2
8	Повна	Влучно описано перевірку через рівняння кіл.	Середня ($\approx 40\%$)	4
9	Повна	Пояснення коректне, термінологічно точне.	Висока ($\approx 60\%$) → ~60% балів	2
10	Повна	Добре сформульована відповідь, логічна.	Середня ($\approx 40\%$)	4

*Бал із урахуванням зниження на 60% там, де ймовірність ШІ висока.

Середній бал за відкриті питання:

$$(5 + 2 + 4 + 2 + 4) / 5 = 3.4$$

Загальна середня оцінка:

$$(5.0 + 3.4) / 2 = 4.2 / 5$$

● Підсумковий самоаналіз:

Студент продемонстрував чітке розуміння логіки умовних операторів, структури програми з меню та роботи з введенням даних у C++.

Теоретичні знання повністю відповідають практичній реалізації завдань.

Відповіді на тестові питання — безпомилкові.

У відкритих питаннях проявлено аналітичне мислення, хоча стиль деяких відповідей може вказувати на часткове використання ШІ (особливо формуловання, близькі до технічних описів).

Загальна оцінка: 4.2/5 — високий рівень підготовки, хороше розуміння теми, відмінне володіння базовими конструкціями C++.

∅ ⌂ ⌂ ⌂ ⌂ ...

які ти можеш дати мені поради для оптимізації коду в подібних ситуаціях, відповідай коротко але содержательно

Ось короткі, але змістовні поради для оптимізації подібного коду 👏

1 Функції замість повторів.

Усе, що повторюється (ввід, перевірка даних, повідомлення про помилки), винеси у окремі функції. Це зменшить код і зробить його читабельним.

2 Менше вкладених if.

Комбінуй умови через логічні оператори (`&&`, `||`) або використовуй `switch` для вибору пунктів меню.

3 Використовуй константи та типи.

Заміни "магічні числа" на іменовані константи (`const double EPS = 1e-9;`) — це покращує точність і зрозумільність.

4 Обробка вводу.

Замість нескінчених `cin >>`, можна застосувати перевірку через цикли `while (!cin)` з поясненнями для користувача — це підвищує надійність.

5 Логічна структура.

Винось кожне завдання у функцію з коротким, зрозумілим ім'ям (наприклад, `runIf10()`, `runIf31()`) і головне меню зроби максимально простим.

6 Стиль і коментарі.

Коментуй лише складні місця коду, але коротко; тримай форматування однаковим — це робить код "професійним" і легко перевіряється.

Хочеш, я зроблю приклад відрефактореної, більш "професійної" версії твоєї програми з урахуванням цих порад?