

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. 310 група, 7 номер в списку ЛР

Виконав студент гр.

_____310_____

_____ Стеценко

_____ СофіяОлександрівна

(підпис, дата)

(П.І.Б.)

Перевірів

_____ к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

Харків 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.

If26: Вирішити завдання на алгоритми з розгалуженням. Для заданого дійсного x знайти значення наступної функції f , що приймає дійсні значення:

$$f(x) = \begin{cases} -x, & \text{если } x \leq 0, \\ x^2, & \text{если } 0 < x < 2, \\ 4, & \text{если } x \geq 2. \end{cases}$$

Рис.1 - Завдання 1

Завдання 2.

Дано координати точки на площині (x, y) . Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур) і вивести відповідне повідомлення. Геометрична область 30. (Зелена область на Рис.2)

r - радіус кола

- 27 вар.
- 28 вар.
- 29 вар.
- 30 вар.
- 31 вар.

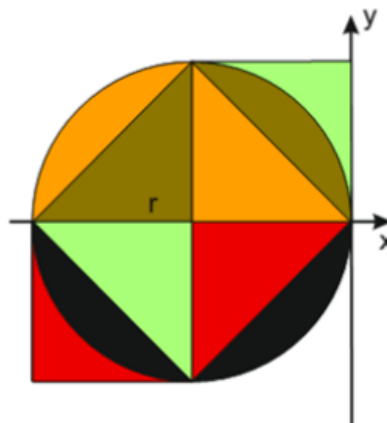


Рис.2 - Постановка завдання 2 та 3

Завдання 3. Обчислити площу і периметр плоскої фігури. (Рис.2)

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 if26

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

Для дійсного x знайти значення функції f , яка приймає дійсні значення за певних умов (Рис.1)

Вихідні дані (ім'я, опис, тип):

Алгоритм вирішення показано на Рис.3

```
main.cpp
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  void task_if26() { // завдання 1 реалізація
6      double x;
7      cout << " if 26 " << endl;
8      cout << "Enter the value of x: ";
9
10     //повідомлення про помилку
11     if (!(cin >> x)) {
12         cout << "Invalid input! Please enter a valid number." << endl;
13         return;
14     }
15
16     double result;
17
18     if (x == 0){ // умова якщо введено 0
19         result = 0;
20     }
21     else if (x <= 0) { // умова якщо введено число менше 0
22         result = -x;
23     }
24     else if (x > 0 && x < 2) { // умова якщо введено від 0 до 2
25         result = x * x;
26     }
27     else {
28         result = 4;
29     }
30
31     cout << "f(x) value is: " << result << endl;
32 }
33
34 }
```

Рис.3 - Вирішення завдання 1

Лістинг коду вирішення задачі наведено в дод. А (стор. 9).

Екран роботи програми показаний на Рис. 4 та Рис. 5.

```
18- if (x == 0){
19-     result = 0;
20-
21- } else if (x <= 0) {
    ...
Task number:26
if 26
Enter the value of x: -8
f(x) value is: 8
sh: 1: pause: not found

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

Рис.4 - Екран роботи програми до завдання 1

```
18- if (x == 0){
19-     result = 0;
20-
21- } else if (x <= 0) {
    ...
Task number:26
if 26
Enter the value of x: 0
f(x) value is: 0
sh: 1: pause: not found

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

Рис. 5 - Екран роботи програми до Завдання 1

Завдання 2.

Вирішення задачі Geom30

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

Координати x,y; коло як загальна площа; фігури, виділені зеленим на Рис.2

Вихідні дані (ім'я, опис, тип):

Повідомлення про те, чи потрапляє задана точка у фігуру зеленого кольору.

