

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«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. 175. 318. 02 ЛР

Виконав студент гр. _____ 318

_____ Бондаренко С. Б.
(підпис, дата) (П.І.Б.)

Перевірив
_____ к.т.н., доц. Гавриленко О. В.
(підпис, дата) (П.І.Б.)

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 Visual Studio. Також опанувати та відпрацювати навички структурування програми з функціями.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на алгоритми з розгалуженням.

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. Обчислити площу і периметр плоскої фігури.

Завдання 4. Для вибору користувачем одного з трьох зазначених вище завдань розробити алгоритм організації меню в командному вікні з використанням інструкції вибору.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі If 28

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

year — рік для перевірки на високосність, int, > 0, числове значення

Вихідні дані (ім'я, опис, тип):

result — результат перевірки року: високосний чи ні, рядок.

На рис. 1 показаний приклад діаграми для завдання If 28.

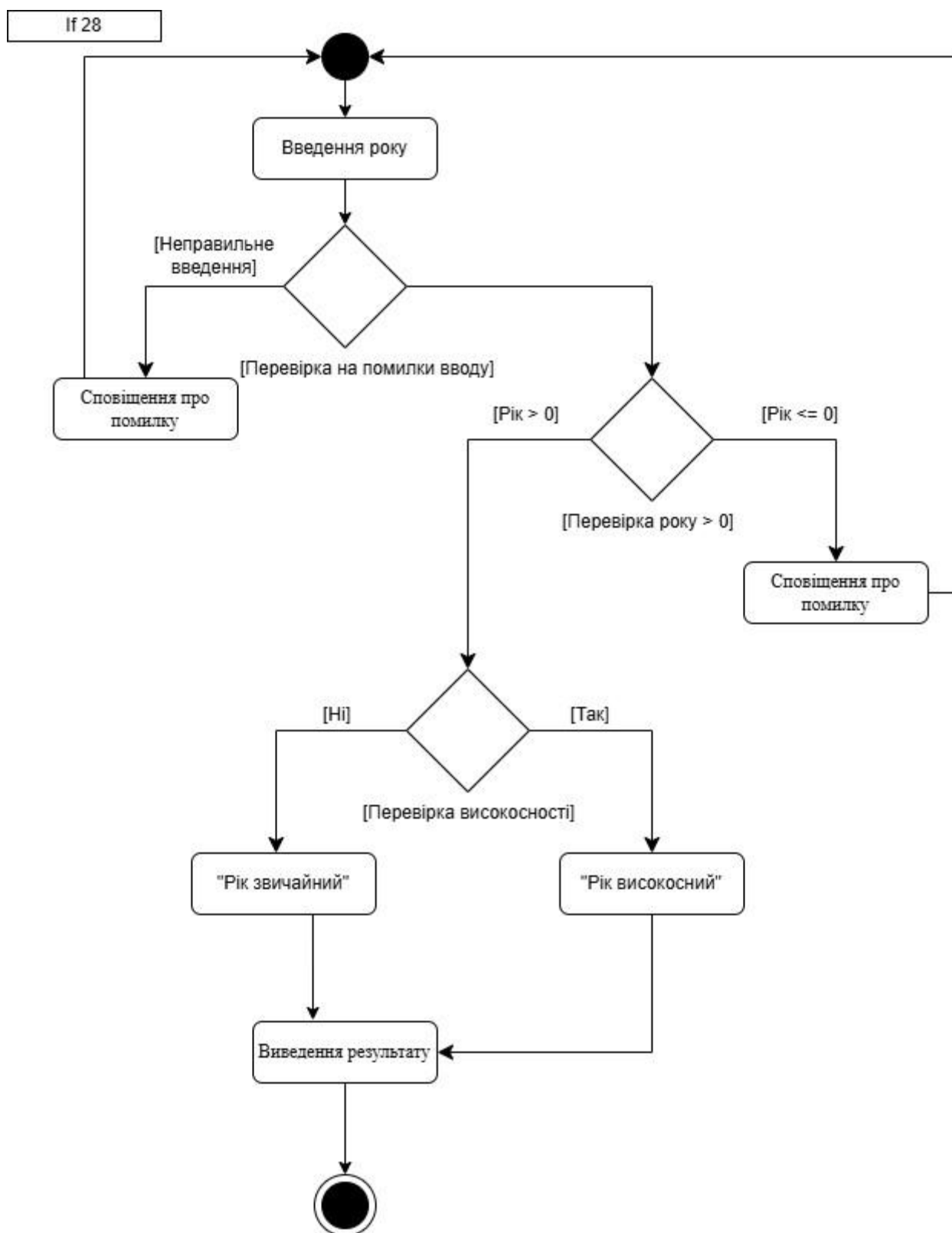


Рисунок 1 – Діаграма для завдання If 28

Лістинг коду вирішення задачі If 28 наведено в дод. А (стор. 8).
Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Geom 10.

