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 № 10**

з дисципліни «Алгоритмізація та програмування»

на тему «Реалізація алгоритмів обробки одновимірних масивів мовою С ++

»

ХАІ.301. 175. 318.14 ЛР

Виконав студент гр. \_\_\_\_\_\_318\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_ Катерина ОВЧАРЕНКО*\_\_\_\_\_\_\_*

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_ к.т.н., доц. Олена  ГАВРИЛЕНКО

(підпис, дата) (П.І.Б.)

2023

# МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення структур (записів)

мовою С ++, а також їх передачі в функції, і реалізувати декларування

обробку структур мовою C ++ в середовищі Visual Studio.

# ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити задачу зі структурами даних. Варіанти завдань

представлено в табл.1.

Завдання 2. Для задач з табл.2-3:

A. Описати структуру, яка містить всі вхідні і всі вихідні дані задачі.

B. Визначити функцію (\*метод), що реалізує обробку структури

відповідно до задачі.

C. Визначити функцію (\*метод), що перевіряє на коректність і заповнює

відповідні поля вхідних даних стуктури

D. Викликати функції (\*методи) з пунктів С, B після оголошення змінної

(об’єкту) структури.

E. Вивести значення полів вихідних даних.

Завдання 3. Рішення всіх трьох задач реалізувати в одному консольному

додатку, \*структурувати на модулі.

# ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Begin 20

Вхідні дані:

x1: Координата x першої точки.

Тип: double.

y1: Координата y першої точки.

Тип: double.

x2: Координата x другої точки.

Тип: double.

y2: Координата y другої точки.

Тип: double.

Вихідні дані:

distance: Відстань між двома точками.

Тип: double.

Алгоритм вирішення:

1.Введення координат першої та другої точок з консолі.

2.Обчислення різниці між координатами по обох вимірах.

3.Обчислення відстані між двома точками за допомогою формули відстані між двома точками у просторі.

4.Виведення отриманої відстані.

Лістинг коду вирішення задачі Begin 20 наведено в додатку А (стор.)

Екран роботи програми показаний на рис. .

Завдання 2.

Вирішення задачі Boolean 27

Вхідні дані:

x : Координата x точки для перевірки.

Тип: double.

y : Координата y точки для перевірки.

Тип: double.

Вихідні дані:

Рядок, який підтверджує, що точка лежить у другій чверті.

Рядок, який підтверджує, що точка НЕ лежить у другій чверті.

Алгоритм вирішення:

1.Введення координат точки з консолі.

2.Перевірка, чи координати точки відповідають другій чверті за допомогою функції isInSecondQuadrant.

3.Виведення результату перевірки на екран.

Завдання 3

Вирішення задачі Param76.

Вхідні данні:

times: Масив структур типу TTime, що містить п'ять моментів часу.

Тип: TTime.

Обмеження: Немає

Вихідні данні: NextSec - змінені значення моментів часу після виклику

функції

Алгоритм вирішення:

1.Визначення структури TTime, що містить три поля: години, хвилини та

секунди.

2.Оголошення функції isValidTime, яка перевіряє правильність часу.

3.Оголошення процедури NextSec, яка додає одну секунду до моменту часу.

4.Оголошення функції printTime, яка виводить момент часу у заданому

форматі.

Визначення масиву times з п'ятьма моментами часу.

5.Застосування процедури NextSec до кожного моменту часу та виведення

результатів до та після додавання однієї секунди до кожного моменту часу.

# ВИСНОВКИ

Під час роботи було вивчено та застосовано основні концепції створення структур даних та функцій для роботи з ними у мові програмування C++. Практично закріплено навички створення функцій для перевірки та маніпулювання даними, що дозволило впевнено розробляти функції для роботи з часом.

ДОДАТОК А

Лістинг коду програми

*#include <iostream>*

*#include <cmath> // Для функції sqrt*

*// Функція для обчислення відстані між двома точками*

*double calculateDistance(double x1, double y1, double x2, double y2) {*

*double deltaX = x2 - x1;*

*double deltaY = y2 - y1;*

*return std::sqrt(deltaX \* deltaX + deltaY \* deltaY);*

*}*

*int main() {*

*// Змінні для зберігання координат*

*double x1, y1, x2, y2;*

*// Введення координат першої точки*

*std::cout << "Введіть координати першої точки (x1 y1): ";*

*std::cin >> x1 >> y1;*

*// Введення координат другої точки*

*std::cout << "Введіть координати другої точки (x2 y2): ";*

*std::cin >> x2 >> y2;*

*// Обчислення відстані між точками*

*double distance = calculateDistance(x1, y1, x2, y2);*

*// Виведення результату*

*std::cout << "Відстань між точками: " << distance << std::endl;*

*return 0;*

*}*

ДОДАТОК Б

*#include <iostream>*

*// Функція для перевірки, чи лежить точка у другій чверті*

*bool isInSecondQuadrant(double x, double y) {*

*return (x < 0 && y > 0);*

*}*

*int main() {*

*double x, y;*

*// Введення координат точки*

*std::cout << "Введіть координати точки (x y): ";*

*std::cin >> x >> y;*

*// Перевірка, чи точка лежить у другій чверті*

*if (isInSecondQuadrant(x, y)) {*

*std::cout << "Точка з координатами (" << x << ", " << y << ") лежить у другій координатній чверті." << std::endl;*

*} else {*

*std::cout << "Точка з координатами (" << x << ", " << y << ") НЕ лежить у другій координатній чверті." << std::endl;*

*}*

*return 0;*

*}*

ДОДАТОК С  
*#include <iostream>*

*struct TTime {*

*int hours; // Години: 0-23*

*int minutes; // Хвилини: 0-59*

*int seconds; // Секунди: 0-59*

*};*

*// Функція для перевірки правильності часу*

*bool isValidTime(const TTime& T) {*

*return (T.hours >= 0 && T.hours < 24) &&*

*(T.minutes >= 0 && T.minutes < 60) &&*

*(T.seconds >= 0 && T.seconds < 60);*

*}*

*// Процедура для додавання однієї секунди до часу*

*void NextSec(TTime& T) {*

*if (!isValidTime(T)) {*

*return; // Час неправильний, не змінюємо його*

*}*

*T.seconds++;*

*if (T.seconds >= 60) {*

*T.seconds = 0;*

*T.minutes++;*

*if (T.minutes >= 60) {*

*T.minutes = 0;*

*T.hours++;*

*if (T.hours >= 24) {*

*T.hours = 0;*

*}*

*}*

*}*

*}*

*// Функція для виведення часу у форматі HH:MM:SS*

*void printTime(const TTime& T) {*

*std::cout << (T.hours < 10 ? "0" : "") << T.hours << ":"*

*<< (T.minutes < 10 ? "0" : "") << T.minutes << ":"*

*<< (T.seconds < 10 ? "0" : "") << T.seconds << std::endl;*

*}*

*int main() {*

*// Визначення п'яти моментів часу*

*TTime times[5] = {*

*{23, 59, 59},*

*{12, 30, 45},*

*{0, 0, 0},*

*{11, 59, 59},*

*{13, 14, 59}*

*};*

*// Застосування процедури NextSec до кожного моменту часу*

*for (int i = 0; i < 5; i++) {*

*std::cout << "Час до NextSec: ";*

*printTime(times[i]);*

*NextSec(times[i]);*

*std::cout << "Час після NextSec: ";*

*printTime(times[i]);*

*std::cout << std::endl;*

*}*

*return 0;*

*}*

Скрін-шоти вікна виконання програми

Рисунок Б.1 – Екран виконання програми для вирішення завдання   
Begin 20

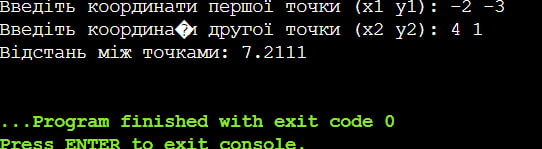
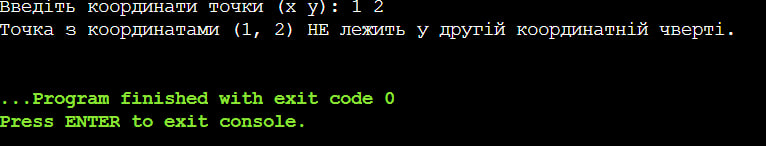


Рисунок Б.2 – Екран виконання програми для вирішення завдання   
Boolean27



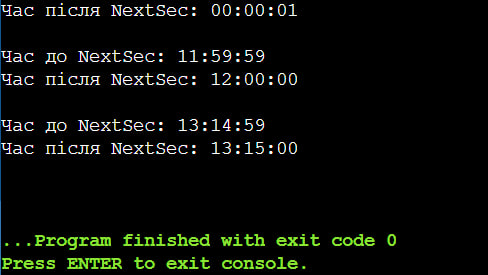


Рисунок Б.3 – Екран виконання програми для вирішення завдання   
Param76