Алгоритм вирішення показано на Рис. 6

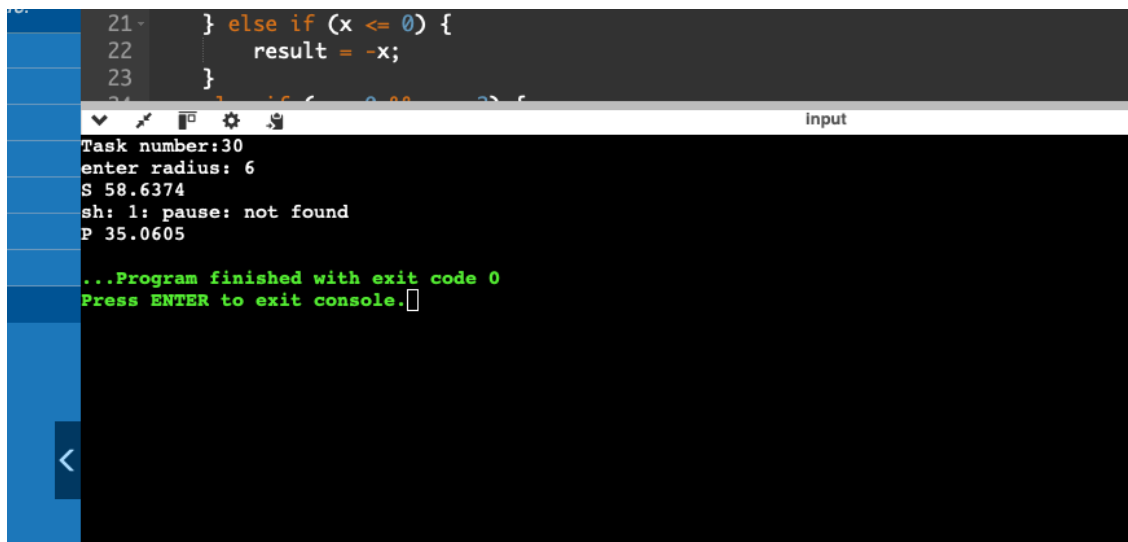
```

34 }
35
36 // завдання 2 реалізація
37 void task_geom30() {
38     #define PI 3.14159265358979323846
39     const double doable = PI;
40
41     cout << "enter radius: ";
42
43     float r;
44     float rightSideArea;
45     float rightSideperimeter;
46     float leftSideArea;
47     float leftSideperimeter;
48     float answerArea;
49     float answerperimeter;
50
51     //повідомлення про помилку
52     if (!(cin >> r)) {
53         cout << "Invalid input! Please enter a valid number." << endl;
54         return;
55     }
56
57     leftSideArea = (r * r) * sqrt(2);
58     leftSideperimeter = (2 * r) + (r * sqrt(2));
59
60     rightSideArea = (r * r) - ((PI * (r * r)) / 4);
61     rightSideperimeter = (4 * r) - (0.5 * PI * r);
62
63
64     answerArea = leftSideArea + rightSideArea;
65     answerperimeter = leftSideperimeter + rightSideperimeter;
66
67     cout << "S " << answerArea << "\n" << "P " << answerperimeter;
68 }

```

Рис. 6 - Алгоритм вирішення Завдання 2

Лістинг коду вирішення задачі geom30 наведено в дод. Б (сторінка 10).
Екран роботи програми показаний на Рис.7 та Рис.8.



```

21 } else if (x <= 0) {
22     result = -x;
23 }
...
Task number:30
enter radius: 6
S 58.6374
sh: 1: pause: not found
P 35.0605

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

```

Рис.7 - Екран роботи програми до Завдання 2

```
18- if (x == 0){
19-     result = 0;
20-
21- } else if (x <= 0) {
    input
Task number:30
enter radius: 82
S 10952.2
sh: 1: pause: not found
P 479.16
...Program finished with exit code 0
Press ENTER to exit console.
```

Рис.8 - Екран роботи програми до Завдання 2

Завдання 3.