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

x — координата x точки, double, довільне дійсне число.

y — координата y точки, double, довільне дійсне число.

a — сторона квадрата, double, > 0 , числове значення.

Вихідні дані (ім'я, опис, тип):

result — результат перевірки: точка належить чи не належить до коричневої області, рядок

На рис. 2 показаний приклад діаграми для завдання Geom 10.

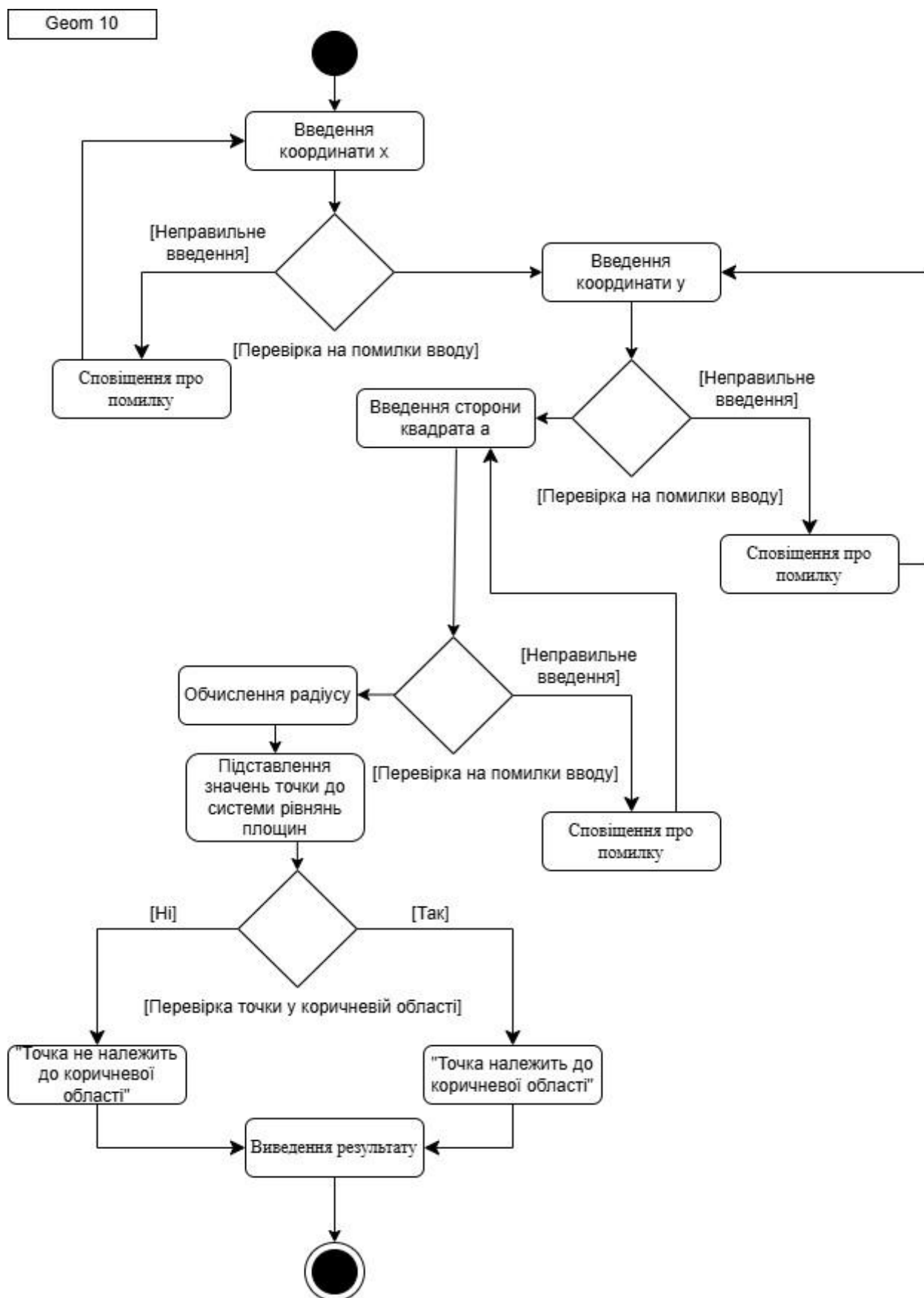


Рисунок 2 – Діаграма для завдання Geom 10

Лістинг коду вирішення задачі Geom 10 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.2.

