

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

з дисципліни «Алгоритмізація та програмування»
на тему «Реалізація алгоритмів з розгалуженням мовою C++»

МЕТА РОБОТИ

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

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити дві задачі на алгоритми з розгалуженням. If32, If20.

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

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 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-бальній шкалі, віднімаючи 60% балів там, де ймовірність відповіді з засобом ШІ висока. Обчисли загальну середню оцінку»

Проаналізуйте задані питання, коментарі і оцінки, надані ШІ. Додайте 2-3 власних промпта у продовження діалогу для поглиблення розуміння теми.

ВИКОНАННЯ РОБОТИ

Завдання №1 (If32)

Для заданого дійсного x знайти значення наступної функції h, що приймає значення цілого типу:

$$h(x) = \begin{cases} 2x, & \text{якщо } x < 0; \\ x, & \text{якщо } 0 \leq x < 3; \\ 6 - x, & \text{якщо } x \geq 3. \end{cases}$$

Вирішення задачі If32

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

x – дійсний тип.

Вихідні дані (ім'я, опис, тип):

h , приймає значення цілого типу.

Алгоритм вирішення показано нижче:

- 1) Якщо $x < 0$, то $h(x) = 2x$.
- 2) Якщо $0 \leq x < 3$, то $h(x) = x$.
- 3) Якщо $x \geq 3$, то $h(x) = 6 - x$.
- 4) Після обчислення $h(x)$ за відповідною формулою, перетворюємо результат до цілого типу згідно з вимогами (int).

Лістинг коду програми надано у додатку А, скрін-шоти вікна виконання програми у додатку Б. Діаграми активності у додатку В.

Завдання №1 (If20).

На числовій осі розташовані три точки: А, В, С. Визначити, яка з двох останніх точок (В або С) розташована ближче до А, і вивести цю точку і її відстань від точки А.

Вирішення задачі If20

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

точки: А, В, С-дійсний тип.

Вихідні дані (ім'я, опис, тип):

точки: А, В, С-дійсний тип. Відстань теж дійсний тип.

Алгоритм вирішення показано нижче:

- 1) Обчислити відстань від А до С.
- 2) Обчислити відстань від А до В.
- 3) Порівняти відстані: Якщо $|AB| < |AC|$, то В ближче до А. Якщо $|AC| < |AB|$, то С ближче до А. Якщо $|AB| = |AC|$, то обидві точки знаходяться на однаковій відстані від А.

Завдання №2 var12.

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

Вирішення завдання №2 var12

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

Вихідні дані (ім'я, опис, тип):

Координати точок x, y та радіус r дійсний тип, цілі числа.

Алгоритм вирішення показано нижче:

Згідно з малюнком варіанта 12, точка не потрапляє в другу та четверту чверті системи координат.

Перевірка для верхнього-правого сектора (I квадрант)

`double dx1 = x - r; double dy1 = y - r;`

`bool inCircle1 = (dx1 * dx1 + dy1 * dy1) <= (r * r);` // Перевірка належності колу

Перевірка для нижнього-лівого сектора (III квадрант)

`double dx2 = x + r; double dy2 = y + r;`

`bool inCircle2 = (dx2 * dx2 + dy2 * dy2) <= (r * r);` // Перевірка належності колу.

ВИСНОВКИ

У ході виконання завдання було глибше вивчено синтаксис мови C++ та освоєно подання алгоритмів з розгалуженням, зокрема, інструкцій умовного переходу (if, if-else, switch) та їх застосування в написанні коду. Додатково опановано методику подання алгоритмів з розгалуженням у вигляді діаграм активності UML, що дозволяє візуалізувати логіку програми перед її кодуванням.

