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 С ++"»

ХАІ.301.175.318.13 ЛР

Виконав студент гр. № 319а

Михайло ТЮТЮННИК

(підпис, дата) (П.І.Б.)

Перевірив к.т.н., доцент

( вчена ступінь, вчене звання)

Олена ГАВРИЛЕНКО (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. If19. Вирішити завдання на алгоритми з розгалуженням.

If19. Дано чотири цілих числа, одне з яких відмінно від трьох інших, рівних між собою.Визначити порядковий номер числа, відмінного від інших.

Завдання 2. Дано координати точки на площині (x, y). Визначити, чи

потрапляє точка в фігуру заданого кольору (або групу фігур) і вивести відповідне повідомлення. Обчислити площу і периметр плоскої фігури.

Таблиця 2– Геометричні фігури №31

Изображение выглядит как диаграмма, снимок экрана, Красочность, линия

Автоматически созданное описание

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 If19.

Вхідні дані:

Ім'я змінної: a

Опис: Перше число з чотирьох введених.

Тип: Ціле число (int).

Обмеження: Будь-яке допустиме ціле число.

Ім'я змінної: b

Опис: Друге число з чотирьох введених.

Тип: Ціле число (int).

Обмеження: Будь-яке допустиме ціле число.

Ім'я змінної: c

Опис: Третє число з чотирьох введених.

Тип: Ціле число (int).

Обмеження: Будь-яке допустиме ціле число.

Ім'я змінної: d

Опис: Четверте число з чотирьох введених.

Тип: Ціле число (int).

Обмеження: Будь-яке допустиме ціле число.

Вихідні дані:

Ім'я змінної: odd number position

Опис: Позиція числа, яке відрізняється від інших трьох.

Тип: Ціле число (int).

Обмеження: Значення в межах [1; 4], якщо "особливе" число знайдено; або повідомлення "No odd number found!", якщо всі числа однакові.

Алгоритм:

Введення значень чисел:

Запитати у користувача чотири цілі числа.

Вивести повідомлення: "Enter four integers (separated by spaces):"

Зчитати значення змінних a, b, c, d.

Перевірка вводу:

Якщо ввід некоректний (не цілі числа), вивести повідомлення: "Wrong input! Must be integers."

Завершити виконання функції.

Логіка визначення "особливого" числа:

Перевірити, яке з чисел a, b, c, d відрізняється від інших.

Якщо a відрізняється, результат: "The odd number is at position: 1".

Якщо b відрізняється, результат: "The odd number is at position: 2".

Якщо c відрізняється, результат: "The odd number is at position: 3".

Якщо d відрізняється, результат: "The odd number is at position: 4".

Обробка випадку, якщо всі числа однакові:

Вивести повідомлення: "No odd number found!".

Виведення результату:

Вивести позицію "особливого" числа (або повідомлення про його відсутність).

Лістинг коду до завдання 1 (If19.) наведено в дод. А (стор. 7).

Зображеня вікна виконання роботи наведено в дод. Б(стор. 9, рис. 1 , 2)

Завдання 2

Вирішення задачі Таблиця 2 – Геометричні фігури

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

1.Ім'я змінної: x;

2.Опис: Координата точки по осі X;

3.Тип: Дійсне число (double);

4.Обмеження: Немає;

5.Ім'я змінної: y;

6.Опис: Координата точки по осі Y;

7.Тип: Дійсне число (double);

8.Обмеження: Немає;

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

1.Ім'я змінної: isInFourthQuadrant 2.Опис: Логічне значення, що визначає, чи точка з координатами (x, y) лежить в четвертій координатній чверті. 3.Тип: Логічний тип (bool).

Алгоритм:

1. Введення значення x. Вивести запит для введення числа: "Enter the x-coordinate: ".
2. Введення значення y. Вивести запит для введення числа: "Enter the y-coordinate: ".
3. Перевірка, чи точка знаходиться в четвертій координатній чверті. Встановити логічну змінну isInFourthQuadrant = (x > 0 && y < 0).
4. Якщо значення isInFourthQuadrant є істинним (точка лежить у четвертій чверті), вивести повідомлення: "The point (x, y) is in the fourth quadrant.".
5. Якщо значення isInFourthQuadrant є хибним (точка не лежить у четвертій чверті), вивести повідомлення: "The point (x, y) is not in the fourth quadrant.".

Лістинг коду до завдання 1 (Таблиця 2– Геометричні фігури) наведено в дод. A (стор. 7).

Зображеня вікна виконання роботи наведено в дод. Б (стор. 9, рис. 3 , 4)

Завдання 3

Вирішення задачі Таблиця 2 – Геометричні фігури

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

1. Ім'я змінної: a  
   Опис: Перша сторона трикутника.  
   Тип: Дійсне число (double).  
   Обмеження: Сторона повинна бути позитивною (> 0).
2. Ім'я змінної: b  
   Опис: Друга сторона трикутника.  
   Тип: Дійсне число (double).  
   Обмеження: Сторона повинна бути позитивною (> 0).
3. Ім'я змінної: c  
   Опис: Третя сторона трикутника.  
   Тип: Дійсне число (double).  
   Обмеження: Сторона повинна бути позитивною (> 0).

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

1. Ім'я змінної: area  
   Опис: Площа трикутника.  
   Тип: Дійсне число (double).  
   Обмеження: Має бути більше або дорівнювати 0.
2. Ім'я змінної: perimeter  
   Опис: Периметр трикутника.  
   Тип: Дійсне число (double).  
   Обмеження: Має бути більше 0.

Алгоритм:

1. Введення значення сторони a. Вивести запит для введення числа: "Enter the first side (a): ".
2. Введення значення сторони b. Вивести запит для введення числа: "Enter the second side (b): ".
3. Введення значення сторони c. Вивести запит для введення числа: "Enter the third side (c): ".
4. Перевірити, чи всі сторони додатні (a > 0, b > 0, c > 0). Якщо ні, вивести повідомлення: "All sides must be positive!" і завершити алгоритм.
5. Перевірити, чи можливо утворити трикутник із введеними сторонами (a + b > c, a + c > b, b + c > a). Якщо ні, вивести повідомлення: "A triangle with such sides cannot exist!" і завершити алгоритм.
6. Обчислити периметр трикутника: perimeter = a + b + c.
7. Вивести повідомлення: "Perimeter of the triangle: <значення периметра>".
8. Обчислити площу трикутника за формулою Герона:
   * Обчислити напівпериметр: s = perimeter / 2.
   * Обчислити площу: area = sqrt(s \* (s - a) \* (s - b) \* (s - c)).
9. Вивести повідомлення: "Area of the triangle: <значення площі>".

Лістинг коду до завдання 1 (Таблиця 2– Геометричні фігури) наведено в дод. A (стор. 7).

Зображеня вікна виконання роботи наведено в дод Б (стор. 9, рис. 5, 6)

Висновки

Отже, було вивчено синтаксис мови C++ та реалізовано алгоритми з умовними переходами. Створено програму для обробки даних і перевірки умов за допомогою бібліотеки cmath. Також опрацьовано створення UML-діаграм активності для відображення алгоритмів з розгалуженням. Усі завдання були реалізовані в середовищі Visual Studio, що сприяло покращенню навичок програмування та структуризації програм.

ДОДАТОК А

Лістинг коду програми

#include <iostream>

#include <cmath> // Для використання математичних функцій

using namespace std;

// Прототипи функцій

void task\_if19(); // Завдання 1: знайти "особливе" число (If19)

void task\_table2(); // Завдання 2: перевірка точки у фігурі (Table 2 - Geometric figures №31)

void task\_geometry(); // Завдання 3: обчислення площі та периметру плоскої фігури

int main() {

int menu; // Змінна для вибору завдання

cout << "Task number (1 - If19, 2 - Table 2 - Geometric figures №31, 3 - Geometry - Area and Perimeter): ";

cin >> menu; // Введення номера завдання

// Перемикання між завданнями за допомогою оператора switch

switch (menu) {

case 1:

task\_if19(); // Виклик функції для завдання 1

break;

case 2:

task\_table2(); // Виклик функції для завдання 2

break;

case 3:

task\_geometry(); // Виклик функції для завдання 3

break;

default:

// Якщо введено неправильний номер завдання, вивести повідомлення

cout << "Wrong task! (Only 1, 2, 3)" << endl;

}

system("pause"); // Затримка перед завершенням програми (корисно в Windows)

return 0; // Завершення програми

}

// Завдання 1: знайти "особливе" число серед 4 введених (If19)

void task\_if19() {

int a, b, c, d; // Чотири числа для перевірки

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\* Task 1: If19 - Odd Number \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Enter four integers (separated by spaces): ";

cin >> a >> b >> c >> d; // Введення чотирьох чисел

// Перевірка, чи ввід був коректним

if (!cin) {

cout << "Wrong input! Must be integers." << endl;

}

else {

// Перевірка, яке з чисел є "особливим" (відрізняється від інших)

if (a != b && a != c && a != d)

cout << "The odd number is at position: 1" << endl;

else if (b != a && b != c && b != d)

cout << "The odd number is at position: 2" << endl;

else if (c != a && c != b && c != d)

cout << "The odd number is at position: 3" << endl;

else if (d != a && d != b && d != c)

cout << "The odd number is at position: 4" << endl;

else

// Якщо всі числа однакові, "особливого" числа немає

cout << "No odd number found!" << endl;

}

}

// Завдання 2: перевірка належності точки до певної області (Table 2 - Geometric figures №31)

void task\_table2() {

float x, y, r; // Координати точки та радіус круга

cout << "\*\*\*\*\*\*\*\*\*\*\* Task 2: Table 2 - Geometric figures №31 \*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Radius r: ";

cin >> r; // Введення радіуса

// Перевірка коректності введення радіуса

if (!cin || r <= 0) {

cout << "Radius must be numeric and positive!" << endl;

return; // Завершити функцію, якщо ввід некоректний

}

cout << "Point x, y: ";

cin >> x >> y; // Введення координат точки

// Перевірка коректності введення координат точки

if (!cin) {

cout << "Coordinates must be numeric!" << endl;

return; // Завершити функцію, якщо ввід некоректний

}

// Логіка перевірки належності точки до коричневої області у 1-й чверті

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

cout << "In brown region!" << endl;

}

else {

cout << "Out of brown region!" << endl;

}

}

// Завдання 3: обчислення площі та периметру плоскої фігури

void task\_geometry() {

float a, b, c; // Сторони трикутника

cout << "\*\*\*\*\*\*\*\*\*\*\* Task 3: Geometry - Area and Perimeter \*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Enter lengths of three sides of a triangle (a, b, c): ";

cin >> a >> b >> c; // Введення довжин сторін

// Перевірка коректності введення сторін

if (!cin || a <= 0 || b <= 0 || c <= 0) {

cout << "Sides must be numeric and positive!" << endl;

return;

}

// Перевірка чи трикутник існує

if (a + b <= c || a + c <= b || b + c <= a) {

cout << "Triangle with such sides does not exist!" << endl;

return;

}

// Обчислення периметра

float P = a + b + c;

cout << "Perimeter: " << P << endl;

// Обчислення площі за формулою Герона

float s = P / 2; // Напівпериметр

float area = sqrt(s \* (s - a) \* (s - b) \* (s - c));

cout << "Area: " << area << endl;

}

ДОДАТОК Б

Скрін-шоти вікна виконання програми

На рисунку 1 , 2 зображена робота програми до завдання If 19

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 1 — Результат роботи програми

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 2 — Результат роботи програми

На рисунку 3 , 4 зображена робота програми до завдання Таблиця – Геометричні фігури варіант № 31

Изображение выглядит как текст, Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Рисунок 3 — Результат роботи програми

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 4 — Результат роботи програми

На рисунку 5 , 6 зображена робота програми до завдання Таблиця – Геометричні фігури варіант № 31 обчисленя площі і перимета плоскої фігури.

Изображение выглядит как текст, программное обеспечение, снимок экрана, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 5 - Результат роботи програми

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 6 - Результат роботи програми

Додаток С

Діаграми активності програм

На рисунку 1 зображено діаграму активності всієї програми

Изображение выглядит как текст, диаграмма, План, линия

Автоматически созданное описание

Рис.1 Діаграма активності програми

На рисунку 2 зображено діаграму активності програми до завдання If19

Изображение выглядит как текст, диаграмма, рисунок

Автоматически созданное описание

Рис.2 Діаграма активності програми

На рисунку 3 зображено діаграму активності програми до завдання If19

Изображение выглядит как текст, диаграмма, Технический чертеж, План

Автоматически созданное описание

Рис.3 Діаграма активності програми