Завдання 3.

Вирішення задачі Geom 10.

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

a — сторона квадрата, double, > 0, числове значення.

Вихідні дані (ім'я, опис, тип):

area — площа коричневої області, double.

perimeter — периметр коричневої області, double.

На рис. 3 показаний приклад діаграми для завдання Geom 10.

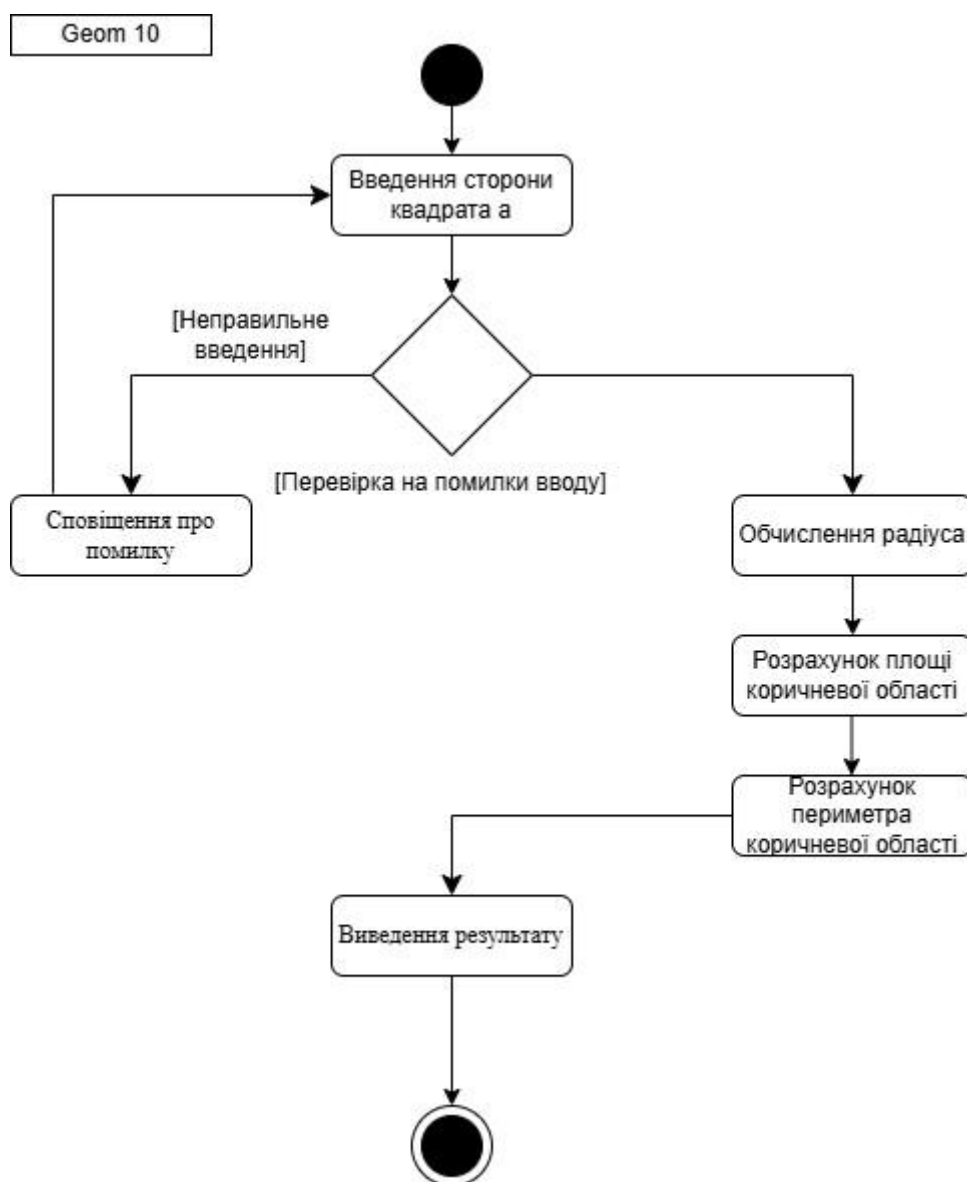


Рисунок 3 – Діаграма для завдання 3

Лістинг коду вирішення задачі Geom 10 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.3.

