

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 ++»

ХАІ.301.173.310.1 ЛР

Виконав студент гр. 310

Микола АНДРЮШКІН
(підпис, дата) (П.І.Б.)

Перевірив

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

2024

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

If20, табл.1, варіант 1.

Завдання 2. Дано координати точки на площині (x, y). Визначити, чи потрапляє точка в фігуру(рис. 1) заданого кольору (або групу фігур) і вивести відповідне повідомлення.

Geo36, табл.2, варіант 1.

Завдання 3. Обчислити площу і периметр плоскої фігури(рис. 1).

Geo36, табл.2, варіант 1.

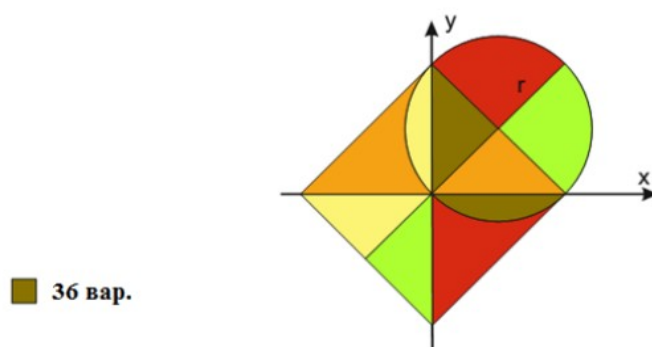


Рисунок — 1 — Плоска фігура

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 If20.

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

a - точка a, double, від $1.7E-308$ до $1.7E+308$.

b - точка b, double, від $1.7E-308$ до $1.7E+308$.

c - точка c, double, від $1.7E-308$ до $1.7E+308$.

Вихідні дані (ім'я, опис, тип):

disc — відстань між точками a та c, double, від $1.7E-308$ до $1.7E+308$.

disb - відстань між точками a та b, double, від $1.7E-308$ до $1.7E+308$.

Алгоритм вирішення показано на рис. 2.

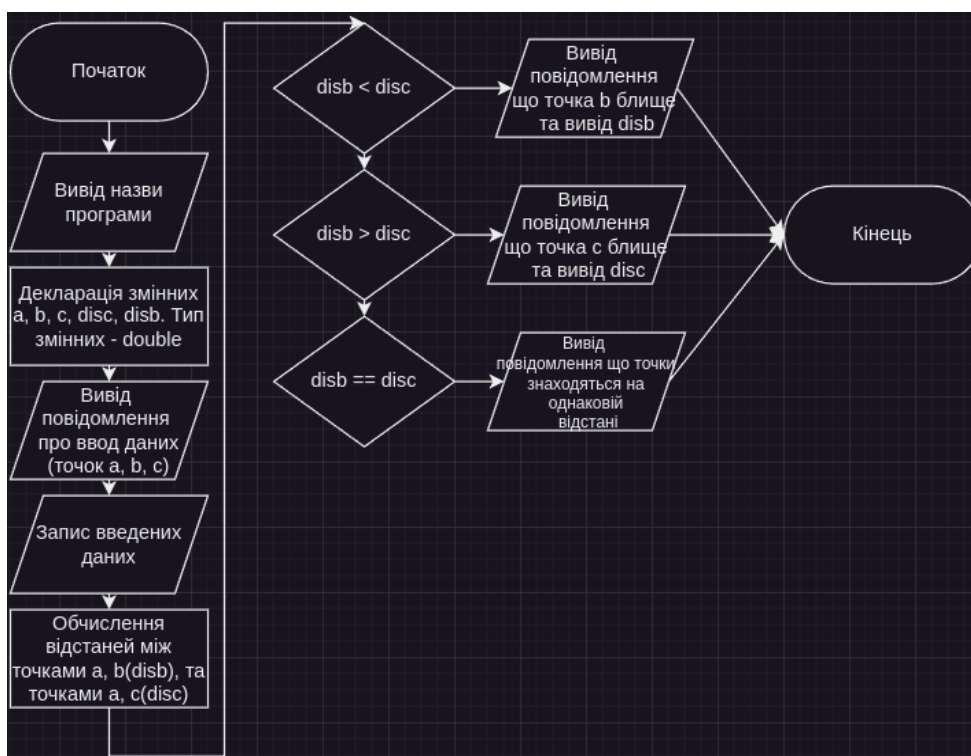


Рисунок 2 – Алгоритм розв'язку If20

Лістинг коду вирішення задачі If20 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Geo36_2.

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

r — радіус кола, double, від $1.7E-308$ до $1.7E+308$.

Вихідні дані (ім'я, опис, тип):

ax — координата x точки a , double, від $1.7E-308$ до $1.7E+308$.

ay — координата y точки a , double, від $1.7E-308$ до $1.7E+308$.

bx — координата x точки b , double, від $1.7E-308$ до $1.7E+308$.

by — координата y точки b , double, від $1.7E-308$ до $1.7E+308$.

cx — координата x точки c , double, від $1.7E-308$ до $1.7E+308$.

cy — координата y точки c , double, від $1.7E-308$ до $1.7E+308$.

px — координата x обраної точки, double, від $1.7E-308$ до $1.7E+308$.

py — координата y обраної точки, double, від $1.7E-308$ до $1.7E+308$.

$distAB$ — дистанція між точками a , b , double, від $1.7E-308$ до $1.7E+308$.

$distAC$ — дистанція між точками a , c , double, від $1.7E-308$ до $1.7E+308$.

$distBC$ — дистанція між точками c , b , double, від $1.7E-308$ до $1.7E+308$.

S_org — площа трикутника обмеженого точками a , b , c , double, від $1.7E-308$ до $1.7E+308$.

$distPA$ — дистанція між точками a , p , double, від $1.7E-308$ до $1.7E+308$.

$distPB$ — дистанція між точками p , b , double, від $1.7E-308$ до $1.7E+308$.

$distPC$ — дистанція між точками p , c , double, від $1.7E-308$ до $1.7E+308$.

S_PAB — площа трикутника обмеженого точками a , b , p , double, від $1.7E-308$ до $1.7E+308$.

S_PBC — площа трикутника обмеженого точками p , b , c , double, від $1.7E-308$ до $1.7E+308$.

S_PCA — площа трикутника обмеженого точками a , p , c , double, від $1.7E-308$ до $1.7E+308$.

S_point — сума $S_PAB + S_PBC + S_PCA$, double, від $1.7E-308$ до $1.7E+308$.

S_org_char — S_org але переведена у char, char, від -128 до 127.

S_point_char — S_point але переведена у char, char, від -128 до 127.

$rsquared$ — r у квадраті, double, від $1.7E-308$ до $1.7E+308$.

$Leftside$ — рівняння кола, необхідне для обчислення результату, double, від $1.7E-308$ до $1.7E+308$.

Алгоритм вирішення показано на рис. 3.

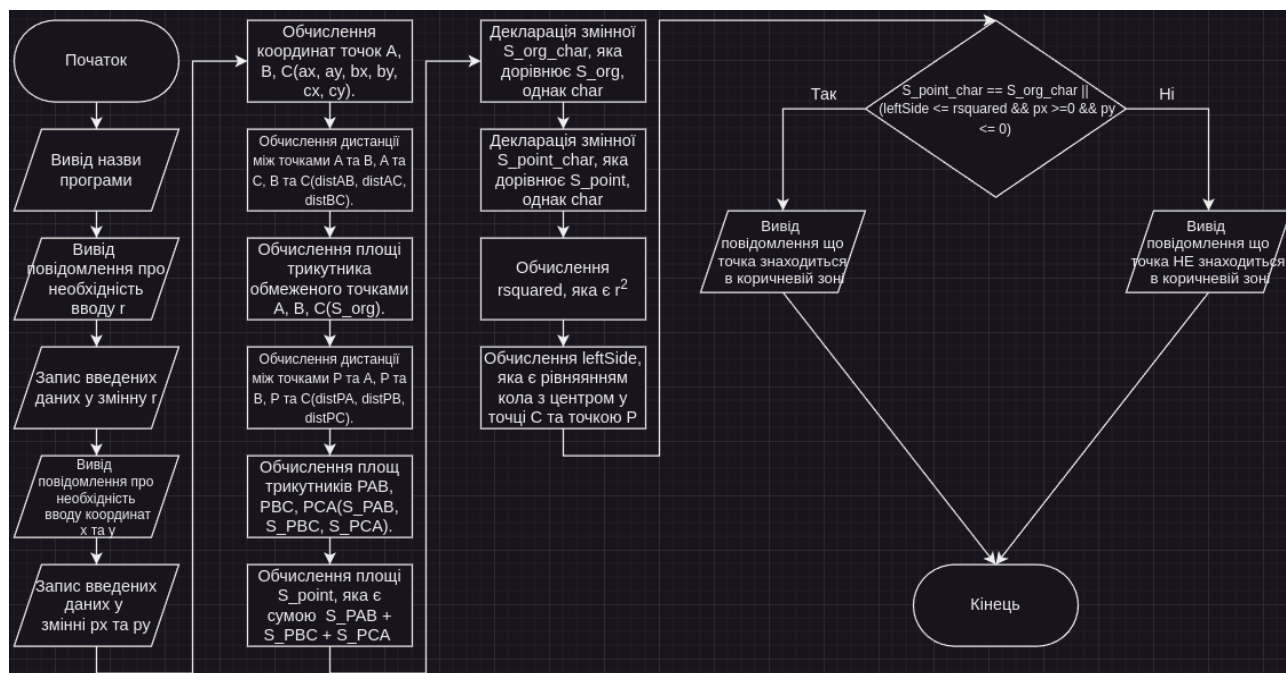


Рисунок 3 — Алгоритм розв'язку Geo36_2

Лістинг коду вирішення задачі Geo36_2 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.2.

Завдання 3.

Вирішення задачі Geo36_3.

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

r — радіус кола, double, від $1.7E-308$ до $1.7E+308$.

Вихідні дані (ім'я, опис, тип):

chord — довжина хорди кола, double, від $1.7E-308$ до $1.7E+308$.

QuaterAR — площа чверті кола, double, від $1.7E-308$ до $1.7E+308$.

curl — довжина дуги кола, double, від $1.7E-308$ до $1.7E+308$.

Perim — периметр коричневої зони, double, від $1.7E-308$ до $1.7E+308$.

Алгоритм вирішення показано на рис. 4.

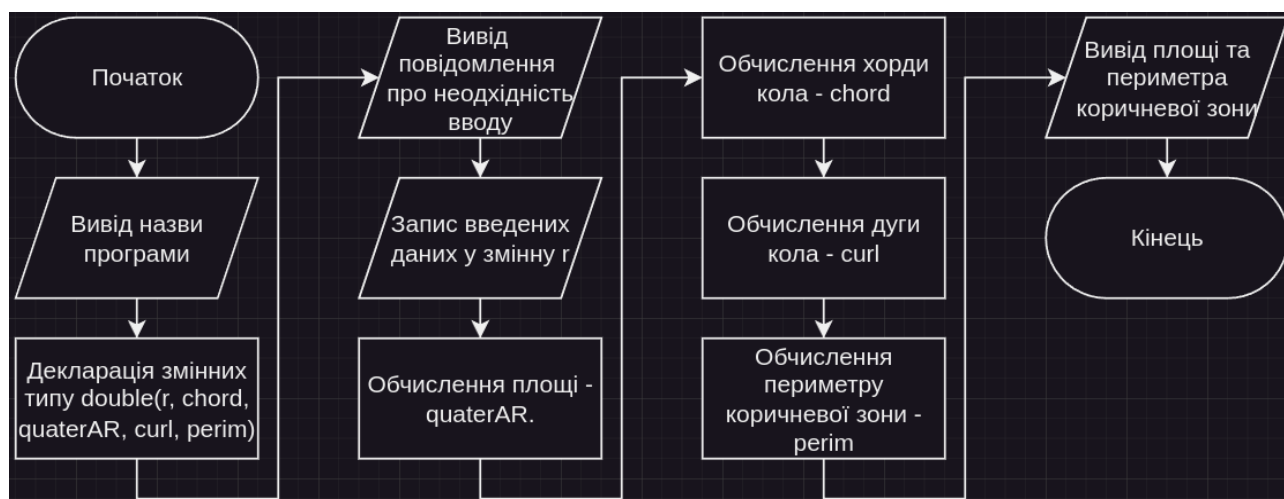


Рисунок 4 — Алгоритм розв'язку Geo36_3

Лістинг коду вирішення задачі Geo36_3 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.3.

Завдання 4.

Вирішення задачі 4.

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

crs — курсор(вибір задачі), int, від -2 147 483 648 до 2 147 483 647.

Алгоритм вирішення показано на рис. 5.

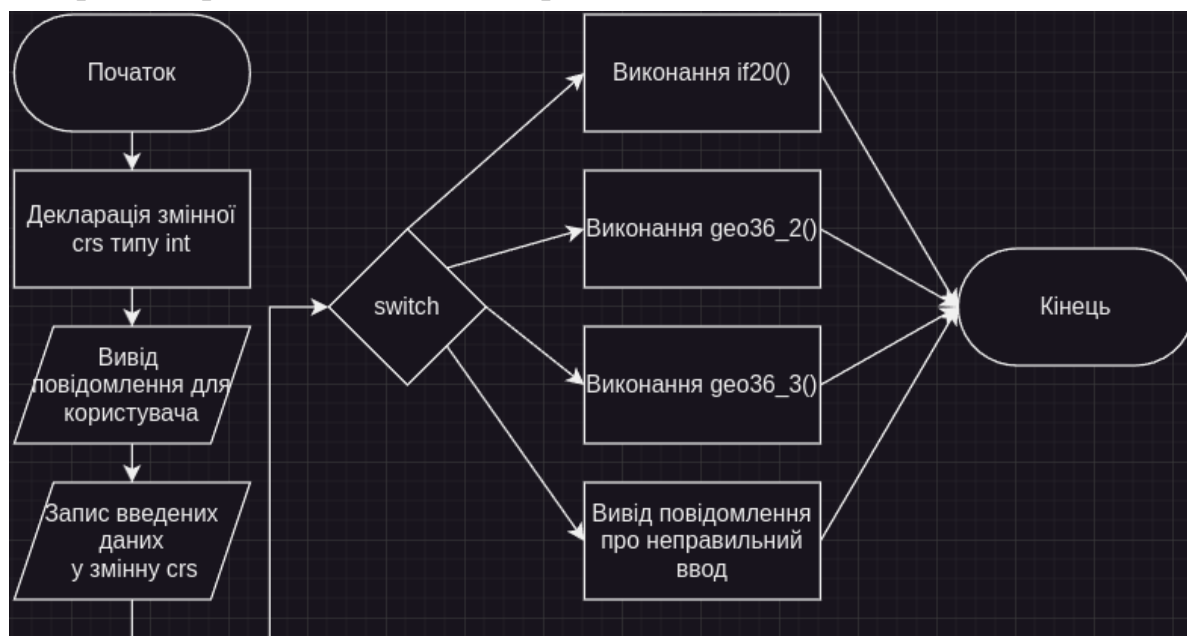


Рисунок 5 — Алгоритм виконання завдання 4

Лістинг коду вирішення завдання 4 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.4.

ВИСНОВКИ

Вивчив теоретичний матеріал щодо синтаксису у мові C++ і подання у вигляді UML діаграм активності алгоритмів з розгалуженням та реалізував алгоритми з використанням інструкцій умовного переходу і вибору мовою C++. Також опанував та відпрацював навички структурування програми з функціями.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
#include <cmath>
using namespace std;
#define _USE_MATH_DEFINES

void if20(); //if20 declaration
void geo36_2(); //geo36_2 declaration
void geo36_3(); //geo36_3 declaration

double degToRad(double x) { //Degrees to radians
    return x * (M_PI / 180.0);
}

double pointDist(double ax, double ay, double bx, double by){ //calculation of
distance between 2 points
    double dist = sqrt(pow(bx - ax, 2.0) + pow(by - ay, 2.0));
    return dist;
}

double triangArea(double AB, double AC, double BC){ //calculation of triangle
area
    double p = (AB + AC + BC) / 2.0;
    double S = sqrt(p * (p - AB) * (p - AC) * (p - BC));
    return S;
}

int main(){
    int crs; //declaration of variable
    cout << "Select task \n" << "1 - if20 \n" << "2 - geo36_2 \n" << "3 -
geo36_3 \n" << "Select - "; //menu for user
    cin >> crs;
    switch (crs){ //menu backend
        case 1: if20(); break;
        case 2: geo36_2(); break;
        case 3: geo36_3(); break;
        default: cout << "Only 1, 2 or 3!!! \n"; break;
    }
    return 0;
}

void if20(){ // if20 realisation
    cout << "***** If 20 *****" << endl;
    double a,b,c,disc,discb; //declaration of variables
    cout << "Enter values of A, B, C - "; //message for user

```



```

    cin >> a >> b >> c;
    disb = fabs(a - b); //distance calculation
    disc = fabs(a - c);
    if (disb < disc){ //returning results for user
        cout << "B \n" << "Distance - " << disb;
    }
    if (disb > disc){
        cout << "C \n" << "Distance - " << disc;
    }
    if (disb == disc) {
        cout << "Same distance";
    }
}

void geo36_2(){
    cout << "***** GEO 36_2 *****" << endl;
    double ax, ay, bx, by, cx, cy, px, py, r; //declaration of variables
    cout << "R - "; //message for user
    cin >> r; //Input
    cout << "Enter point (x, y) - "; //message for user
    cin >> px >> py; //user input

    //calculation of 3 main points: A, B, C
    ax = 0;
    ay = 0; //A is placed in the beginning of coordinates(0, 0)
    bx = 0; //B is placed on OY axis. X = 0, Y is calculated using Pythagorean
theorem
    by = sqrt(pow(r, 2) + pow(r, 2));
    cx = by / 2; //C is centre of the circle. X is calculated using by
coordinate. Y is calculated using Pythagorean theorem
    cy = sqrt(pow(r, 2) - pow(cx, 2));

    //calculation of triangle sides lengthes
    double distAB = pointDist(ax, ay, bx, by);
    double distAC = pointDist(ax, ay, cx, cy);
    double distBC = pointDist(bx, by, cx, cy);
    double S_org = triangArea(distAB, distAC, distBC); //calculation of triangle
area

    //calculations of point to triangle point distance
    double distPA = pointDist(px, py, ax, ay);
    double distPB = pointDist(px, py, bx, by);
    double distPC = pointDist(px, py, cx, cy);

    //calculation of area of sub triangles
    double S_PAB = triangArea(distAB, distPB, distPA);
    double S_PBC = triangArea(distPB, distPC, distBC);
    double S_PCA = triangArea(distPA, distPC, distAC);

```

```

//additional calculations
double S_point = S_PAB + S_PBC + S_PCA; //sum of the sub triangles area
char S_org_char = S_org; //double to char conversion
char S_point_char = S_point; //double to char conversion
double rsquared = pow(r, 2); //separate calculations for if
double leftSide = pow(px - cx, 2) + pow(py - cy, 2);

    if (S_point_char == S_org_char || (leftSide <= rsquared && px >=0 && py <=
0)){ //is point places in brown zone
        cout << "Point is in brown region \n";
    }
    else {
        cout << "Point isn`t in brown region \n";
    }
}

void geo36_3(){ //geo36_3 realisation
    cout << "***** GEO 36_3 *****" << endl;
    double r, chord, quaterAR, curl, perim;
    cout << "Enter radius - ";
    cin >> r;

    //area calculation
    quaterAR = ((M_PI * pow(r, 2)) / 360) * 90;

    //perimeter calcualtion
    chord = 2 * r * sin(degToRad(45)); //chord calculation
    curl = ((M_PI * r) / 180) * 90; //curvature calculation
    perim = chord + curl + r + r + chord; //perimeter calculation

    cout << endl << "Area is - " << quaterAR << endl << "Perimeter is - " <<
perim << endl; //results for user
}

```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
***** If 20 *****  
Enter values of A, B, C - 1 4 7  
B  
Distance - 3
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання If20

```
***** GEO 36_2 *****  
R - 2.1  
Enter point (x, y) - 0.5 1.5  
Point is in brown region
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Geo36_2

```
***** GEO 36_3 *****  
Enter radius - 2  
  
Area is - 3.14159  
Perimeter is - 12.7984
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання
Geo36_3

```
Select task
1 - if20
2 - geo36_2
3 - geo36_3
Select - 1
***** If 20 *****
```

Рисунок Б.4 – Екран виконання програми для вирішення задачі 4