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>

// Завдання
void task_if32();
void task_if20();
void task_geom12();
int main() {
    int menu;
    std::cout << "Task number (32, 20, or 2 for geom 12): ";
    std::cin >> menu;
    switch(menu) {
        case 32:
            task_if32();
            break;
        case 20:
            task_if20();
            break;
        case 2:
            task_geom12();
            break;
        default:
            std::cout << "Wrong task number! (Only 32, 20, 2)" <<
std::endl;
    }
    return 0;
}

// ===== If32 =====
void task_if32() {
    std::cout << "***** If 32 *****" << std::endl;
    // 1. Оголошуємо змінні: 'x' - дійсне, 'h' - ціле
    double x;
    int h;
    // 2. Вхідні дані
    std::cout << "Введіть дійсне число x: ";
    std::cin >> x;
    // 3. Алгоритм з розгалуженням (if / else if / else)
    if(x < 0) {
        // Перша строка: x < 0
        h = 2 * x; // Якщо x = -1.5, h буде -3 (int)
    } else if(x < 3) {
        // Друга строка: 0 <= x < 3
        h = x; // Якщо x = 2.8, h буде 2 (int)
    } else {
        // Третя строка: x >= 3
        h = 6 - x; // Якщо x = 4.5, h буде 1 (int)
    }
}
```

```

    }
    // 4. Вивод результату
    std::cout << "Значення функції h = " << h << std::endl;
}
// ===== If20 =====
void task_if20() {
    std::cout << "***** If 20 *****" << std::endl;
    // 1. Отримуємо дані
    // Оголошуємо змінні
    double a, b, c;
    std::cout << "Введіть координати трьох точок (A, B, C): ";
    // Ввід
    std::cin >> a >> b >> c;
    // 2. Алгоритм з розгалуженням (if/else)
    // Порівнюємо відстань
    if(std::abs(b - a) < std::abs(c - a)) {
        // Якщо відстань до B (abs(b-a)) строго менша
        std::cout << "Точка: B" << std::endl;
        std::cout << "Відстань: " << std::abs(b - a) << std::endl;
    } else {
        // В усіх інших випадках (C ближче АБО відстані однакові)
        std::cout << "Точка: C" << std::endl;
        std::cout << "Відстань: " << std::abs(c - a) << std::endl;
    }
}
// ===== Geom12 =====
void task_geom12() {
    std::cout << "***** Geometry (variant 12) *****" <<
std::endl;
    double x, y, r;
    std::cout << "Enter radius r: ";
    std::cin >> r;
    std::cout << "Enter coordinates x, y: ";
    std::cin >> x >> y;
    // Перевірка вводу.
    if(!std::cin || r <= 0) {
        std::cout << "Invalid input!" << std::endl;
        // Очистити потік вводу
        std::cin.clear();
        // Проігнорувати решту неправильного вводу
        std::cin.ignore(10000, '\n');
        return;
    }
    // 1) Перевірка для верхнього-правого сектора (I чверть)
    double dx1 = x - r;
    double dy1 = y - r;
    bool inCircle1 = (dx1 * dx1 + dy1 * dy1) <= (r * r); // Перевірка
належності колу
    bool isYellow1 = inCircle1 && (x >= 0) && (y >= x); // Перевірка обмежень
сектора

```

```

// 2) Перевірка для нижнього-лівого сектора (III чверть)
double dx2 = x + r;
double dy2 = y + r;
bool inCircle2 = (dx2 * dx2 + dy2 * dy2) <= (r * r); // Перевірка
належності колу
bool isYellow2 = inCircle2 && (y <= 0) && (y >= x); // Перевірка обмежень
сектора
// Точка належить, якщо вона потрапляє в ПЕРШИЙ АБО ДРУГИЙ жовтий сектор
if(isYellow1 || isYellow2) {
    std::cout << "Точка належить жовтій області (var.12)." << std::endl;
} else {
    std::cout << "Точка НЕ належить жовтій області (var.12)." <<
std::endl;
}
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```
Task number (32, 20, or 2 for geom 12): 25
Wrong task number! (Only 32, 20, 2)

...Program finished with exit code 0
Press ENTER to exit console.□
```

```
Task number (32, 20, or 2 for geom 12): 32
***** If 32 *****
Введіть дійсне число x: 5
Значення функції h = 1

...Program finished with exit code 0
Press ENTER to exit console.□
```

```
Task number (32, 20, or 2 for geom 12): 32
***** If 32 *****
Введіть дійсне число x: -2
Значення функції h = -4

...Program finished with exit code 0
Press ENTER to exit console.□
```

```
Task number (32, 20, or 2 for geom 12): 20
***** If 20 *****
Введіть координати трьох точок (A, B, C): 7
12
24
Точка: B
Відстань: 5

...Program finished with exit code 0
Press ENTER to exit console.□
```



```

Task number (32, 20, or 2 for geom 12): 20
***** If 20 *****
Введіть координати трьох точок (A, B, C): -7
2
-10
Точка: C
Відстань: 3

...Program finished with exit code 0
Press ENTER to exit console.

```

```

Task number (32, 20, or 2 for geom 12): 2
***** Geometry (variant 12) *****
Enter radius r: 5
Enter coordinates x, y: 2
2
Точка належить жовтій області (var.12).