Завдання 4. Організація меню.

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

choice – номер завдання, обраний користувачем, int, від 1 до 3 включно, числове значення.

Вихідні дані (ім'я, опис, тип):

result – результат виконання обраного завдання, рядок або числа (залежно від завдання).

На рис. 4 показаний приклад діаграми для завдання 4

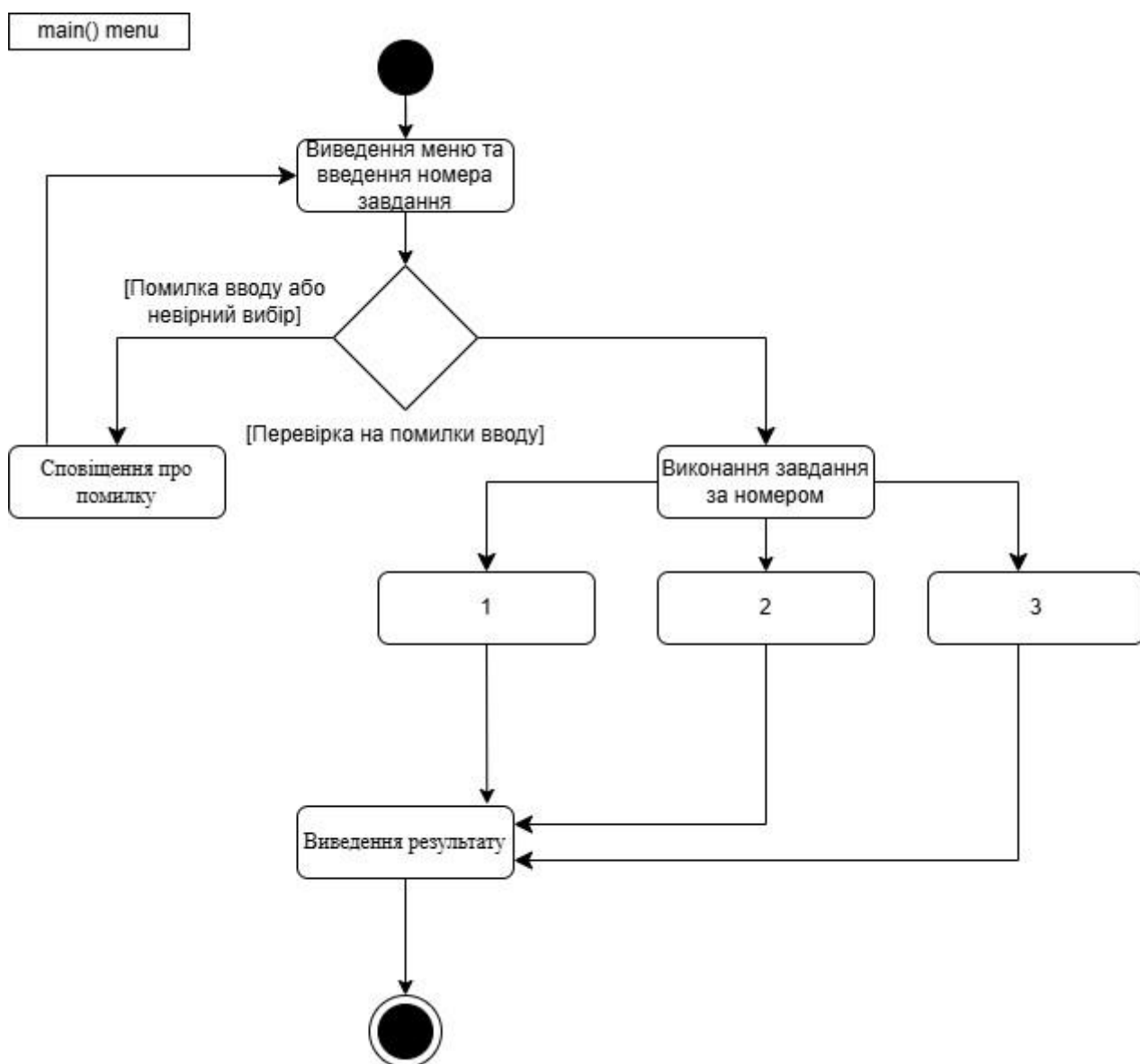


Рисунок 4 – Діаграма для завдання 4

ВИСНОВКИ

Розглянуто синтаксис мови C++ та визначено практичне представлення алгоритмів з розгалуженням у вигляді діаграм активності UML.

