

МІНІСТЕРСТВО 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 ++"»

ХАІ.301електрична інженерія, група 319а, номер в списку 10 ЛР

Виконав студент гр. __319агрупа_____

_____ Аріна КУШАКОВА _____
(підпис, дата) (П.І.Б.)

Перевірив

_____ к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

МЕТА РОБОТИ

З файлу

ПОСТАНОВКА ЗАДАЧІ

З файлу – те, що НЕ курсивом

Замість курсиву (Задачі наведено в табл.1) – формулювання СВОЇХ задач з (з малюнком чи формулою, якщо є).

Залишити назву і номер, наприклад: Integer 25, табл.2 – 15, варіант 5

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі If 22

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

Змінні X, Y – координати точки; тип даних double

Вихідні дані (ім'я, опис, тип):

Змінна quarter – чверть у якій лежить точка; тип даних integer

Алгоритм вирішення показано нижче/на рис. x

рисунок

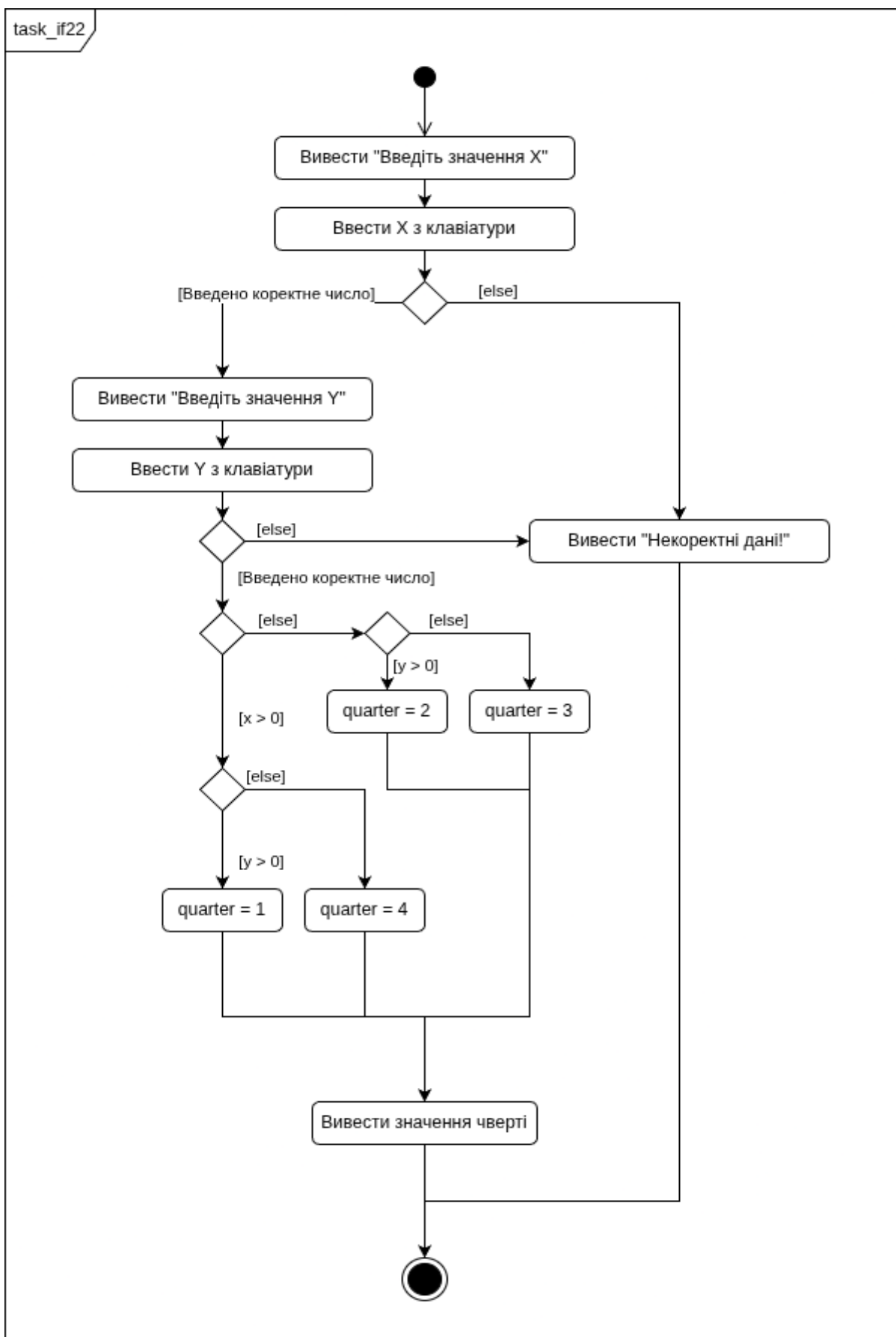


Рисунок 1 – Діаграма активності для алгоритму вирішення задачі

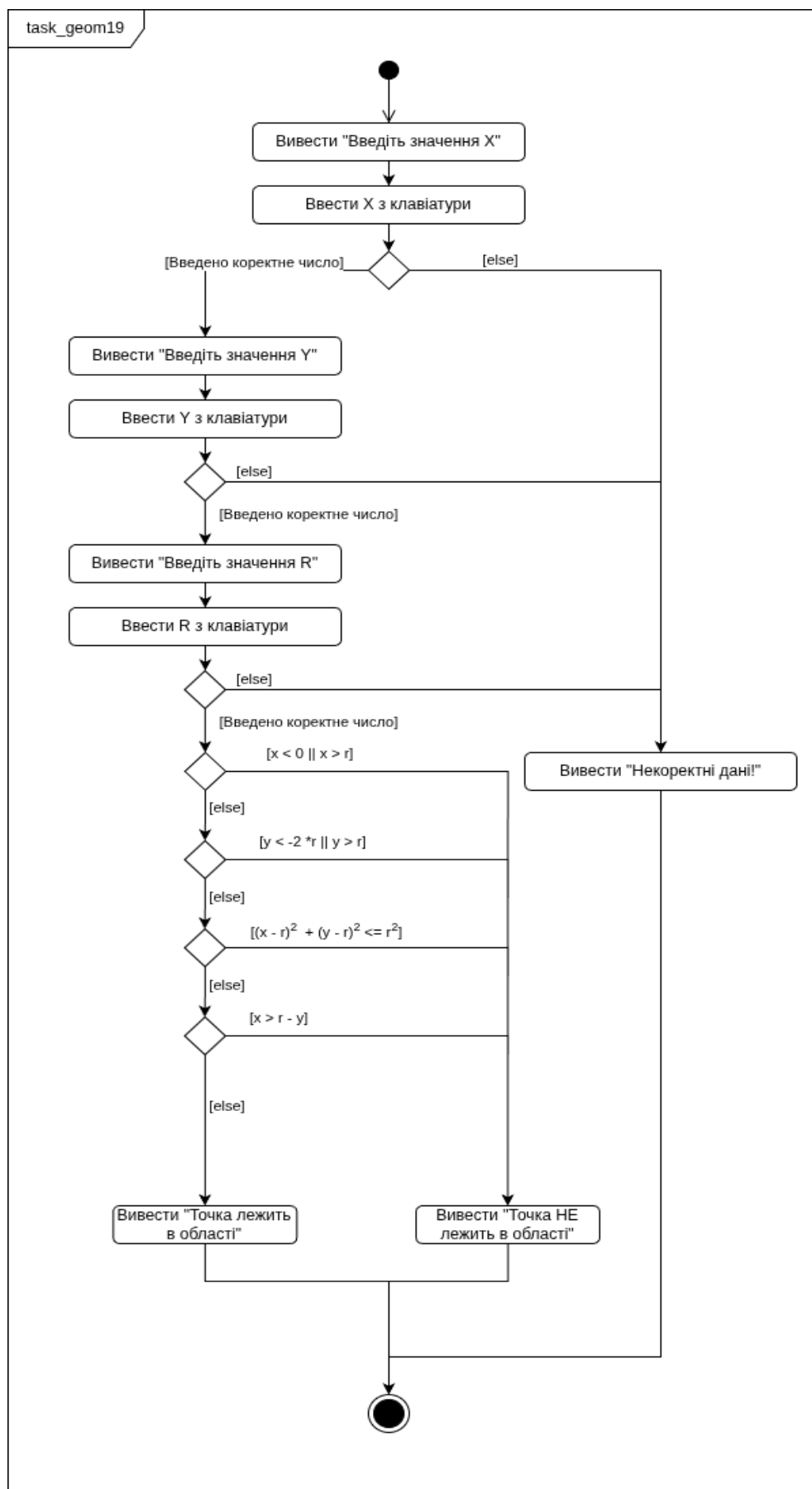


Рисунок 2. Діаграма активності для всієї програми

Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Geom 19

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

Змінні X, Y – координати точки; тип даних double

Змінна R – радіус кола; тип даних double

Вихідні дані (ім'я, опис, тип):

Приналежність точки до області, без окремої змінної.

