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»

Тема: "Реалізація алгоритмів з розгалуженням мовою С ++"

ХАІ.301.175.318.13 ЛР

Виконав студент гр. № 318

Мальківська В. А

(підпис, дата) (П.І.Б.)

Перевірив к.т.н., доцент

( вчена ступінь, вчене звання)

  Гавриленко О. В

(підпис, дата) (П.І.Б.)

2024

# 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 C++ в середовищі Visual Studio. Також опанувати та відпрацювати навички

структурування програми з функціями.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на алгоритми з розгалуженням.

Вираз представлено в табл.1. Варіант 12.

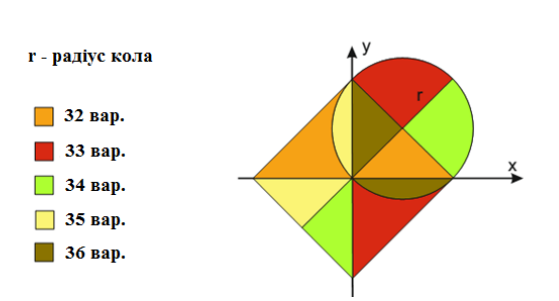
If12. Дано три числа. Знайти найменше з них.

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. Варіант 32.



Завдання 3. Обчислити площу і периметр плоскої фігури.

Варіанти представлено в табл.2. Варіант 32.

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 (If12.).

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

Ім'я: a, b, c;

Тип: int;

Опис: Три цілі числа, між якими потрібно знайти найменше;

Вихідні дані (ім’я, опис, тип, обмеження):

Ім'я: min\_val;

Тип: int;

Опис: Найменше з трьох чисел;

Алгоритм:

Користувач вводить три числа a, b, c;

Використовується функція min() для знаходження найменшого з них;

Результат виводиться на екран;

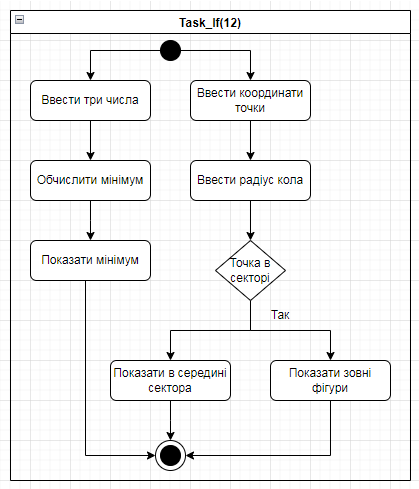


Рисунок 1 – Діаграма для завдання 1

Завдання 2.

Визначити, чи точка потрапляє в фігуру (варіант 32.).

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

Ім'я: x, y, r;

Тип: float;

Опис: Координати точки (x,y) та радіус кола.

Обмеження: r>0 — радіус кола має бути додатнім числом;

x≥0, y≥0 — точка має бути у першому квадранті (для потрапляння в оранжевий сектор);

Вихідні дані (ім’я, опис, тип, обмеження):

Ім'я: result;

Тип: string;

Опис: Повідомлення про те, чи знаходиться точка всередині оранжевого сектора;

Алгоритм:

1. Користувач вводить координати точки x,y і радіус кола r;
2. Перевіряється, чи точка знаходиться у першому квадранті: x≥0 і y≥0;
3. Обчислюється відстань від початку координат до точки: x² + у² ≤ r², що перевіряє, чи знаходиться точка всередині кола;
4. Перевіряється умова y≤x, щоб перевірити, чи точка знаходиться нижче або на прямій y=x (всередині сектора).
5. Якщо всі умови виконані, виводиться повідомлення "Точка знаходиться всередині оранжевого сектора", інакше — "Точка поза фігурою";

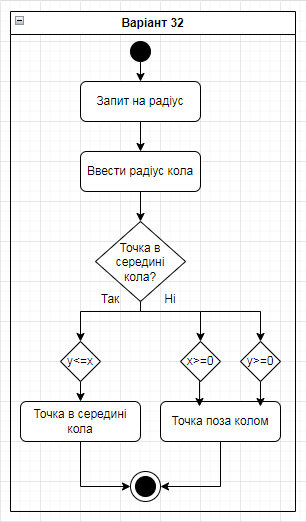


Рисунок 2 – Діаграма для завдання 2

Завдання 3.Обчислення площі і периметра кола (варіант 32.)

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

Ім'я: r;

Тип: float;

Опис: Радіус кола;

Обмеження: r>0 — радіус має бути додатнім числом;

Вихідні дані (ім’я, опис, тип, обмеження):

Ім'я: area, perimeter;

Тип: float;

Опис: area-площа кола, яка обчислюється за формулою πr²;

perimeter-периметр кола (довжина кола), яка обчислюється за формулою 2πr;

Алгоритм:

1. Користувач вводить радіус кола r;
2. Якщо r>0, обчислюється площа кола за формулою πr²;
3. Обчислюється периметр кола за формулою 2πr;
4. Результати виводяться на екран;
5. Якщо радіус недійсний (менше або рівний нулю), виводиться повідомлення про помилку;



Рисунок 3 – Діаграма для завдання 3

Завдання 4.

Меню для вибору завдання

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

Ім'я: choice;

Тип: int;

Опис: Номер завдання, яке хоче виконати користувач;

Вихідні дані (ім’я, опис, тип, обмеження):

Тип: void (виводиться результат виконання вибраного завдання);

Алгоритм:

1. Виводиться меню для вибору завдання:

Для виконання завдання 1 (найменше з трьох чисел);

Для виконання завдання 2 (перевірка координат точки);

Для виконання завдання 3 (площа і периметр кола);

1. Користувач вводить свій вибір choice;
2. Залежно від вибору користувача викликається відповідна функція:

Якщо choice == 1, викликається функція task\_if12();

Якщо choice == 2, викликається функція task\_geom32();

Якщо choice == 3, викликається функція task\_area\_perimeter32();

Якщо введений номер завдання недійсний, виводиться повідомлення про помилку;

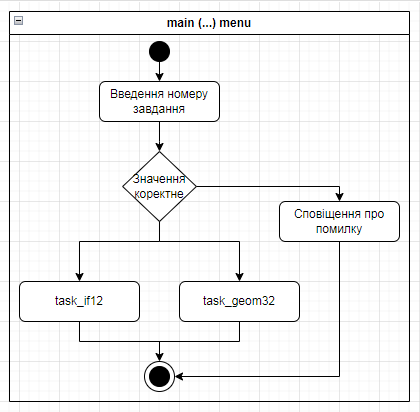


Рисунок 4 - Діаграма для завдання 4

ВИСНОВКИ

Під час виконання лабораторної роботи було створено програму на С++, яка дозволяє знаходити найменше з трьох чисел, перевіряти, чи знаходиться точка в середині заданої області, а також обчислювати площу і периметр частини кола.

ДОДАТОК А

Лістинг до коду

#include <iostream>

#include <cmath>

using namespace std;

// Назва функції для кожного завдання:

void task\_if12(); // Завдання 1: Алгоритми з розгалуженням (варіант 12)

void task\_geom32(); // Завдання 2: Визначити, чи точка потрапляє в фігуру (варіант 32)

void task\_area\_perimeter32(); // Завдання 3: Обчислення площі і периметра фігури (варіант 32)

void display\_menu(); // Завдання 4: Меню для вибору завдання

int main() {

display\_menu(); // Показати меню і виконати вибране завдання

return 0;

}

// Завдання 1: Алгоритми з розгалуженням (варіант 12)

void task\_if12() {

int a, b, c;

cout << "Завдання 1: Введіть три числа для знаходження найменшого з них: ";

cin >> a >> b >> c;

int min\_val = min(a, min(b, c)); // Пошук найменшого з трьох

cout << "Найменше число: " << min\_val << endl;

}

// Завдання 2: Визначити, чи точка потрапляє в фігуру (варіант 32)

void task\_geom32() {

float x, y, r;

cout << "Завдання 2: Введіть координати точки (x, y): ";

cin >> x >> y;

cout << "Введіть радіус кола: ";

cin >> r;

// Умова для перевірки, чи точка в оранжевому секторі

if (x >= 0 && y >= 0 && (x \* x + y \* y <= r \* r) && y <= x) {

cout << "Точка знаходиться всередині оранжевого сектора (варіант 32)." << endl;

} else {

cout << "Точка поза фігурою." << endl;

}

}

// Завдання 3: Обчислення площі і периметра кола (варіант 32)

void task\_area\_perimeter32() {

float r;

cout << "Завдання 3: Введіть радіус кола: ";

cin >> r;

if (r > 0) {

float area = M\_PI \* r \* r; // Площа кола

float perimeter = 2 \* M\_PI \* r; // Периметр кола

cout << "Площа кола: " << area << endl;

cout << "Периметр кола: " << perimeter << endl;

} else {

cout << "Радіус повинен бути додатнім числом!" << endl;

}

}

// Завдання 4: Меню для вибору завдання

void display\_menu() {

int choice;

cout << "Оберіть завдання:\n";

cout << "1 - Найменше з трьох чисел (Завдання 1)\n";

cout << "2 - Точка в фігурі (Завдання 2)\n";

cout << "3 - Площа і периметр кола (Завдання 3)\n";

cout << "Ваш вибір: ";

cin >> choice;

switch (choice) {

case 1: task\_if12(); break;

case 2: task\_geom32(); break;

case 3: task\_area\_perimeter32(); break;

default: cout << "Неправильний вибір! Спробуйте ще раз.\n"; break;

}

}

ДОДАТОК Б

На рисунку 1 представлено скріншот вікна виконання програми

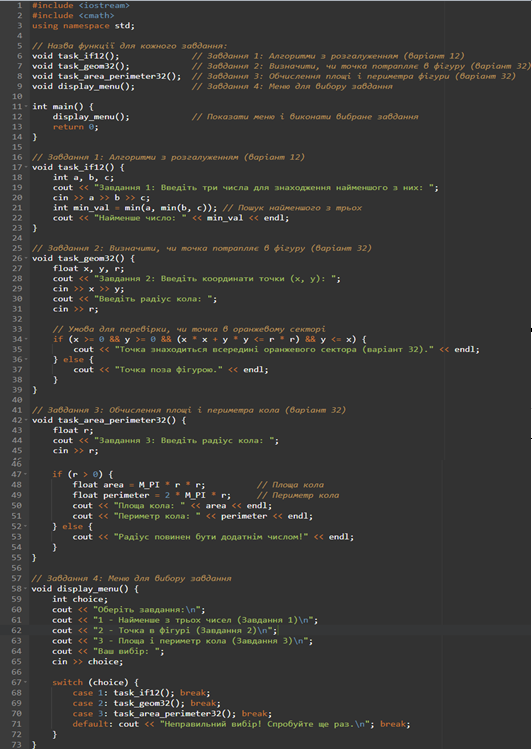


Рисунок 1 – Алгоритм вирішення задач

На рисунку 2 представлено скріншот вікна виконання роботи

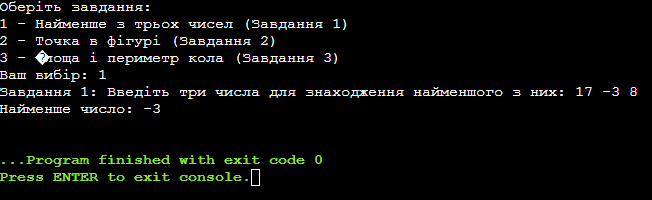


Рисунок 2 – Результат запуску програми