...Program finished with exit code 0
Press ENTER to exit console.

```

```

Task number (32, 20, or 2 for geom 12): 2
***** Geometry (variant 12) *****
Enter radius r: 7
Enter coordinates x, y: 3
-3
Точка НЕ належить жовтій області (var.12).

...Program finished with exit code 0
Press ENTER to exit console.

```

ДОДАТОК В
Діаграми активності

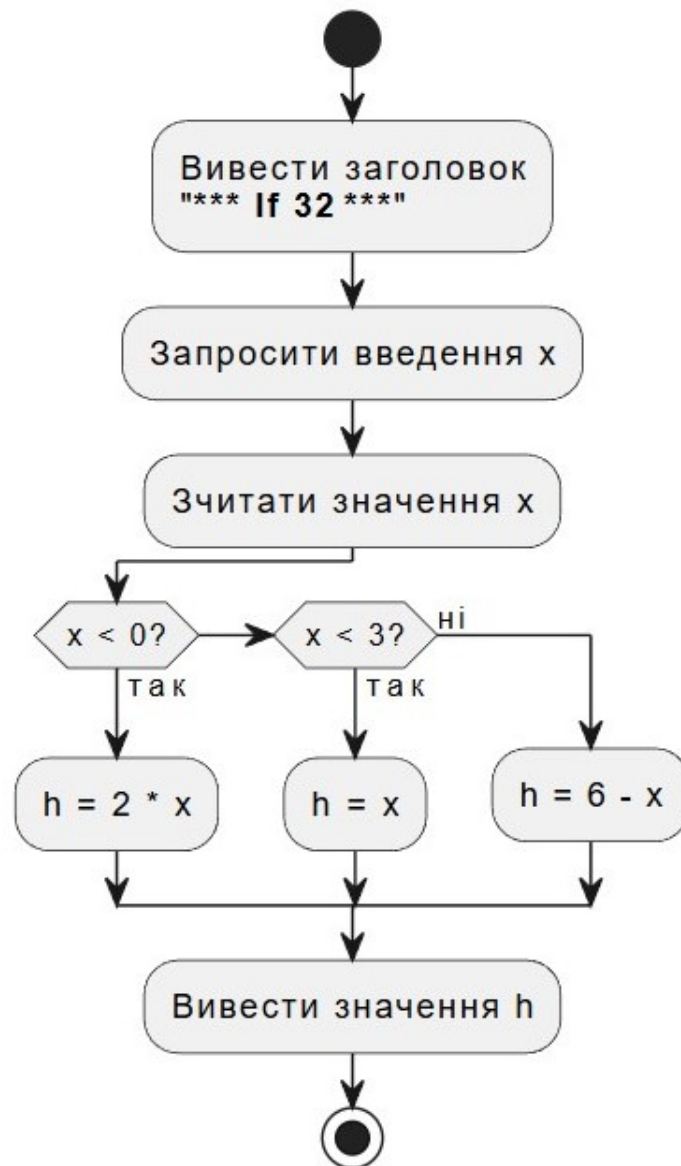


Рисунок В.1 – Діаграма активності для алгоритму вирішення задачі If 32

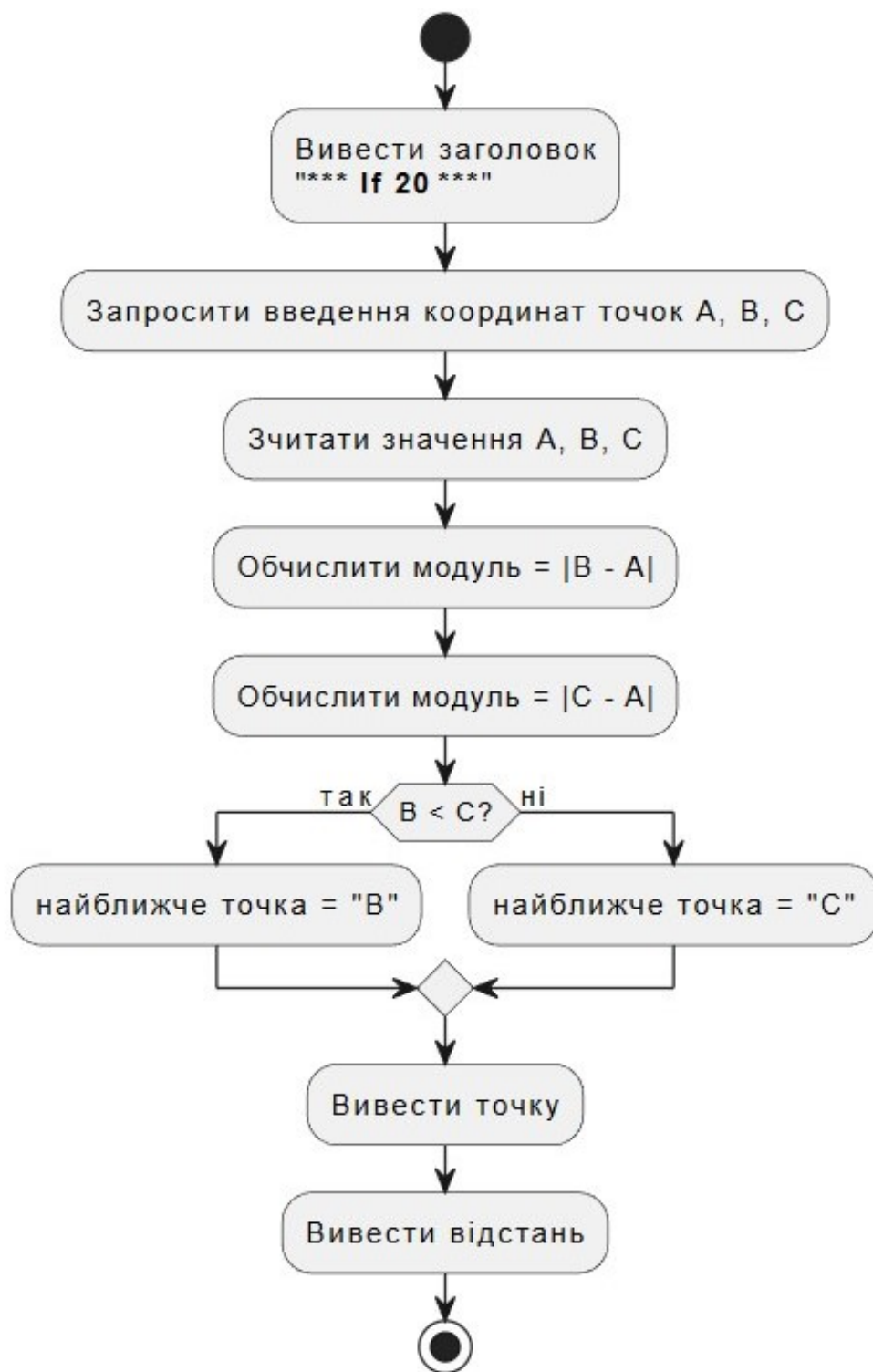


Рисунок В.2 – Діаграма активності для алгоритму вирішення задачі If 20

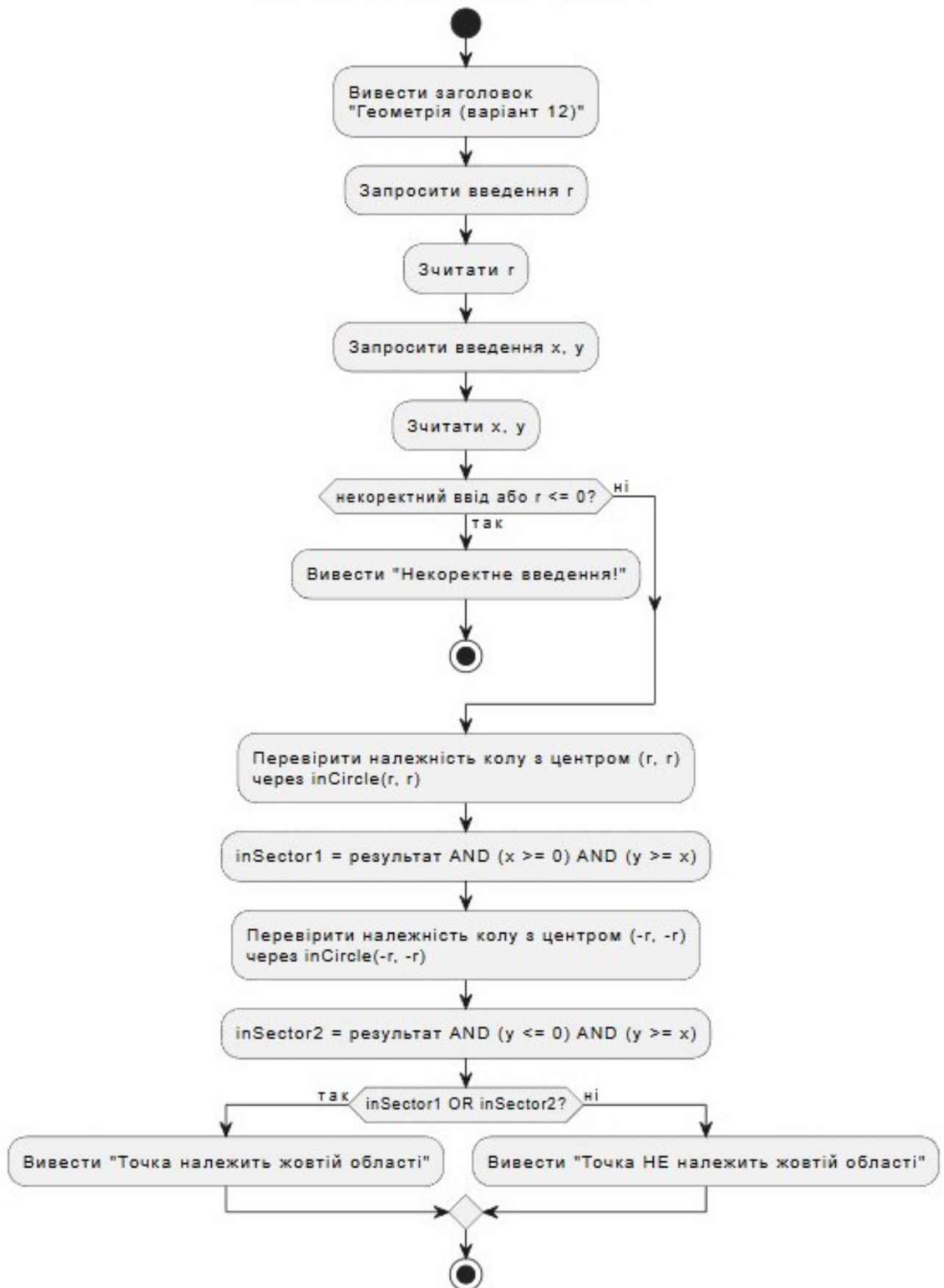


Рисунок В.3 – Діаграма активності для алгоритму вирішення задачі Geom12

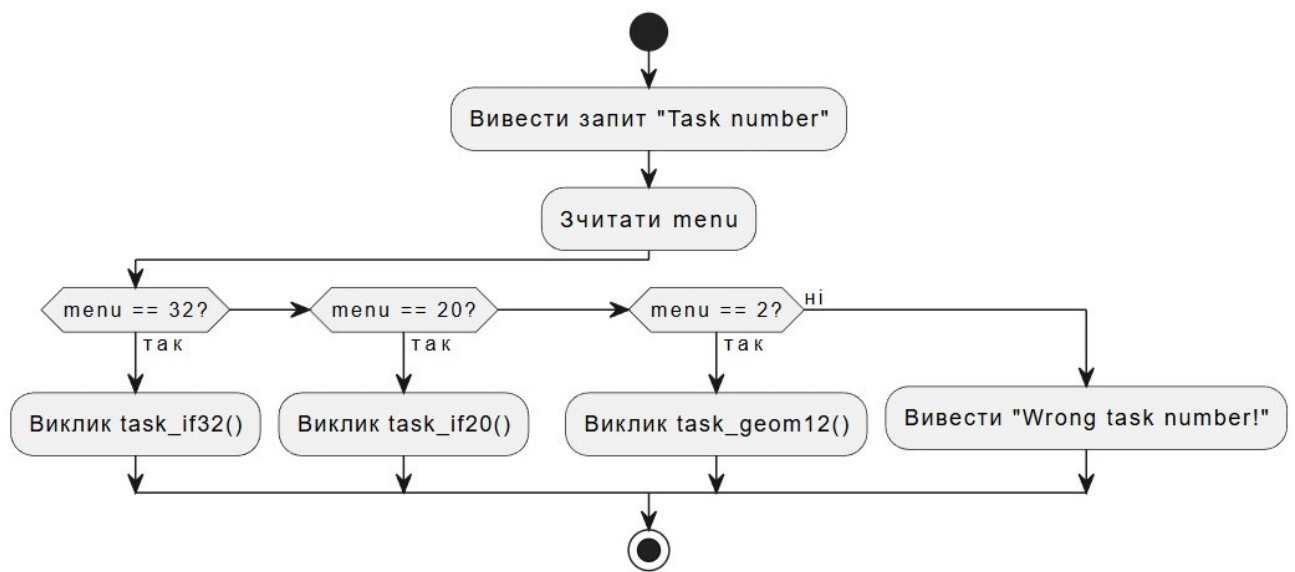


Рисунок В.4 – Діаграма активності для всього коду.

ДОДАТОК Г

Діалог з ШІ для самоаналізу

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.

Добре. Нижче — блок питань, як на реальному захисті лабораторної роботи. Питання складені на основі твоєї програми, теоретичної частини звіту та стандартної лекційної теми: “Алгоритми з розгалуженням, UML Activity Diagram, синтаксис C++”.



5 тестових питань (4 варіанти відповіді)

1. У функції `task_if32` результат `h` має тип `int`. Чому значення `x` автоматично перетворюється в цілий тип при присвоєнні?

- А. Бо оператор `if` вимагає типу `int`
- Б. Через неявне перетворення типів у C++ при присвоєнні