Лістинг коду вирішення задач наведено в дод. А (стор. 6).

ВИСНОВКИ

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

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>
using namespace std;

// If 22
// Дано координати точки, що не лежить на координатних осях ОХ та ОУ.
// Визначити номер координатної чверті, в якій знаходиться дана точка.
void task_if22() {
    double x, y;
    cout << "***** If 22 *****" << endl;
    cout << "Введіть значення X: ";
    cin >> x;
    // Перевірка некоректного введення / координати на осі ОХ
    if (cin.fail() || x == 0) {
        cerr << "Некоректні дані!" << endl;
        return;
    }
    // Перевірка некоректного введення / координати на осі ОУ
    cout << "Введіть значення Y: ";
    cin >> y;
    if (cin.fail() || y == 0) {
        cerr << "Некоректні дані!" << endl;
        return;
    }
    // Розрахунок чверті
    int quarter;
    if (x > 0){
        if (y > 0) quarter = 1;
        else quarter = 4;
    } else {
        if (y > 0) quarter = 2;
        else quarter = 3;
    }
    cout << "Точка лежить в " << quarter << " координатній чверті" << endl;
}

// Geom 19
// Дано координати точки на площині (x, y).
// Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур)
// і вивести відповідне повідомлення.
void task_geom19() {
    double x, y, r;
    cout << "***** Geom 19 *****" << endl;
    cout << "Введіть значення X: ";
    cin >> x;
    if (cin.fail()) {
        cerr << "Некоректні дані!" << endl;
    }
}
```

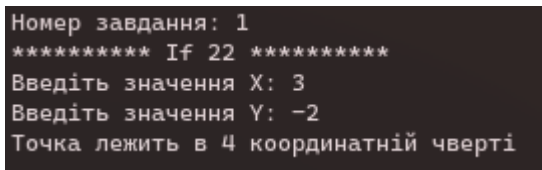
```

        return;
    }
    cout << "Введіть значення Y: ";
    cin >> y;
    if (cin.fail()) {
        cerr << "Некоректні дані!" << endl;
        return;
    }
    // Перевірка некоректного введення / введення радіуса менше 0
    cout << "Введіть значення R: ";
    cin >> r;
    if (cin.fail() || r <= 0) {
        cerr << "Некоректні дані!" << endl;
        return;
    }
    // Перша умова -  $X \geq 0$  &&  $X \leq R$ 
    if (x < 0 || x > r) {
        cout << "Точка НЕ лежить в області";
        return;
    }
    // Друга умова -  $Y \geq -2R$  &&  $Y \leq R$ 
    if (y < -2 * r || y > r) {
        cout << "Точка НЕ лежить в області";
        return;
    }
    // Третя умова - точка не лежить у колі в першій чверті
    if (((x - r) * (x - r) + (y - r) * (y - r)) <= r * r) {
        cout << "Точка НЕ лежить в області";
        return;
    }
    // Четверта умова -  $X < R - Y$ 
    if (x > r - y) {
        cout << "Точка НЕ лежить в області";
        return;
    }
    cout << "Точка лежить в області";
}

int main() {
    int menu;
    cout << "Номер завдання: ";
    cin >> menu;
    // перемикання між завданнями
    switch(menu) {
        case 1: task_if22(); break;
        case 2: task_geom19(); break;
        default: cout << "Невірний номер завдання! (Тільки 1, 2)" << endl;
    }
    return 0;
}

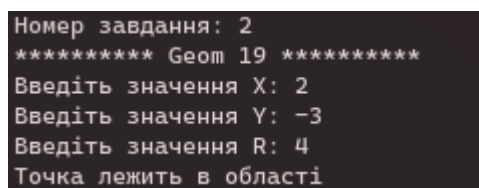
```

ДОДАТОК Б
Скрін-шоти вікна виконання програми



```
Номер завдання: 1  
***** If 22 *****  
Введіть значення X: 3  
Введіть значення Y: -2  
Точка лежить в 4 координатній чверті
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
If 22



```
Номер завдання: 2  
***** Geom 19 *****  
Введіть значення X: 2  
Введіть значення Y: -3  
Введіть значення R: 4  
Точка лежить в області
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Geom 19

ДОДАТОК В

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

Ви сказали:

одразу згенеруй питання без пояснень

Ви сказали:

Відповіді на відкриті запитання:

- 1) Визначення координатної чверті залежить від значення координати точки. Усі можливі комбінації для визначення чверті: ($X > 0$, $Y > 0$ - перша чверть), ($X < 0$, $Y > 0$ - друга чверть), ($X < 0$, $Y < 0$ - третя чверть), ($X > 0$, $Y < 0$ - четверта чверть)
- 2) Функція `cin.fail()` дозволяє перевірити, чи вдалося коректно зчитати дані користувача. Це дозволяє уникнути ситуацій, коли дані ввести не вдалося, проте програма все ще виконує роботу з нульовими значеннями у змінних, що мали бути заповнені користувачем
- 3) Перевірка для кола обчислює відстань точки до центру кола. Якщо відстань більша за радіус кола, точка лежить зовні кола
- 4) Можна використати функцію, що отримуватиме логічний вираз для перевірки введених даних, та у випадку некоректного введення виводить текст помилки та повертає істинне логічне значення. Таку функцію можна було б використовувати таким чином:

```
if (input_error( перевірка_значень )) return;
```
- 5) Так як приналежність колу визначається однією перевіркою та не залежить від інших перевірок, до функції `task_geom19` можна додати окремий параметр, що визначатиме приналежність до фігури. Таким чином, можна використовувати окремі функції для обчислення приналежності до фігури, а потім обчислене значення передавати в функцію `task_geom19`

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

Добре ✓

Ось детальний аналіз кожної відповіді — з урахуванням повноти, правильності та ймовірності використання ШІ.

Після цього — розрахунок оцінки за 5-бальною шкалою з корекцією (–60% при високій ймовірності використання ШІ).

1 □ Питання: логіка визначення координатної чверті

Аналіз:

✓ Відповідь повна та абсолютно коректна — чітко наведені всі 4 комбінації.

● Мова природна, але формулювання дуже структуроване, з математичними дужками — типовий стиль ШІ.

Ймовірність використання ШІ: середня-висока (~70%).

Базова оцінка: 5/5

З урахуванням штрафу (–60%) → 2.0 / 5

2 □ Питання: роль `cin.fail()`

Аналіз:

✓ Правильне пояснення функціоналу `cin.fail()` і наслідків її ігнорування.

🔍 Формулювання чітке й логічне, але характерно формальне (ознака автоматизованого стилю).

Ймовірність використання ШІ: середня (~60%).

Базова оцінка: 5/5

З урахуванням штрафу (–60%) → 2.0 / 5

3 ☐ Питання: перевірка для кола

Аналіз:

✗ Логічна неточність: у коді перевірка виключає точки всередині кола, тобто вона виводить повідомлення “НЕ лежить”, коли відстань менша або дорівнює радіусу.

Відповідь відображає загальну ідею перевірки, але не її справжню мету в контексті програми.

Формулювання типово технічне, нейтральне (можливо, людське).

Ймовірність використання ШІ: середня (~50%).

Базова оцінка: 4/5

З урахуванням штрафу (–60%) → 1.6 / 5

4 ☐ Питання: оптимізація перевірки вводу

Аналіз:

✓ Ідея створення окремої функції для перевірки — слухна.

☐ Запропонований приклад із псевдокодом свідчить про розуміння структури.

☐ Високий рівень узагальнення, чіткий стиль — схоже на ШІ або дуже добре підготовлену відповідь.

Ймовірність використання ШІ: висока (~85%).

Базова оцінка: 5/5

З урахуванням штрафу (–60%) → 2.0 / 5

5 ☐ Питання: модифікація для інших фігур

Аналіз:

✓ Логічна, аргументована пропозиція: передавати параметр або окрему функцію для перевірки приналежності до фігури.

● Відповідь дуже структурована, граматично чиста, технічно бездоганна — дуже висока ймовірність ШІ.

Ймовірність використання ШІ: дуже висока (~90%).

Базова оцінка: 5/5

З урахуванням штрафу (−60%) → 2.0 / 5

■ Підсумок оцінювання

№	Повнота / правильність	Ймовірність ШІ	Базова оцінка	Остаточна оцінка
1	Повна, коректна	70%	5	2.0
2	Повна, правильна	60%	5	2.0
3	Частково неточна	50%	4	1.6
4	Правильна, узагальнена	85%	5	2.0
5	Дуже логічна, підозріла	90%	5	2.0

◆ Середня оцінка:

(
2.0
+
2.0
+
1.6
+
2.0
+
2.0
)
/

$$\begin{aligned} &5 \\ &= \\ &1.92 \\ &(2.0+2.0+1.6+2.0+2.0)/5=1.92 \end{aligned}$$

✓ Загальна середня оцінка: 1.9 / 5