Розроблено структуру програми з функціями та доступом до меню.

ДОДАТОК А

Лістинг коду програми

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

// Перевірка високосного року
bool isLeapYear(int year) {
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
        return true;
    }
    return false;
}

void checkLeapYear() {
    Step1:
    cout << "\nПеревірка високосного року\n";
    cout << "Введіть номер року: ";
    int year;
    cin >> year;

    if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного від користувача числа
        cout << "\n\"year\" повинно бути числом\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    if (year <= 0) {
        cout << "\n\"year\" має бути більше нуля\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    if (isLeapYear(year)) {
        cout << "Рік " << year << " є високосним і має 366 днів.\n";
    }
    else {
        cout << "Рік " << year << " є звичайним і має 365 днів.\n";
    }
}
```



```

    }
}

double inputSideOfSquare() {
    Step1:
    cout << "Введіть сторону квадрата a: ";
    double a;
    cin >> a;

    if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного від користувача числа
        cout << "\n\"a\" повинно бути числом\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    if (a <= 0) {
        cout << "\n\"a\" має бути більше нуля\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    return a;
}

double inputCoordinate(char nameOfAxis) {
    Step1:
    cout << "Введіть координату " << nameOfAxis << ": ";
    double coordinate;
    cin >> coordinate;

    if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного від користувача числа
        cout << "\n\"" << nameOfAxis << "\" повинно бути числом\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    return coordinate;
}

// Функція перевірки точки в коричневій області
bool isPointInBrownArea(double x, double y, double a) {
    double r = a / 2.0; // Радіус кола

    // Перевірка для for all sectors
    bool inFirstBrownSector = (x * x + y * y <= r * r && x <= 0 && y <= x); //
    перевірка на наявність точки в секторі кола коричневої області
    bool inSecondBrownSector = (x * x + y * y >= r * r && x >= -r && y >= -r && y <=

```

```

0 && y >= x); // перевірка на наявність точки у маленькому шматочку коричневої
області зліва вниз
    bool inThirdBrownSector = (x * x + y * y >= r * r && x <= r && y <= r && y >= 0
&& y <= x); // перевірка на наявність точки у маленькому шматочку коричневої області
праворуч угорі
    return inFirstBrownSector || inSecondBrownSector || inThirdBrownSector; //
Повертає true, якщо точка буде в одному з коричневих секторів

}

void pointInBrownAreaCheck() {
    double x, y, a;
    x = inputCoordinate('x');
    y = inputCoordinate('y');
    a = inputSideOfSquare();

    if (isPointInBrownArea(x, y, a)) cout << "Точка належить до коричневої області";
    else cout << "Точка не належить до коричневої області";
}

// Розрахунок площі та периметра коричневої області
double calculateAreaOfBrownArea(double a) {
    double r = a / 2.0;

    // оскільки нам треба знайти площу - то перенесемо всі фігури в один квадрат. За
    підсумком отримуємо квадрат із вирізаним сектором та половину цього сектора
    // Розрахунок площі коричневої області
    double sectorArea = (M_PI * r * r) / 4.0; // площа чверті кола
    double halfSectorArea = sectorArea / 2.0; // площа однієї восьмої кола
    double squareWithoutSectorArea = r * r - sectorArea; // сума площ двох маленьких
    шматочків коричневої області
    double brownArea = halfSectorArea + squareWithoutSectorArea; // сума всіх площ
    коричневої області

    return brownArea;
}

double calculatePerimetrOfBrownArea(double a) {
    double r = a / 2.0;

    double lengthOfArc = M_PI * r / 4.0; // знаходження довжини дуги, яка становить
    одну восьму від кола
    double lengthOfSector = r * 2 + lengthOfArc; // знаходження периметра сектора

    double hypotenuse = sqrt(r * r * 2);
    double hypotenuseWithoutRadius = hypotenuse - r; // знаходження маленького
    діагонального шматочка

    double lengthOfLittleStubOfBrownArea = lengthOfArc + r + hypotenuseWithoutRadius;
    // знаходження периметра маленького діагонального шматочка

    double lengthOfBrownArea = lengthOfSector + lengthOfLittleStubOfBrownArea * 2; //
    знаходження периметра всієї коричневої області
    return lengthOfBrownArea;
}

```

