Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Тихоокеанский государственный университет»

Высшая школа кибернетики и цифровых технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

ПО КУРСУ «ПРОГРАММИРОВАНИЕ»

«РАЗРАБОТКА МОДУЛЬНЫХ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ, УКАЗАТЕЛЕЙ И ССЫЛОК»

Выполнил: студент 2 курса группы ПО(б)-31

Корейбо Дмитрий Алексеевич

Проверил: ассистент ВШ КЦТ

Крылов Владимир Андреевич

Хабаровск 2024 г.

Цель работы: изучить теоретический материал по лабораторной работе и реализовать на языке программирования C++ программы для решения заданий.

Задание: необходимо изучить теоретический материал для работы с модульными программами с использование функций, указателей и ссылок, и реализовать несколько программ на языке программирования C++, чтобы получить практический опыт работы с этими концепциями.

Ход работы:

1. работа с программой “Visual Studio” для созданий программ по работе со строками.

limits в C++ — это заголовок, который предоставляет набор шаблонов и констант, содержащих информацию о свойствах числовых типов.

Algorithm — это заголовочный файл в стандартной библиотеке языка программирования C++, включающий набор функций для выполнения алгоритмических операций над контейнерами и над другими последовательностями.

1. составление с помощью данной программы для первого задания (листинг 1).

Листинг 1 — код файла Visual Studio

#include <iostream>

#include <vector>

#include <algorithm>

int main() {

setlocale(LC\_ALL, "RUS");

int n;

std::cout << "Введите количество элементов массива: ";

std::cin >> n;

std::vector<double> arr(n);

std::cout << "Введите элементы массива: ";

for (int i = 0; i < n; ++i) {

std::cin >> arr[i];

}

double sum\_negative = 0;

for (const auto& elem : arr) {

if (elem < 0) {

sum\_negative += elem;

}

}

int first\_zero\_index = -1;

int last\_zero\_index = -1;

for (int i = 0; i < n; ++i) {

if (arr[i] == 0) {

if (first\_zero\_index == -1) {

first\_zero\_index = i;

}

last\_zero\_index = i;

}

}

double sum\_between\_zeros = 0;

if (first\_zero\_index != -1 && last\_zero\_index != -1 && first\_zero\_index != last\_zero\_index) {

for (int i = first\_zero\_index + 1; i < last\_zero\_index; ++i) {

sum\_between\_zeros += arr[i];

}

}

std::sort(arr.begin(), arr.end());

std::cout << "Сумма отрицательных элементов: " << sum\_negative << std::endl;

std::cout << "Сумма элементов между первыми нулями: " << sum\_between\_zeros << std::endl;

std::cout << "Упорядоченный массив: ";

for (const auto& elem : arr) {

std::cout << elem << " ";

}

std::cout << std::endl;

return 0;

}

Результат работы программы первого задания (рисунок 1).

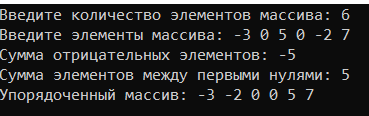


Рисунок 1 – результат работы программы первого задания

1. составление с помощью данной программы для второго задания (листинг 2).

Листинг 2 — код файла Visual Studio

#include <iostream>

#include <vector>

#include <limits>

int main() {

setlocale(LC\_ALL, "RUS");

int n;

std::cout << "Введите размер квадратной матрицы (n x n): ";

std::cin >> n;

std::vector<std::vector<int>> matrix(n, std::vector<int>(n));

std::cout << "Введите элементы матрицы:" << std::endl;

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n; ++j) {

std::cin >> matrix[i][j];

}

}

std::vector<int> column\_sums(n, 0);

std::vector<bool> has\_negative(n, false);

for (int j = 0; j < n; ++j) {

for (int i = 0; i < n; ++i) {

if (matrix[i][j] < 0) {

has\_negative[j] = true;

break;

}

}

if (!has\_negative[j]) {

for (int i = 0; i < n; ++i) {

column\_sums[j] += matrix[i][j];

}

}

}

std::cout << "Суммы элементов в столбцах без отрицательных значений:" << std::endl;

for (int j = 0; j < n; ++j) {

if (!has\_negative[j]) {

std::cout << "Столбец " << j + 1 << ": " << column\_sums[j] << std::endl;

}

}

int min\_sum = std::numeric\_limits<int>::max();

for (int k = -n + 1; k < n; ++k) {

int current\_sum = 0;

for (int i = 0; i < n; ++i) {

int j = n - 1 - i + k;

if (j >= 0 && j < n) {

current\_sum += abs(matrix[i][j]);

}

}

if (current\_sum < min\_sum) {

min\_sum = current\_sum;

}

}

std::cout << "Минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали: " << min\_sum << std::endl;

return 0;

}

Результат работы программы второго задания (рисунок 2).

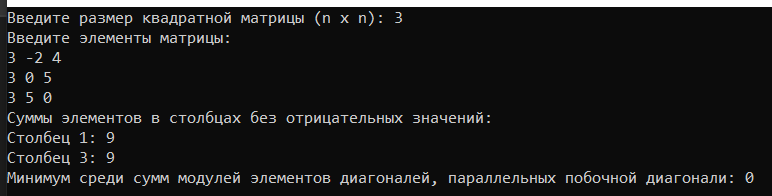


Рисунок 2 – результат работы программы второго задания

Вывод: в ходе выполнения лабораторной работы мы изучили как разрабатывать модульные программы с использованием функций, указателей и ссылок на языке программирования C++. C++ продолжает оставаться популярным благодаря своей универсальности и способности обеспечивать высокую производительность, что делает его идеальным выбором для разработки сложных и высокопроизводительных приложений.