

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 № 4

з дисципліни «Алгоритмізація та програмування»  
Тема: "Структурування програм з використанням функцій"

XAI.301. 141. 319a. 19 ЛР

Виконав студент гр. \_\_\_\_\_ 319a \_\_\_\_\_

\_\_\_\_\_ Овчинніков Станіслав \_\_\_\_\_

(підпис, 20.11.2024)

(П.І.Б.)

Перевірив

\_\_\_\_\_ к.т.н., доц. Олена

ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

## МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису оголошення, визначення і виклику функцій в C++ і реалізувати консольний додаток з використанням функцій з параметрами і поверненням результату на мові програмування C++ в середовищі Visual Studio.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1 ( Proc27 ) . Описати функцію (декларація, визначення і виклик) відповідно до варіанту. У тілі і при виклику функцій не використовувати цикли. Варіанти наведено в табл.1.

Завдання 2 (Boolean 30 ) . Для вирішення завдання з логічними змінними відповідно до варіанту визначити дві функції:

- 1) функцію введення і перевірки вхідних даних на коректність;
- 2) функцію розрахунку на підставі коректних вхідних даних результату (false / true). При виклику функцій в разі введення некоректних вхідних даних вивести відповідне повідомлення і розрахунки не виконувати. Варіанти представлено в табл.2.

Завдання 3.(Integer 2 ) Для вирішення завдання з цілочисельними змінними відповідно до варіанту визначити три функції:

- 1) функцію введення вхідних значень з консолі з перевіркою їх на коректність;
  - 2) функцію підрахунку результату;
  - 3) функцію виведення результату в консоль.
- При виклику функцій в разі введення некоректних вхідних даних вивести відповідне повідомлення і розрахунки не виконувати. Варіанти представлено в табл.3

Завдання 4. Для вибору користувачем одного з трьох зазначених вище завдань організувати меню з використанням інструкції вибору. Кожне завдання має бути реалізовано у вигляді окремої процедури (функції без параметрів), що містить необхідні оголошення змінних і виклики інших функцій.

## ВИКОНАННЯ РОБОТИ

Завдання 1 Proc27

Описати функцію RectP (x1, y1, x2, y2) дійсного типу, яка обчислює периметр прямокутника зі сторонами, паралельними осям координат, за дійсними координатами (x1, y1), (x2, y2) його протилежних вершин. За допомогою цієї функції знайти периметри трьох прямокутників з даними протилежними вершинами.

Вхідні дані (ім'я, опис, тип): x1, y1, x2, y2) - координати прямокутника, double; вихід: периметр прямокутника, double.

Вихідні дані (ім'я, опис, тип): (периметр) - double.

### Алгоритм вирішення

Запросить у пользователя значения переменных x1, y1, x2, y2.  
Проверить, входят ли значения в диапазон [-100, 100].  
Если не входят, вывести сообщение об ошибке и завершить выполнение.  
Рассчитать ширину прямоугольника как  $\text{abs}(x2 - x1)$ .  
Рассчитать высоту прямоугольника как  $\text{abs}(y2 - y1)$ .  
Найти периметр прямоугольника по формуле:  
Периметр =  $2 \cdot (\text{ширина} + \text{высота})$ .  
Вывести значение периметра.  
Завершить выполнение задачи.

```
// --- Task 1: Calculate the perimeter of a rectangle ---
double RectP(double x1, double y1, double x2, double y2) {
    // Check the range of input data
    if (x1 < -100 || x1 > 100 || y1 < -100 || y1 > 100 ||
        x2 < -100 || x2 > 100 || y2 < -100 || y2 > 100) {
        cout << "Error: Coordinates must be within the range [-100, 100]!\n";
        return -1; // Return an error value
    }
    // Calculate width, height, and perimeter
    double width = abs(x2 - x1);
    double height = abs(y2 - y1);
    return 2 * (width + height);
}
```

## Рисунок 1 – Proc 27

Лістинг коду вирішення задачі :

```
// --- Task 1: Calculate the perimeter of a rectangle ---
double RectP(double x1, double y1, double x2, double y2) {
    // Check the range of input data
    if (x1 < -100 || x1 > 100 || y1 < -100 || y1 > 100 ||
        x2 < -100 || x2 > 100 || y2 < -100 || y2 > 100) {
        cout << "Error: Coordinates must be within the range [-100, 100]!\n";
        return -1; // Return an error value
    }
    // Calculate width, height, and perimeter
    double width = abs(x2 - x1);
    double height = abs(y2 - y1);
    return 2 * (width + height);
}
```

```
Select a task to execute:
1. Task 1: Calculate the perimeter of rectangles
2. Task 2: Check if a triangle is equilateral
3. Task 3: Calculate the number of full tons
0. Exit
Your choice: 1
Enter the coordinates of 3 rectangles (x1, y1, x2, y2):
Rectangle 1: 2 3 5 6
Perimeter: 12
Rectangle 2: 2
3
4
5
Perimeter: 8
Rectangle 3: 6
7
8
9
Perimeter: 8
```

Екран роботи програми показаний на рис 2.

## Завдання 2 ( Boolean 30 )

Дано цілі числа  $a$ ,  $b$ ,  $c$ , що є сторонами деякого трикутника. Перевірити істинність висловлювання: «Трикутник з сторонами  $a$ ,  $b$ ,  $c$  є рівностороннім».

Вхідні дані (ім'я, опис, тип, обмеження): ( $a$ ,  $b$ ,  $c$ ) - длины сторон треугольника, `int`.

Вихідні дані: ( равносторонний: `true/false`) - `bool`.

### Алгоритм вирішення:

Запросить у пользователя значения переменных  $a$ ,  $b$ ,  $c$ .

Проверить, входят ли значения в диапазон  $[-100, 100]$ .

Если не входят, вывести сообщение об ошибке и завершить выполнение.

Проверить, равны ли все три стороны треугольника:  $a=b$  и  $b=c$

Если равны, вывести сообщение: "Треугольник равносторонний".

Если не равны, вывести сообщение: "Треугольник не равносторонний".

Завершить выполнение задачи.



```
// --- Task 2: Check if the triangle is equilateral ---
bool isEquilateral(int a, int b, int c) {
    // Check the range of input data
    if (a < -100 || a > 100 || b < -100 || b > 100 || c < -100 || c > 100) {
        cout << "Error: Side lengths must be within the range [-100, 100]!\n";
        return false; // Treat the triangle as invalid
    }
    // Check if all sides are equal
    return a == b && b == c;
}
```

Рисунок 3- код (Boolean 30 )

Лістинг коду :

```
// --- Task 2: Check if the triangle is equilateral ---
bool isEquilateral(int a, int b, int c) {
    // Check the range of input data
    if (a < -100 || a > 100 || b < -100 || b > 100 || c < -100 || c > 100) {
        cout << "Error: Side lengths must be within the range [-100, 100]!\n";
        return false; // Treat the triangle as invalid
    }
    // Check if all sides are equal
    return a == b && b == c;
}
```

```
Select a task to execute:
1. Task 1: Calculate the perimeter of rectangles
2. Task 2: Check if a triangle is equilateral
3. Task 3: Calculate the number of full tons
0. Exit
Your choice: 2
Enter the sides of the triangle (a, b, c): 3 8 8
The triangle is not equilateral or has invalid data
```

Екран роботи програми показаний на рис 4.

Завдання 3 ( Integer 2 ) Дана маса М в кілограмах. Використовуючи операцію ділення остачі, знайти кількість повних тонн в ній (1 тонна = 1000 кг).

Вхідні дані (ім'я, опис, тип, обмеження): (М) - маса в кілограмах, int.

Вихідні дані : (количество полных тонн) - int.

Алгоритм вирішення:

Запросить у пользователя значение переменной М (масса в килограммах).

Проверить, входит ли значение в диапазон [-100, 100].

Если не входит, вывести сообщение об ошибке и завершить выполнение.

Рассчитать количество полных тонн: Тонны= $\lfloor M/1000 \rfloor$ .

Вывести количество полных тонн.

Завершить выполнение задачи.

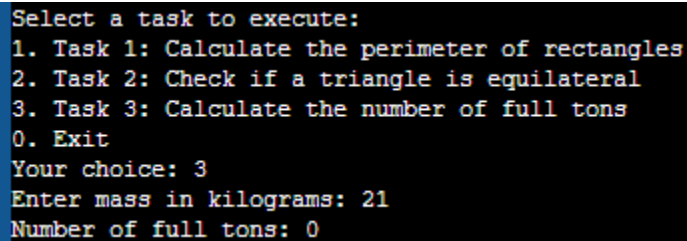
```
// --- Task 3: Calculate the number of full tons ---
int fullTons(int M) {
    // Check the range of input data
    if (M < 0 || M > 100000) { // Assuming mass cannot be negative
        cout << "Error: Mass must be within the range [0, 100000]!\n";
        return -1; // Return an error value
    }
    // Calculate the number of tons
    return M / 1000;
}
```

Рисунок 5- код ( Integer 2 ) .

Лістинг коду :

```
// --- Task 3: Calculate the number of full tons ---
int fullTons(int M) {
    // Check the range of input data
```

```
if (M < 0 || M > 100000) { // Assuming mass cannot be negative
    cout << "Error: Mass must be within the range [0, 100000]!\n";
    return -1; // Return an error value
}
// Calculate the number of tons
return M / 1000;
}
```



```
Select a task to execute:
1. Task 1: Calculate the perimeter of rectangles
2. Task 2: Check if a triangle is equilateral
3. Task 3: Calculate the number of full tons
0. Exit
Your choice: 3
Enter mass in kilograms: 21
Number of full tons: 0
```

Екран роботи програми показаний на рис 6.

## ВИСНОВКИ

У ході виконання завдання було розроблено програму, що реалізує три різні задачі: обчислення периметру прямокутника, перевірка рівносторонності трикутника та визначення кількості повних тонн у заданій масі. Програма забезпечує перевірку коректності вхідних даних і працює в інтерактивному режимі, що дозволяє користувачу вибирати потрібну задачу для розв'язання.