**ЛАБОРАТОРНАЯ РАБОТА No4**

**«ООПC++.ШАБЛОНЫ»**

**Задание** **4:**

Выполнить задание на тему «Одномерные массивы», оформив каждый пункт задания в виде шаблона функции. Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается. Привести примеры программ, использующих эти шаблоны для типов int, float и double.

**Шаблоны функций:**

**Вариант 16**

В одномерном массиве, состоящем из п вещественных элементов, вычислить:

1) количество отрицательных элементов массива;

2) Сумму модулей элементов массива, расположенных после минимального по модулю элемента.

3) Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

**1.Постановка задачи**

В данный отчет будет сделан для второго пункта задания «**Шаблоны функций**».

Разработать шаблонную функцию, выполняющую указанные в задании действии.

Входные данные будут состоять из переменной содержащей тип данных, размера одномерного массива и самого одномерного массива (Табл. 1).

Табл. 1

Входные данные

|  |  |
| --- | --- |
| Входные данные | Тип данных |
| Выбор типа данных | Будет уточнен при разработке программы |
| Размер массива | Будет уточнен при разработке программы |
| Одномерный массив | int, float или double |

Выходные данные будут состоять только из суммы модулей элементов массива, расположенных после минимального по модулю элемента

Табл. 2

Выходные данные

|  |  |
| --- | --- |
| Выходные данные | Тип данных |
| Сумма модулей элементов массива, расположенных после минимального по модулю элемента | int, float или double |

Необходимо предусмотреть граничные и исключительные ситуации и способы их обработки.

**Ограничения:**

1. Нельзя выбрать другой тип данных

2. Нельзя ввести нулевой или отрицательный размер массива

3. Если минимальный элемент является последним или отсутствует (пустой массив).

4. Если массив содержит только нулевые элементы (их больше 1), то в качестве суммы модулей будет возвращен 0

**Обработка ограничений (исключений):**

1. При обработке первого исключения программа закончит свою работу.
2. При обработке второго исключения программа не сумеет просуммировать массив, так как в нем будут отсутствовать элементы. Будет выведено сообщение об отсутствии элементов после минимального элемента (т.к. сам минимальный элемент отсутствует).
3. Будет выведено сообщение об отсутствии элементов после минимального элемента.

**Среда разработки**

Необходимо разработать консольное приложение в среде разработки Visual Studio Community 2019 версии 16.3.0 C++ в соответствии с заданием. Продемонстрировать работу классов и выявить проблемы возникшие при разработки программы.

**2. Разработка программы**

**2.1 Разработка структуры программы**

В соответствии с заданием разработаем программу, нам нужно будет создать файл .cpp в котором мы распишем шаблонную функцию и main.

Пользователю нужно будет выбрать тип данных для одномерного массива, а также заполнить сам массив.

В качестве поля, принимающего тип данных для одномерного массива, предлагаю использовать int (1 - int, 2 - float, 3 - double).

В качестве массива предлагаю использовать класс vectorи переменную типа int, для ввода размера массива (Табл. 3)**.**

Табл. 3

Массив и переменные связанные с ним

|  |  |
| --- | --- |
| Тип данных | Смысл |
| int | Выбор типа данных массива |
| int | Ввести тип данных для одномерного массива |
| vector | Одномерный массив для работы с программой |

**2.2 Разработка алгоритма**

Для выполнения задания, мы сначала должны отыскать минимальное по модулю число, для этого мы должны пройти весь массив от начала до конца, найти минимальный элемент по модулю и запомнить его индекс. Затем, начиная со следующего после минимального элемента, просуммировать все до последнего по модулю элементы и вернуть это значение. Схема алгоритма представлена на Рис. 1.

**2.3 Разработка пользовательского интерфейса**

Пользователем может являться человек, которому необходимо работать с большим объемом данных одного типа.

Программа разрабатывается в консольном виде. В ней будет разработан следующий диалог с пользователем (Рис. 1).

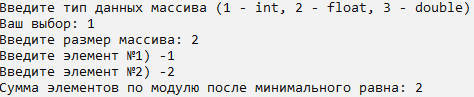


Рис. 2 Пример пользовательского интерфейса

**3. Реализация и тестирование приложения**

***Реализация***

Программа была реализована в Visual Studio Community 2019 версии 16.3.0 в среде разработки C++. На устройстве характеристиками, указанными ниже (Рис. 2). Реализация кода программы будет расписана ниже в пункте «**Листинг кода**».

**Описание компьютера: Д**ля полноты картины приведены более точные характеристики устройства (Рис. 2)

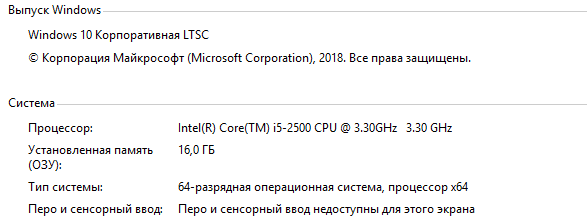




Рис. 3. Характеристики компьютера

***Тестирование***

**Объект тестирования:** Project2.exe

**Цель тестирования:** Проверка общей работоспособности программы.

**Метод тестирование:** функциональное тестирование и проход отладчиком для каждой функции по-отдельности и в общем для всей программы.

Тестирование 1 (Рис 3):

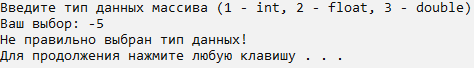


Рис. 4. Тестирование и проверка ограничения №1

Тестирование 2:

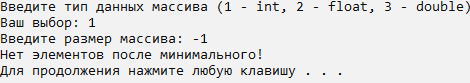


Рис. 5. Тестирование и проверка ограничения №2

Тестирование 3:

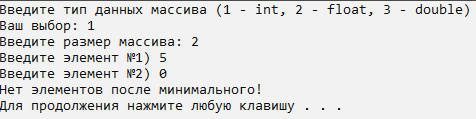


Рис. 6. Тестирование и проверка ограничения №3

Тестирование 4:

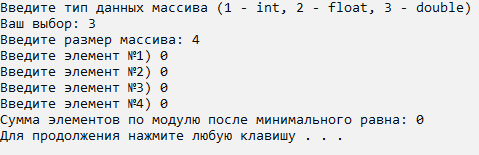


Рис. 7. Тестирование и проверка ограничения №4

Тестирование 5:

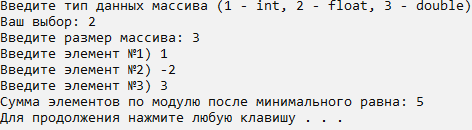


Рис. 7. Тестирование программы в нормальных условиях

**Вывод:** Во время выполнения данной работы я научился создавать и использовать шаблонные функции. Программа работает корректно. Исключения выбрасываются корректно. Можно добавить обработку ввода данных (Вместо числа ввели символ) для полноценной работы программы.

**Листинг кода**

**Подключенные заголовочные файлы:**

#include <iostream>

#include <windows.h>

#include <cmath>

#include <vector>

using namespace std;

**Функция для заполнения вектора:**

//Саутов Давид Группа А-12-18 Вариант №16

template <typename T>

T SumAfterMin(vector<T> massiv) {

int indexMin = 0; //Индекс минимального элемента

for (unsigned int i = 0; i < massiv.size(); i++)

//Цикл для поиска минимального элемента

{

if (fabs(massiv[indexMin] > fabs(massiv[i]))) {

indexMin = i;

}

}

T min = -1;

if ((unsigned int)indexMin + 1 < massiv.size())

//Проверка наличия элемента после минимального элемента

{

++min;

for (unsigned int i = indexMin + 1; i < massiv.size(); i++)

//Цикл для подсчета модуля суммы

{

min += fabs(massiv[i]);

}

}

return min;

}

**Функция для подсчета суммы модулей элементов после минимального элемента массива:**

template <typename T>

void fillVector(vector<T> &a) {

//Заполнения вектора

int size;

cout << "Введите размер массива: ";

cin >> size;

T element;

for (int i = 0; i < size; i++)

{

cout << "Введите элемент №" << i + 1 << ") ";

cin >> element;

a.push\_back(element);

}

}

**main:**

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int tip;//Переменная для определения типа данных

cout << "Введите тип данных массива (1 - int, 2 - float, 3 - double)\nВаш выбор: ";

cin >> tip;//Введения типа данных

vector<int> a1;//Одномерный массив для int

vector<float> a2;//Одномерный массив для float

vector<double> a3;//Одномерный массив для double

int result1;//Переменная для хранения результата функции

float result2;//Переменная для хранения результата функции

double result3;//Переменная для хранения результата функции

switch (tip)

{

case 1:

fillVector<int>(a1);//Заполнение вектора

result1 = SumAfterMin(a1);

if (result1 != -1)//Проверка на наличие элемента/ов после минимального по модулю

{

cout << "Сумма элементов по модулю после минимального равна: " << result1 << endl;

}

else

{

cout << "Нет элементов после минимального!\n";

}

break;

case 2:

fillVector<float>(a2);//Заполнение вектора

result2 = SumAfterMin(a2);

if (result2 != -1)//Проверка на наличие элемента/ов после минимального по модулю

{

cout << "Сумма элементов по модулю после минимального равна: " << result2 << endl;

}

else

{

cout << "Нет элементов после минимального!\n";

}

break;

case 3:

fillVector<double>(a3);//Заполнение вектора

result3 = SumAfterMin(a3);//Проверка на наличие элемента/ов после минимального по модулю

if (result3 != -1)

{

cout << "Сумма элементов по модулю после минимального равна: " << result3 << endl;

}

else

{

cout << "Нет элементов после минимального!\n";

}

break;

default://Для пользователя, который плохо работает с консолью

cout << "Не правильно выбран тип данных!\n";

break;

}

system("pause");

return 0;

}