```

}

void calculateAreaAndPerimetrOfBrownArea() {
    double a = inputSideOfSquare();

    cout << "Площа коричневої області: " << calculateAreaOfBrownArea(a) << "\n";
    cout << "Периметр коричневої області: " << calculatePerimetrOfBrownArea(a) <<
"\n";
}

int inputNumberOfTask() {
    Step1:
    cout << "Оберіть завдання:\n";
    cout << "1 - Перевірка високосного року\n";
    cout << "2 - Розрахунок належності точки до коричневої області\n";
    cout << "3 - Розрахунок параметрів коричневої області\n";
    cout << "Ваш вибір: ";

    int choice;
    cin >> choice;

    if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного від
користувача числа
        cout << "\n\"Ваш вибір\" повинно бути числом\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    if (choice <= 0 || choice >= 4) {
        cout << "\n\"Ваш вибір\" має дорівнювати одному, двом або трьом\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    return choice;
}

int main() {
    int choice = inputNumberOfTask();

    switch (choice) {
    case 1:
        checkLeapYear();
        break;
    case 2:
        pointInBrownAreaCheck();
        break;
    case 3:
        calculateAreaAndPerimetrOfBrownArea();
        break;
    }
}

```

```

default:
    cerr << "Неправильний вибір. Завершення програми.\n";
}

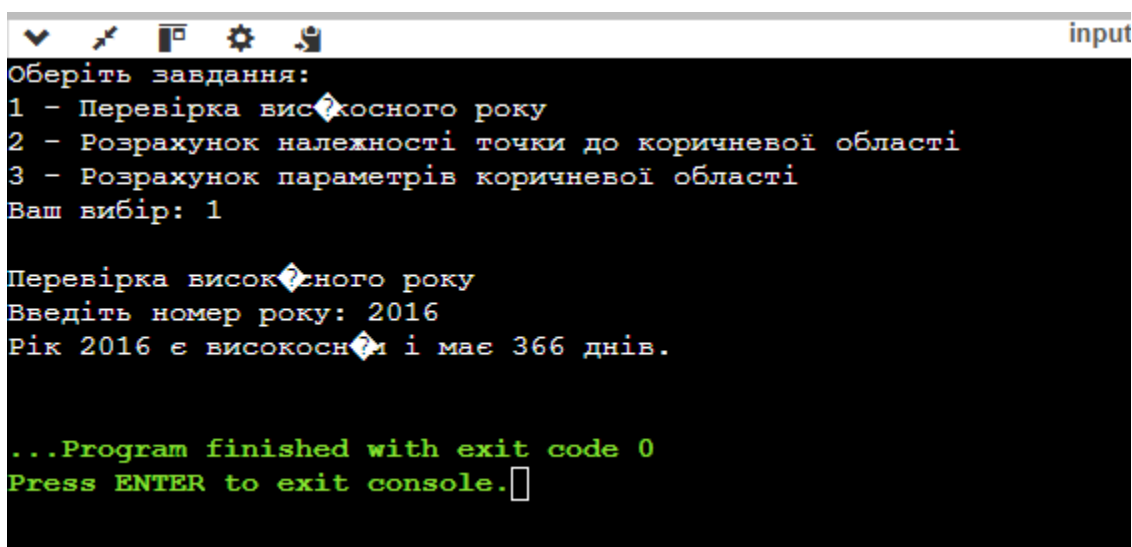
return 0;
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