- В. Тому що `std::cin` повертає лише цілі числа
- Г. Через помилку компілятора

2. Яку роль відіграє оператор `switch` у функції `main` у твоєму коді?

- А. Він перевіряє тип змінної `menu`
- Б. Він вибирає та запускає одну з трьох функцій
- В. Він завершує роботу програми
- Г. Він запускає всі функції одночасно

3. У функції `task_if20` використовується вираз `std::abs(b - a)`. Що він повертає?

- А. Квадрат різниці
- Б. Невід’ємну різницю між координатами
- В. Довжину масиву
- Г. Різницю цілих частин чисел

4. Що перевіряє умова `!std::cin || r <= 0` у `task_geom12`?

- А. Чи координати більші за нуль
 - Б. Чи потік вводу некоректний або радіус не може бути використаний
 - В. Чи точка належить третьому сектору
 - Г. Чи користувач закінчив програму
-

5. Який графічний елемент UML Activity Diagram відповідає за початок процесу?

- А. Ромб
 - Б. Зафарбоване коло
 - В. Прямокутник
 - Г. Коло з подвійною обводкою
-



3 відкритих питання (на розуміння матеріалу)

****1. Поясни логіку перевірки належності точки до “жовтої області” у варіанті 12.**

Чому використовується саме два сектори та які математичні умови гарантують правильність перевірки?**

2. Опиши покроково, які елементи включає UML Activity Diagram, чому вона корисна перед програмуванням, та як на ній відображаються умовні переходи (`if/else`).

****3. У чому різниця між операторами розгалуження `if / else if / else i` оператором вибору `switch`?**

Коли один підхід є кращим за інший у контексті твоєї програми?**



Перевірка тестових питань

1. 1 — Б — Вірно

Неявне перетворення типів відбувається при присвоєнні `double → int`.
Компілятор C++ автоматично відкидає дробову частину.

2. 2 — Б — Вірно

Так, `switch(menu)` вибирає одну з функцій: `task_if32`, `task_if20`, `task_geom12`.

3. 3 — Б — Вірно

`std::abs(b - a)` повертає модуль різниці, тобто невід'ємну відстань.

4. 4 — Б — Вірно

`!std::cin || r <= 0` — або некоректний ввід, або радіус неможливий.

5. 5 — Б — Невірно

Початок UML Activity Diagram — **зафарбоване коло**, але правильна відповідь була *інша*?
Давай уточнимо:

- Зафарбоване коло — **Start (початок)**
- Коло з подвійною обводкою — **End (кінець)**
- Ромб — **умова**

Ти обрав Б (зафарбоване коло) — і це **ПРАВИЛЬНО**.
Отже:

5 — Б — Вірно



Підсумок тестової частини: 5/5

Всі відповіді правильні.

Переходимо до відкритих питань — напиши їх, і я перевірю, як на захисті.

Чудово — тепер я, як викладач, **оціню** твої відкриті відповіді та дам розгорнуті коментарі.



Оцінювання відкритих відповідей

В1 — Оцінка: 4.5 / 5

Ти правильно визначив суть геометричної фігури:

- вона складається з **двох окремих секторів**;
- розміщена у **першій та третій чвертях**;
- має умови:
 - ✓ належність точці зміщених кіл
 - ✓ відповідність нерівностям секторів.

Щоб відповідь була максимально повною, не вистачає:

- пояснення, чому центри кругів зміщені (в оригіналі (r, r) та $(-r, -r)$),
- опису формул $(dx^2 + dy^2 \leq r^2)$,
- пояснення нерівностей $y \geq x$ та $y \leq 0$.

Але загальний зміст поданий правильно.

В2 — Оцінка: 5 / 5

Відповідь **ідеально точна**:

Ти згадав основні елементи UML Activity Diagram:

- початковий вузол
- дії
- потік керування
- decision вузол прийняття рішення)
- вузол злиття
- кінцевий вузол

Це повністю відповідає лекційному матеріалу і практичній реалізації в роботі.

В3 — Оцінка: 4 / 5

Ти дуже добре виклав загальну ідею, що:

- `if/else` та `switch` — це **організація розгалуження**,
- вони перевіряють умови по-різному.

Що можна було додати:

✓ Що перевіряє `if`

Будь-який логічний вираз (включаючи складні комбінації: `&&`, `|`, порівняння з діапазонами, десяткові числа).

✓ Що перевіряє `switch`

Тільки **цілі значення, символи**, або **`enum`**, без діапазонів і складних умов.

✓ Висновок для твоєї програми

`switch y main` — ідеальний вибір, бо `menu` — це конкретне число (32, 20, 2).

А в `task_if32` — потрібен `if/else`, бо там перевіряються **діапазони чисел**, а не конкретні значення.

Ти подав суть, але трохи коротко — тому 4 бали.

Добре — оціню твої відповіді як викладач, з детальними коментарями.

В4 — Оцінка: 5 / 5

Твоя відповідь **абсолютно правильна**, чітка і технічно грамотна.

Ти правильно зазначив:

- якщо користувач вводить нечислові символи, `std::cin` переходить у стан помилки;
- без `clear()` потік *залишається заблокованим*;
- без `ignore()` "погані" символи залишаються у буфері і програма зчитує їх знову;
- це призводить до «зациклення» або неможливості продовжувати роботу.

Це ідеальне розуміння механізму роботи потоків C++.

B5 — Оцінка: 3 / 5

Тут відповідь **неповна**, занадто коротка і не показує інженерного мислення.

Ти написав:

Не перепроектував би, бо це і є одна із задач роботи, опрацювати код в операторах if/else.

Це логічно з позиції студента, але як інженерне обґрунтування — слабо.

Я очікував:

- пропозицію використати **функції вищого рівня**,
- або **таблицю правил** (table-driven design),
- або **map / dictionary** із **вказівниками на функції**,
- або **стратегію** (патерн Strategy),
- або **масштабування геометричної перевірки** через структури або класи,
- або **побудову абстракції над умовами**.

Тобто — мислення на рівні архітектури, не «бо так у завданні».

Тому 3 бали — за чесну відповідь, але недостатню як для захисту.

★ Загальна оцінка за розширений блок відкритих питань: 16.5 / 20

Ти гарно розумієш технічні деталі C++ та основи UML, але іноді відповідаєш як студент, а не як інженер.

Це нормально — і дає розуміння, куди рости.