Вирішення задачі Geom30

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

Радіус r

Вихідні дані (ім'я, опис, тип):

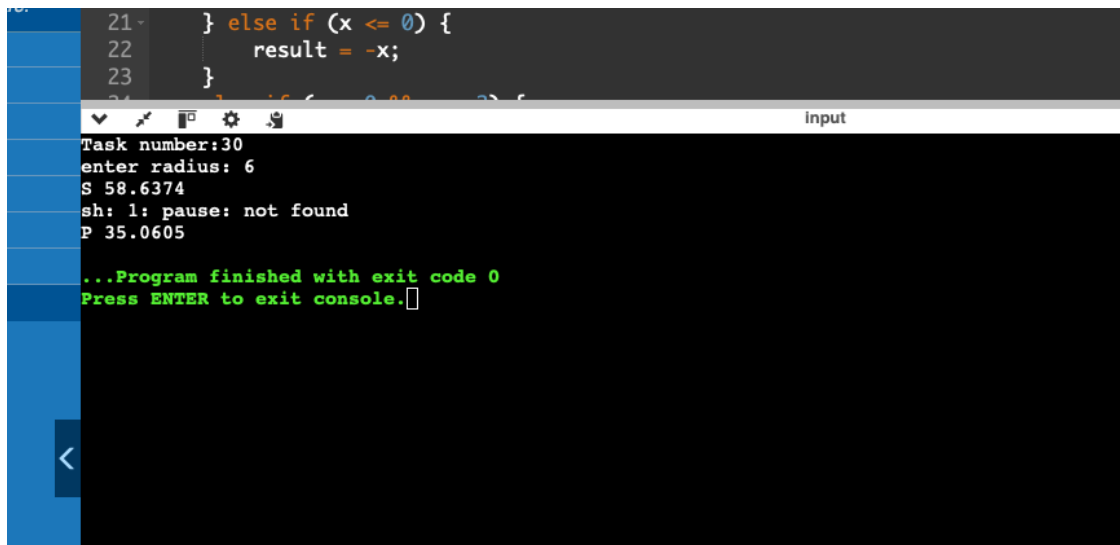
Величина периметру P та площі S фігури.

Алгоритм вирішення показано на Рис. 9.

```
main.cpp
34- }
35-
36- // завдання 2 реалізація
37- void task_geom30() {
38-     #define PI 3.14159265358979323846
39-     const double doable = PI;
40-
41-     cout << "enter radius: ";
42-
43-     float r;
44-     float rightSideArea;
45-     float rightSideperimeter;
46-     float leftSideArea;
47-     float leftSideperimeter;
48-     float answerArea;
49-     float answerperimeter;
50-
51-     //повідомлення про помилку
52-     if (!(cin >> r)) {
53-         cout << "Invalid input! Please enter a valid number." << endl;
54-         return;
55-     }
56-
57-     leftSideArea = (r * r) * sqrt(2);
58-     leftSideperimeter = (2 * r) + (r * sqrt(2));
59-
60-     rightSideArea = (r * r) - ((PI * (r * r)) / 4);
61-     rightSideperimeter = (4 * r) - (0.5 * PI * r);
62-
63-
64-     answerArea = leftSideArea + rightSideArea;
65-     answerperimeter = leftSideperimeter + rightSideperimeter;
66-
67-     cout << "S " << answerArea << "\n" << "P " << answerperimeter;
68- }
```

Рис.9 - Алгоритм вирішення Завдання 3

Лістинг коду вирішення задачі geom30 наведено в дод. Б(стор. 10).
Екран роботи програми показаний на Рис. 10.



```
21 } else if (x <= 0) {
22     result = -x;
23 }
24
Task number:30
enter radius: 6
S 58.6374
sh: 1: pause: not found
P 35.0605
...Program finished with exit code 0
Press ENTER to exit console.
```

Рис.10 - Екран роботи програми до Завдання 3

Завдання 4

Алгоритм вибору користувачем завдання

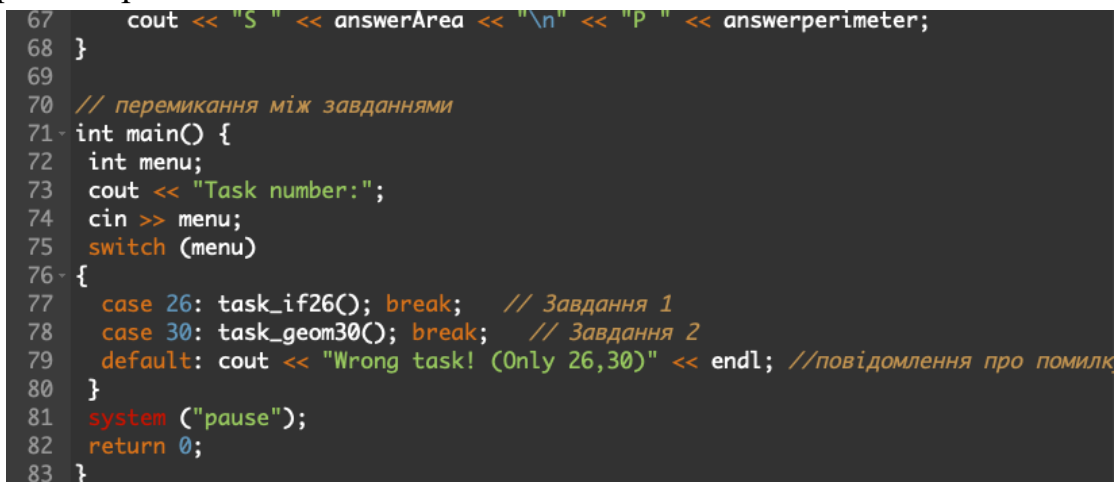
Вхідні дані (ім'я, опис, тип, обмеження):

Попередні завдання

Вихідні дані (ім'я, опис, тип):

Програма, що дозволяє користувачу переключатися між завданнями.

Алгоритм вирішення показано на Рис. 11.



```
67     cout << "S " << answerArea << "\n" << "P " << answerperimeter;
68 }
69
70 // перемикання між завданнями
71 int main() {
72     int menu;
73     cout << "Task number:";
74     cin >> menu;
75     switch (menu)
76     {
77     case 26: task_if26(); break;    // Завдання 1
78     case 30: task_geom30(); break; // Завдання 2
79     default: cout << "Wrong task! (Only 26,30)" << endl; //повідомлення про помилк
80     }
81     system ("pause");
82     return 0;
83 }
```

Рис. 11 - Алгоритм вирішення завдання 11

Лістинг коду вирішення задачі geom30 наведено в дод. В (стор. 11).

ВИСНОВКИ

У ході виконання цієї лабораторної роботи було опрацьовано теоретичні аспекти синтаксису мови C++ та розробки UML-діаграм активності для алгоритмів із розгалуженням. Реалізовано алгоритми з використанням інструкцій умовного переходу та вибору у середовищі Visual Studio. Окрім цього, було відпрацьовано навички модульного програмування шляхом структурування програм із використанням функцій.

Додаток А

Лістинг коду програми виконання завдання 1

```
void task_if26() { // завдання 1 реалізація
    double x;
    cout << " if 26 " << endl;
    cout << "Enter the value of x: ";

    //повідомлення про помилку
    if (!(cin >> x)) {
        cout << "Invalid input! Please enter a valid number." << endl;
        return;
    }

    double result;

    if (x == 0){           // умова якщо введено 0
        result = 0;

    } else if (x <= 0) {   // умова якщо введено число менше 0
        result = -x;
    }
    else if (x > 0 && x < 2) { // умова якщо введено від 0 до 2
        result = x * x;
    }
    else {
        result = 4;
    }

    cout << "f(x) value is: " << result << endl;

}
```