На рис. Б.1 показаний приклад виконання завдання 1.



```

input
Оберіть завдання:
1 - Перевірка високосного року
2 - Розрахунок належності точки до коричневої області
3 - Розрахунок параметрів коричневої області
Ваш вибір: 1

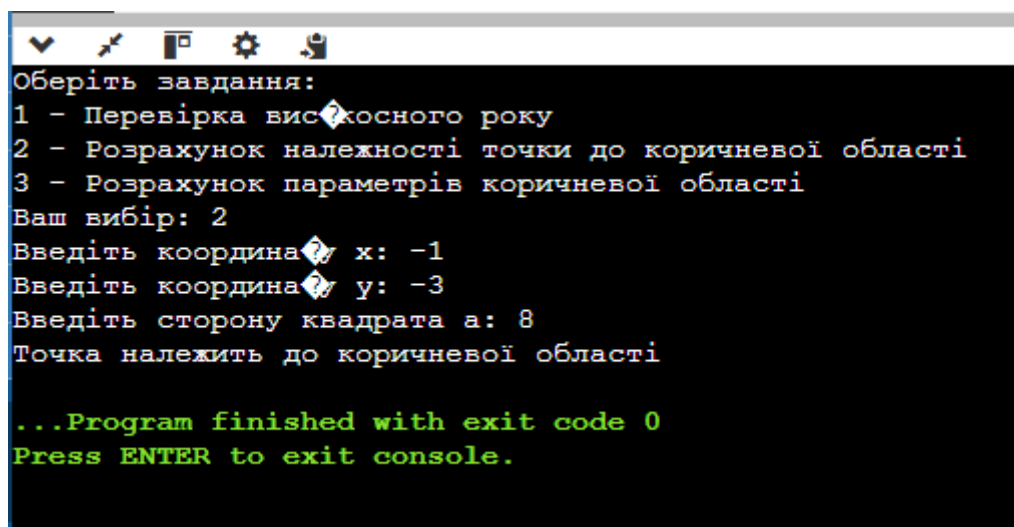
Перевірка високосного року
Введіть номер року: 2016
Рік 2016 є високосним і має 366 днів.

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

```

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

На рис. Б.2 показаний приклад виконання завдання 2.



```

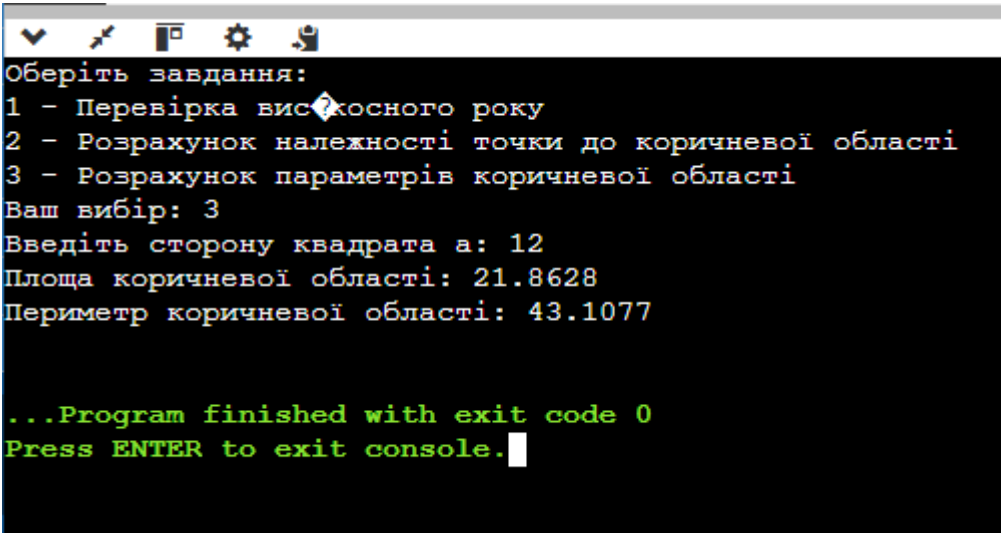
input
Оберіть завдання:
1 - Перевірка високосного року
2 - Розрахунок належності точки до коричневої області
3 - Розрахунок параметрів коричневої області
Ваш вибір: 2
Введіть координату x: -1
Введіть координату y: -3
Введіть сторону квадрата a: 8
Точка належить до коричневої області

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

```

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

На рис. Б.3 показаний приклад виконання завдання 3



```
Оберіть завдання:  
1 - Перевірка висхідного року  
2 - Розрахунок належності точки до коричневої області  
3 - Розрахунок параметрів коричневої області  
Ваш вибір: 3  
Введіть сторону квадрата a: 12  
Площа коричневої області: 21.8628  
Периметр коричневої області: 43.1077  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання