

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 №10

з дисципліни «Алгоритмізація та програмування» Тема:

"Створення і обробка структур даних мовою C++"

XAI.301.175.318.20 ЛР

Виконав студент гр. 318

Аліна ХОБОТ

(підпис, дата)

(П.І.Б.)

Перевірив

к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення структур (записів) мовою C ++, а також їх передачі в функції, і реалізувати декларування і обробку структур мовою C ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1.

Вирішити задачу зі структурами даних.

Param70 Використовуючи типи TPoint, TTriangle (див. Param64-65), описати функцію AreaN (P, N) дійсного типу, яка знаходить площу опуклого N-кутника, вершини якого (в порядку їх обходу) передаються в масиві P розміру N (> 2) з елементами типу TPoint. За допомогою цієї функції знайти площі трьох багатокутників, якщо дано число їх сторін і координати їх вершин.

Завдання 2.

Begin28 Дано число A. Обчислити A¹⁵, використовуючи дві допоміжні змінні і п'ять операцій множення. Для цього послідовно знайдіть A², A³, A⁵, A¹⁰, A¹⁵. Вивести всі знайдені степені числа A.

Boolean27. Дано числа x, y. Перевірити істинність висловлювання: «Точка з координатами (x, y) лежить в другій або третій координатній чверті».

A. Описати структуру, яка містить всі вхідні і всі вихідні дані задачі.

B. Визначити функцію (*метод), що реалізує обробку структури відповідно до задачі.

C. Визначити функцію (*метод), що перевіряє на коректність і заповнює відповідні поля вхідних даних структури.

D. Викликати функції (*методи) з пунктів C, B після оголошення змінної (об'єкту) структури.

E. Вивести значення полів вихідних даних.

Завдання 3. Рішення всіх трьох задач реалізувати в одному консольному додатку, *структурувати на модулі.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Опис функції:

Функція `AreaN` – обчислення площі опуклого N -кутника, вершини якого задані в масиві типу `TPoint`. Реалізовано розбиття багатокутника на $(N-2)$ трикутники, кожен з яких має спільну вершину `P[0]`. Площа кожного трикутника обчислюється за допомогою векторного добутку.

Опис параметрів:

`TPoint P` - масив із N точок типу `TPoint`, що представляють координати вершин багатокутника (в порядку їх обходу).

`int N` - кількість вершин багатокутника. Має бути більше 2 ($N > 2$), інакше функція повертає 0.

Опис результату:

Функція повертає дійсне число (тип `double`), яке є площею заданого опуклого багатокутника.

Якщо кількість вершин менше 3, функція повертає 0.

Лістинг коду вирішення завдання таблиця 1, варіант `Param70` наведено в дод. А (стор. 6).

Екран роботи програми показаний на рис. Б.1 (дод. Б, стор. 11)

Завдання 2.

Завдання `Begin28`.

Опис функції:

`inputData()` - вводить значення A .

`computePowers()` - обчислює всі степені, використовуючи лише 2 змінні і 5 множень.

`printResults()` - виводить усі знайдені степені.

Опис параметрів:

Всі функції працюють зі структурою `PowerData`, яка зберігає одне вхідне значення A і п'ять вихідних значень (степенів A).

Опис результату:

На екран виводяться значення A^2 , A^3 , A^5 , A^{10} , A^{15} .

Лістинг коду вирішення завдання таблиця 2, варіант `Begin28`. наведено в дод. А (стор. 6).

Екран роботи програми показаний на рис. Б.2 (дод. Б, стор. 11)

Завдання Boolean27

Функції:

`inputData()` - зчитує координати x і y .

`checkQuadrant()` - перевіряє, чи точка лежить у 2 або 3 чверті.

`printResult()` - виводить відповідне повідомлення.

`runTask2()` - об'єднує всі кроки в одному виклику.

Структура PointData:

Вхідні поля: x , y

Вихідне поле: `isIn23Quad` - логічне значення істинності твердження.

Лістинг коду вирішення завдання таблиця 3, варіант Boolean27. наведено в дод. А (стор. 9).

Екран роботи програми показаний на рис. Б.3 (дод. Б, стор. 11)

Завдання3

У рамках виконання Завдання 3 усі три задачі були реалізовані в одному консольному додатку з дотриманням модульної структури. Кожна задача винесена в окремі файли, що відповідає принципам організації коду та забезпечує зручність супроводу.

Зокрема, задачі розміщено у відповідних файлах: `task1.h` та `task1.cpp` містять реалізацію першої задачі (Param70), `task2.h` та `task2.cpp` - другої задачі (Begin28), а `task3.h` та `task3.cpp` - третьої задачі (Boolean27). Головна функція `main()` знаходиться в окремому файлі `main.cpp` і викликає функції з усіх трьох задач, об'єднуючи програму в єдине ціле.

Лістинг коду вирішення завдання 3. наведено в дод. А (стор. 6).

Екран роботи програми показаний на рис. Б.4 (дод. Б, стор. 11)

ВИСНОВКИ

Було вивчено теоретичний матеріал з основ представлення структур (записів) мовою C ++, а також їх передачі в функції, і реалізовано декларування і обробка структур мовою C ++ в середовищі Visual Studio.

ДОДАТОК А

```

#include <iostream> // Підключення бібліотеки для вводу/виводу
#include <Windows.h> // Підключення бібліотеки для роботи з Windows API
#include "task1.h" // Заголовочний файл для завдання 1
#include "task2.h" // Заголовочний файл для завдання 2
#include "task3.h" // Заголовочний файл для завдання 3
using namespace std;

int main() {
    // Встановлення підтримки кирилиці
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    setlocale(LC_ALL, "");

    int choice; // Змінна для збереження вибору користувача
    do {
        // Виведення головного меню
        cout << "\n=== ГОЛОВНЕ МЕНЮ ===\n";
        cout << "1. Завдання 1\n";
        cout << "2. Завдання 2\n";
        cout << "3. Завдання 3\n";
        cout << "0. Вихід\n";
        cout << "Ваш вибір: ";
        cin >> choice; // Зчитування вибору користувача

        // Обробка вибору за допомогою конструкції switch
        switch (choice) {
            case 1:
                runTask1(); // Виклик функції для завдання 1
                break;
            case 2:
                runTask2(); // Виклик функції для завдання 2
                break;
            case 3:
                runTask3(); // Виклик функції для завдання 3
                break;
            case 0:
                cout << "Завершення програми.\n"; // Повідомлення про завершення
                break;
            default:
                cout << "Невірний вибір.\n"; // Повідомлення при неправильному вводі
        }
    } while (choice != 0); // Повторення, поки не вибрано "0"

    return 0; // Завершення програми
}

#pragma once

```

```

void runTask1();

#include <iostream> // Бібліотека для роботи з введенням/виведенням
#include <cmath> // Бібліотека для математичних функцій
using namespace std;

// Param70: Реалізація функції AreaN
double AreaN(TPoint P[], int N) {
    double area = 0.0; // Загальна площа багатокутника

    // Перевірка: багатокутник повинен мати принаймні 3 вершини
    if (N < 3) return 0.0;

    // Розбиваємо N-кутник на (N-2) трикутники з базовою точкою P[0]
    for (int i = 1; i < N - 1; i++) {
        // Формуємо трикутник T з вершин P[0], P[i], P[i+1]
        TTriangle T;
        T.A = P[0];
        T.B = P[i];
        T.C = P[i + 1];

        // Обчислюємо площу трикутника за формулою векторного добутку
        double triangleArea = 0.5 * fabs(
            (T.B.X - T.A.X) * (T.C.Y - T.A.Y) -
            (T.C.X - T.A.X) * (T.B.Y - T.A.Y)
        );

        // Додаємо площу цього трикутника до загальної площі
        area += triangleArea;
    }

    return area; // Повертаємо загальну площу багатокутника
}

// Основна функція
void runTask1() {
    // Вхідні дані: координати трьох опуклих багатокутників

    // Багатокутник 1: прямокутник (4 вершини)
    TPoint polygon1[] = {
        {0, 0},
        {4, 0},
        {4, 3},
        {0, 3}
    };
    int n1 = 4;

    // Багатокутник 2: п'ятикутник (5 вершин)
    TPoint polygon2[] = {

```

```

        {1, 0},
        {4, 1},
        {3, 4},
        {0, 4},
        {-1, 1}
    };
    int n2 = 5;

    // Багатокутник 3: шестикутник (6 вершин)
    TPoint polygon3[] = {
        {0, 0},
        {2, 0},
        {3, 2},
        {2, 4},
        {0, 4},
        {-1, 2}
    };
    int n3 = 6;

    // Виведення площ кожного багатокутника
    cout << "Площа багатокутника 1: " << AreaN(polygon1, n1) << endl;
    cout << "Площа багатокутника 2: " << AreaN(polygon2, n2) << endl;
    cout << "Площа багатокутника 3: " << AreaN(polygon3, n3) << endl;

}

#pragma once

void runTask2();

#include <iostream> // Підключення бібліотеки для роботи з введенням/виведенням
using namespace std;

// А. Опис структури з вхідними та вихідними даними
struct PowerData {
    double A;          // Вхідне число
    double A2, A3;     // A^2, A^3
    double A5;         // A^5
    double A10;        // A^10
    double A15;        // A^15
};

// С. Функція для перевірки і введення вхідних даних
void inputData(PowerData& data) {
    cout << "Введіть число А: ";
    cin >> data.A;
}

// В. Функція обробки (обчислення всіх степенів з використанням 5 множень)
void computePowers(PowerData& data) {

```



```

double x = data.A;

// 1) A2 = A * A
data.A2 = x * x;

// 2) A3 = A2 * A
data.A3 = data.A2 * x;

// 3) A5 = A2 * A3
data.A5 = data.A2 * data.A3;

// 4) A10 = A5 * A5
data.A10 = data.A5 * data.A5;

// 5) A15 = A10 * A5
data.A15 = data.A10 * data.A5;
}

// Е. Виведення результатів
void printResults(const PowerData& data) {
    cout << "A^2  = " << data.A2 << endl;
    cout << "A^3  = " << data.A3 << endl;
    cout << "A^5  = " << data.A5 << endl;
    cout << "A^10 = " << data.A10 << endl;
    cout << "A^15 = " << data.A15 << endl;
}

// Основна функція для запуску завдання
void runTask2() {
    PowerData data;          // D. Оголошення змінної структури
    inputData(data);         // С. Виклик функції вводу
    computePowers(data);     // В. Виклик функції обчислення
    printResults(data);      // Е. Вивід результатів
}

#pragma once

void runTask3();

#include <iostream> // Підключення бібліотеки для введення/виведення
using namespace std;

// А. Структура з вхідними та вихідними даними
struct PointData {
    double x, y;             // Вхідні координати
    bool isIn23Quad;         // Вихідне значення: чи точка у 2 або 3 чверті
};

// С. Функція введення і перевірки вхідних даних
void inputData(PointData& data) {

```

```

    cout << "Введіть координати x та y\n";
    cout << "x: ";
    cin >> data.x; // Зчитування x
    cout << "y: ";
    cin >> data.y; // Зчитування y
}

// В. Функція обробки (перевірка чвертей)
void checkQuadrant(PointData& data) {
    // Друга чверть: x < 0, y > 0
    // Третя чверть: x < 0, y < 0
    data.isIn23Quad = (data.x < 0 && data.y != 0);
}

// Е. Виведення результату
void printResult(const PointData& data) {
    if (data.isIn23Quad)
        cout << "Точка лежить у другій або третій координатній чверті." << endl;
    else
        cout << "Точка НЕ лежить у другій або третій координатній чверті." <<
endl;
}

// Д. Основна функція
void runTask3() {
    PointData point;
    inputData(point);          // С. Введення координат
    checkQuadrant(point);      // В. Перевірка належності до чверті
    printResult(point);        // Е. Виведення результату
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програм

```
Площа багатокутника 1: 12
Площа багатокутника 2: 14.5
Площа багатокутника 3: 12
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання Param70.

```
Введіть число A: 2
A^2 = 4
A^3 = 8
A^5 = 32
A^10 = 1024
A^15 = 32768
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання Begin28.

```
Введіть координати x та y
x: -2
y: -7
Точка лежить у другій або третій координатній чверті.
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання Boolean27.

```
=== ГОЛОВНЕ МЕНЮ ===
1. Завдання 1
2. Завдання 2
3. Завдання 3
0. Вихід
Ваш вибір: 1
Площа багатокутника 1: 12
Площа багатокутника 2: 14.5
Площа багатокутника 3: 12

=== ГОЛОВНЕ МЕНЮ ===
1. Завдання 1
2. Завдання 2
3. Завдання 3
0. Вихід
Ваш вибір: 2
Введіть число A: 2
A^2 = 4
A^3 = 8
A^5 = 32
A^10 = 1024
A^15 = 32768

=== ГОЛОВНЕ МЕНЮ ===
1. Завдання 1
2. Завдання 2
3. Завдання 3
0. Вихід
Ваш вибір: 3
Введіть координати x та y
x: -4
y: -5
Точка лежить у другій або третій координатній чверті.

=== ГОЛОВНЕ МЕНЮ ===
1. Завдання 1
2. Завдання 2
3. Завдання 3
0. Вихід
Ваш вибір: 0
Завершення програми.
```

Рисунок Б.4 – Екран виконання програми для вирішення завдання 3

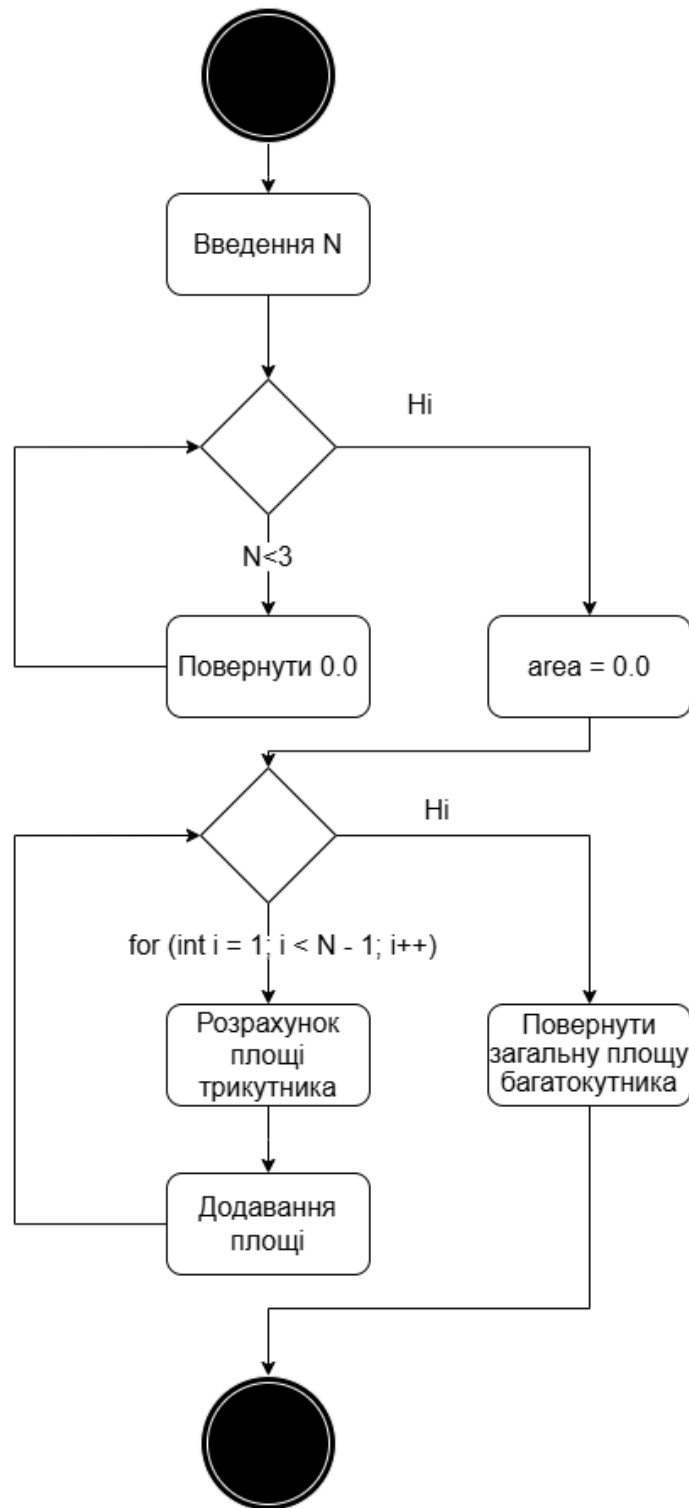


Рисунок Б.5 – Діаграма для завдання Param70.

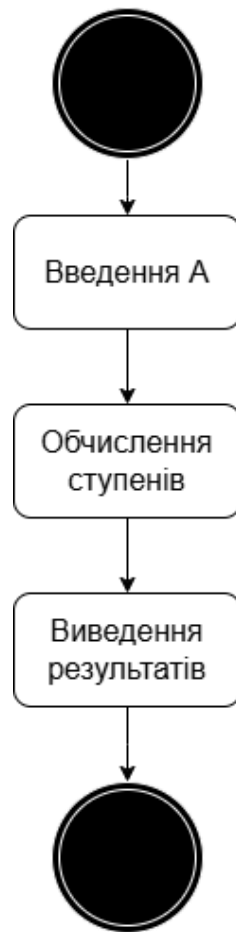


Рисунок Б.6 – Діаграма для завдання Begin28.

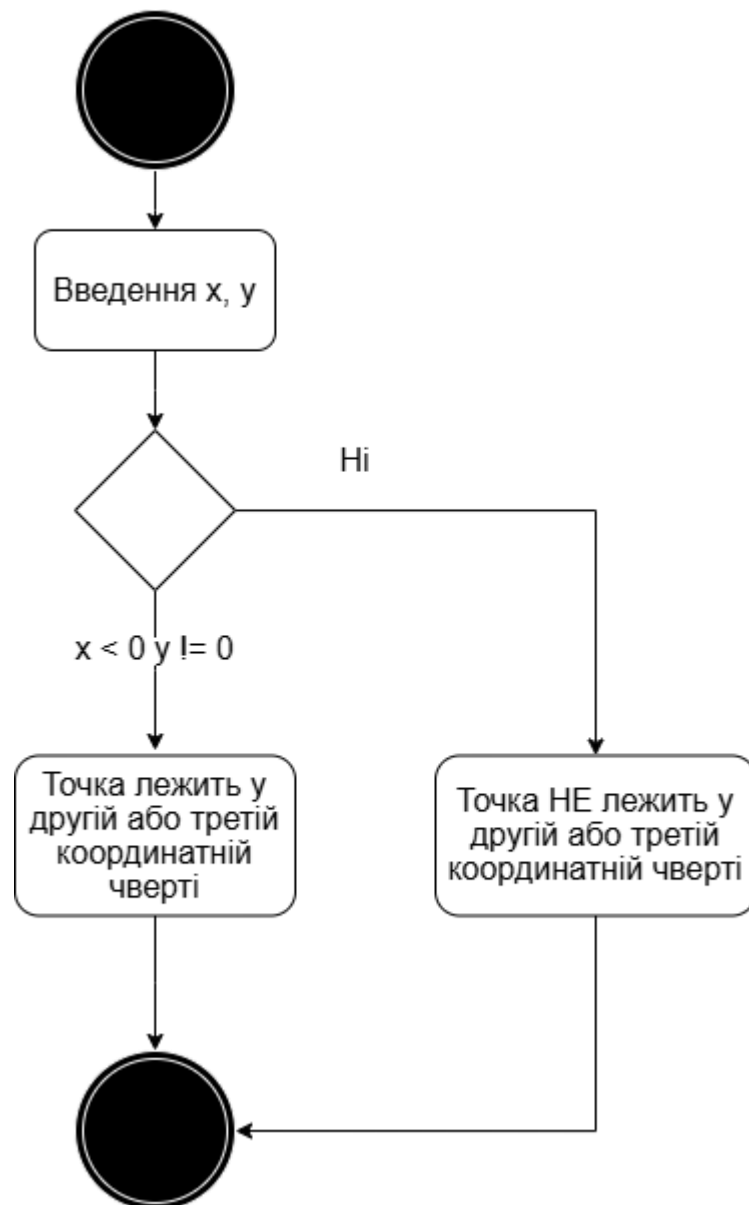


Рисунок Б.7 – Діаграма для завдання Boolean27.