Додаток Б

```
// завдання 2 реалізація
void task_geom30() {
    #define PI 3.14159265358979323846
    const double doable = PI;

    cout << "enter radius: ";

    float r;
    float rightSideArea;
    float rightSideperimeter;
    float leftSideArea;
    float leftSideperimeter;
    float answerArea;
    float answerperimeter;

    //повідомлення про помилку
    if (!(cin >> r)) {
        cout << "Invalid input! Please enter a valid number." << endl;
        return;
    }

    leftSideArea = (r * r) * sqrt(2);
    leftSideperimeter = (2 * r) + (r * sqrt(2));

    rightSideArea = (r * r) - ((PI * (r * r)) / 4);
    rightSideperimeter = (4 * r) - (0.5 * PI * r);

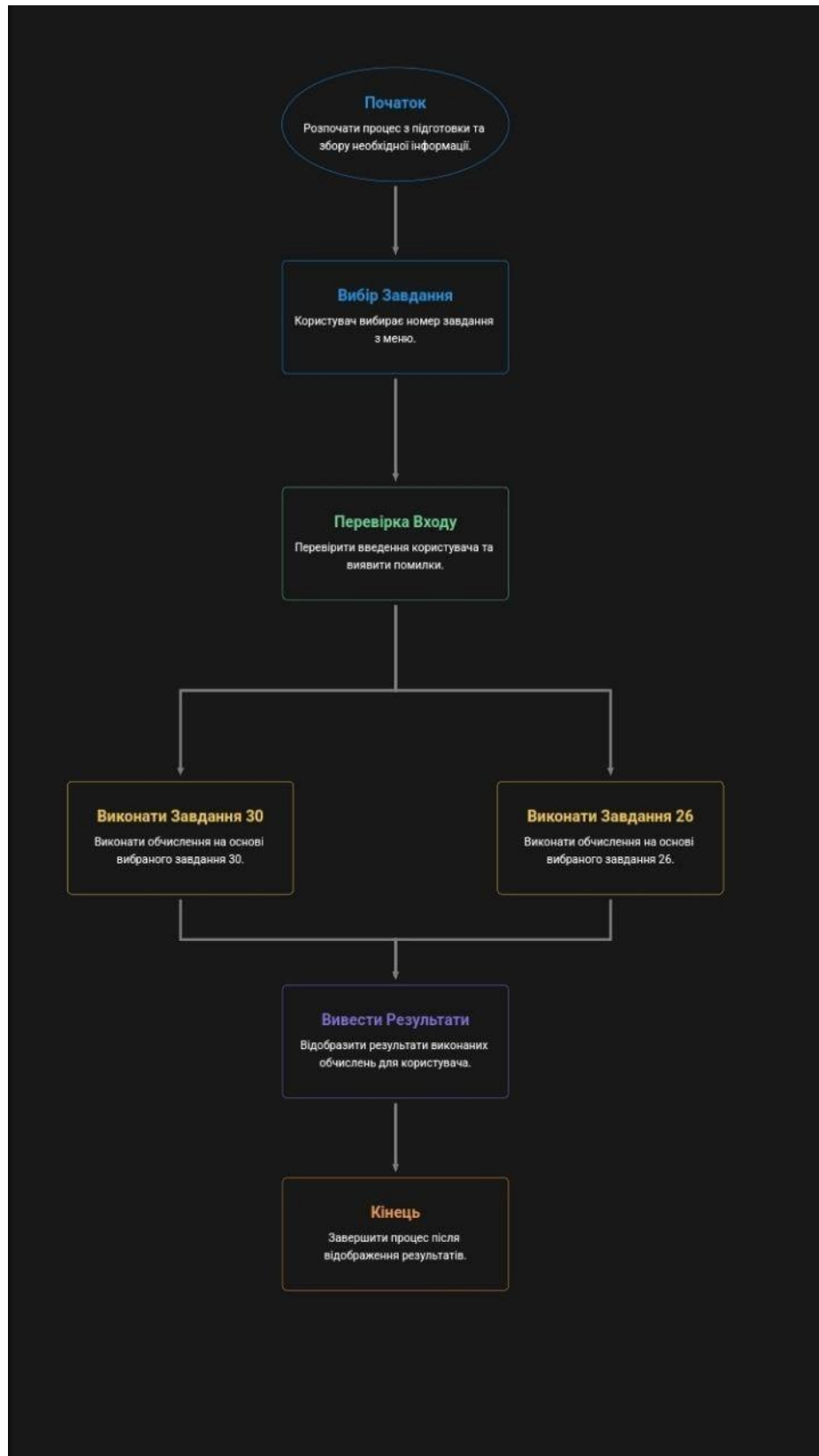
    answerArea = leftSideArea + rightSideArea;
    answerperimeter = leftSideperimeter + rightSideperimeter;

    cout << "S " << answerArea << "\n" << "P " << answerperimeter;
}
```

