

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.301.272.311.1 Лр

Виконала студентка гр. _____ 311 _____

_____ Ніка Коробейникова _____
(підпис, дата) (П.І.Б.)

Перевірів

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

2023

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. Варіанти представлено в табл.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. Варіанти фігур представлено в табл.2.

Завдання 3. Обчислити площу і периметр плоскої фігури. Варіанти представлено в табл.2.

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

використанням інструкції вибору.

Для розв'язання задачі 1f27, де потрібно обчислити значення функції, яка приймає цілі значення залежно від дійсного аргументу, можна скористатися наступним підходом.

Аналіз інтервалів:

1. Для :

належить інтервалу $[0,1)$ тоді, коли належить інтервалу $(-1,0]$.

належить інтервалу $[2,3)$ тоді, коли належить інтервалу $(-3,-2]$.

Загалом, належить інтервалу $[2k, 2k+1)$, де — ціле невід'ємне число, коли належить інтервалу $[-(2k+1), -2k]$.

2. Для :

належить інтервалу $[1,2)$ або $[3,4)$ і так далі, тобто $[2k+1, 2k+2)$, де — ціле невід'ємне число.

Алгоритм:

1. Якщо належить інтервалу $[-(2k+1), -2k]$, то .

2. Якщо належить інтервалу $[2k+1, 2k+2)$, то .

3. В інших випадках не визначена за умовою задачі.

Реалізація на C++:

```
#include <iostream>
```

```
#include <cmath>
```

```
int f(double x) {  
    // Перевірка для  $f(x) = 0$   
    for (int k = 0; ; ++k) {  
        double left =  $-(2 * k + 1)$ ;  
        double right =  $-2 * k$ ;
```

```

    if (x >= left && x < right) {
        return 0;
    }
    // Якщо x менше за left, немає сенсу продовжувати перевірку
    if (x < left) {
        break;
    }
}

// Перевірка для f(x) = 1
for (int k = 0; ; ++k) {
    double left = 2 * k + 1;
    double right = 2 * k + 2;
    if (x >= left && x < right) {
        return 1;
    }
    // Якщо x менше за left, немає сенсу продовжувати перевірку
    if (x < left) {
        break;
    }
}

// Якщо x не належить жодному з інтервалів
std::cerr << "Значення x не належить жодному з визначених інтервалів." << std::endl;
return -1; // Повертаємо -1 як індикатор невизначеності
}

int main() {
    double x;
    std::cout << "Введіть значення x: ";
    std::cin >> x;

    int result = f(x);
    if (result != -1) {
        std::cout << "f(" << x << ") = " << result << std::endl;
    } else {
        std::cout << "f(" << x << ") не визначена для цього значення." << std::endl;
    }

    return 0;
}

```

Пояснення коду:

Функція $f(\text{double } x)$ обчислює значення на основі аналізу інтервалів.

Використовуються цикли для перевірки належності відповідним інтервалам.

Якщо не належить жодному з визначених інтервалів, функція повертає -1 і виводить повідомлення про невизначеність.

Висновки:

Було розроблено програму на C++ для обчислення значення функції відповідно до заданих умов. Закріплено навички роботи з умовними операторами та циклами для перевірки належності значення до певних інтервалів.

29

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
// Функція для обчислення значення f(x)
```

```
int calculateFunction(double x) {
```

```
    // Перевіряємо умови для f(x) = 0
```

```
    if (fmod(-x, 2.0) >= 0 && fmod(-x, 2.0) < 1.0) {
```

```
        return 0; // -x належить [0, 1), [2, 3), ...
```

```
    }
```

```
    // Перевіряємо умови для f(x) = 1
```

```
    if (fmod(x, 2.0) >= 1 && fmod(x, 2.0) < 2.0) {
```

```
        return 1; // x належить [1, 2), [3, 4), ...
```

```
    }
```

```
    return -1; // Якщо x не відповідає жодній умові
```

```
}
```

```
int main() {
```

```
    double x;
```

```
    // Введення значення x
```

```
    cout << "Введіть значення x: ";
```

```
    cin >> x;
```

```
    // Обчислення значення функції
```

```
    int result = calculateFunction(x);
```

```
    // Виведення результату
```

```
    if (result != -1) {
```

```
        cout << "Значення f(x): " << result << endl;
```

```
    } else {
```

```
        cout << "Значення x не відповідає умовам задачі." << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Пояснення

1. Функція calculateFunction:

Перевіряє умови належності та до заданих інтервалів.

Використовує функцію fmod для роботи з остачею від ділення.

Повертає 0, якщо належить до інтервалів [0,1), [2,3), тощо.

Повертає 1, якщо належить до інтервалів [1,2), [3,4), тощо.

Якщо не належить жодному з інтервалів, повертає -1.

2. Основна функція main:

Запитує у користувача значення .

Викликає функцію calculateFunction для обчислення .

Виводить результат або повідомлення про невідповідність умовам задачі.

Висновки

Було виконано:

Реалізовано програму на C++ для обчислення значення функції , що приймає цілі значення.

Закріплено на практиці роботу з умовними операторами, математичними функціями, та перевіркою належності значень до інтервалів.

Додаток А

Лістинг коду програми (за завданням 29)

```
#include <iostream>
```

```

#include <cmath>

using namespace std;

int main() {
    double r; // Радіус кола

    // Введення радіуса
    cout << "Введіть радіус кола (r > 0): ";
    cin >> r;

    if (r <= 0) {
        cout << "Радіус повинен бути більше 0!" << endl;
        return 1;
    }

    // Кут сектора для чорного сегмента (90 градусів у радіанах)
    double theta = M_PI / 2; //  $\pi/2$  радіан

    // Обчислення площі сектора
    double sectorArea = (theta / 2.0) * r * r;

    // Обчислення площі трикутника
    double triangleArea = (1.0 / 2.0) * r * r;

    // Обчислення площі сегмента (чорна частина)
    double segmentArea = sectorArea - triangleArea;

    // Виведення результату
    cout << "Площа чорного сегмента: " << segmentArea << endl;

    return 0;
}

```

Пояснення

1. Значення :

Для чорного сегмента на рисунку центральний кут дорівнює $\pi/2$ або 90° радіан.

2. Обчислення:

Площа сектора обчислюється за формулою .

Площа трикутника дорівнює , оскільки кут прямий.

Чорний сегмент: .

3. Вивід: Програма виводить площу чорного сегмента для заданого радіуса .

(27)

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
int main() {
```

```
    double r; // Радіус кола
```

```
    // Введення радіуса
```

```
    cout << "Введіть радіус кола (r > 0): ";
```

```
    cin >> r;
```

```
    if (r <= 0) {
```

```
        cout << "Радіус повинен бути більше 0!" << endl;
```

```
        return 1;
```

```
    }
```

```
    // Площа круга
```

```
    double circleArea = M_PI * r * r;
```

```
    // Площа жовтого сегмента (1/4 площі кола)
```

```
    double yellowSegmentArea = circleArea / 4;
```

```
    // Виведення результату
```

```
    cout << "Площа жовтого сегмента: " << yellowSegmentArea << endl;
```

```
    return 0;
```

```
}
```

Пояснення

1. Розрахунок площі круга:

Формула для площі круга:

$$A_{\text{круг}} = \pi r^2$$

2. Жовтий сегмент:

Оскільки жовтий сегмент займає рівно чверть круга, його площа:

$$A_{\text{жовтий сегмент}} = \frac{1}{4} A_{\text{круг}}$$

3. Перевірка вводу:

Якщо радіус , програма видасть помилку.

4. Вивід результату:

Програма обчислює площу жовтого сегмента та виводить її значення.