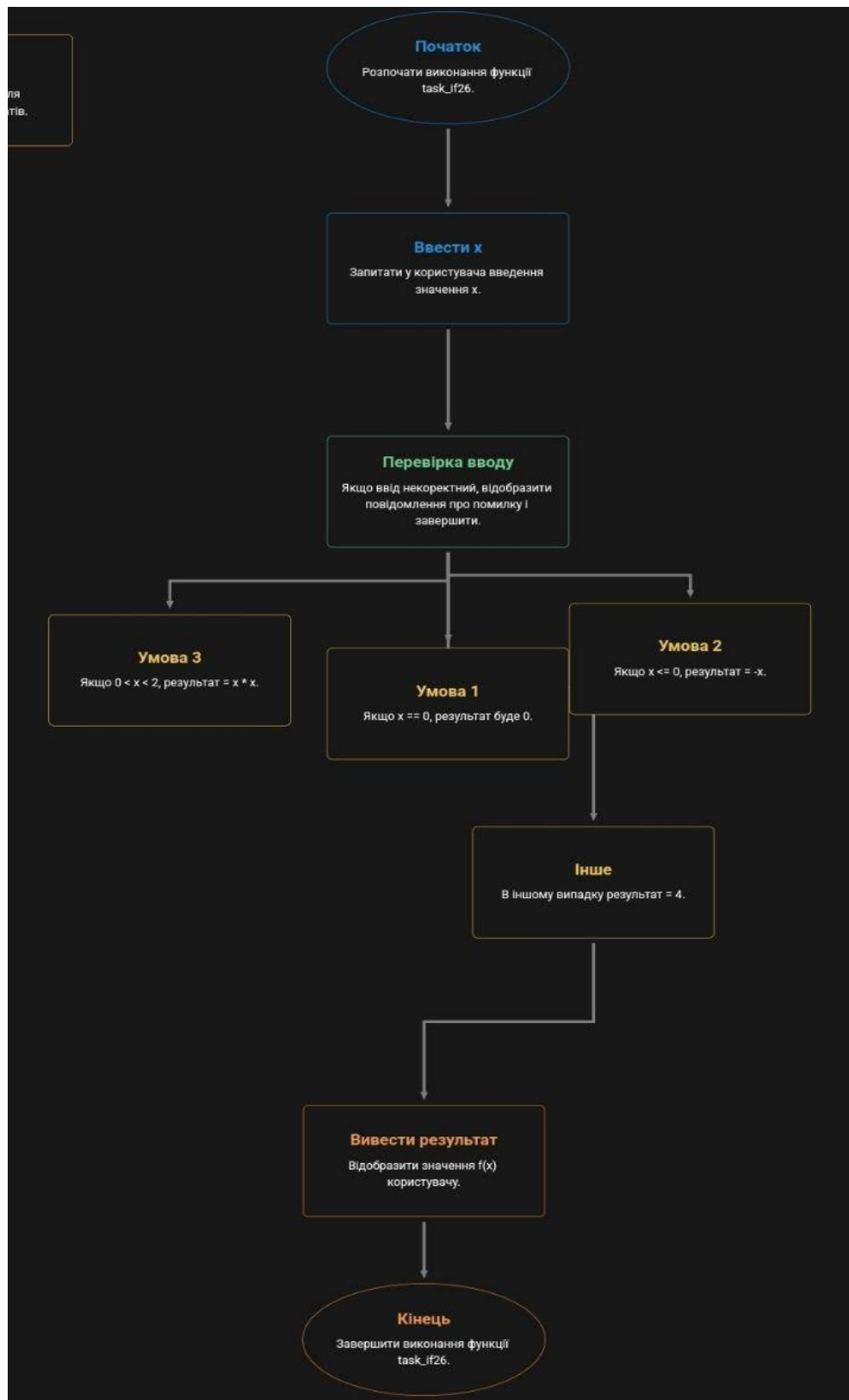
Додаток В

```
// перемикання між завданнями
int main() {
int menu;
cout << "Task number:";
cin >> menu;
switch (menu)
{
case 26: task_if26(); break; // Завдання 1
case 30: task_geom30(); break; // Завдання 2
default: cout << "Wrong task! (Only 26,30)" << endl; //повідомлення про помилку
}
system ("pause");
return 0;
}
```

Діаграма активності всієї програми



Діаграма активності програми завдання 1



Діаграма активності програми завдань 2 та 3

