**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСТИТЕТ**

**им. И.Раззакова**

**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра «**Программное обеспечение компьютерных систем**»

Направление: 710400 «**Программная инженерия**»

Дисциплина: ««**Объектно-ориентированное программирование**»»

ОТЧЕТ

Лабораторная работа №8

Тема: Шаблоны

Выполнил: Кудайбердиев Эрлан

Группа: ПИ-2-21

Проверил: Мусабаев Э. Б.

Бишкек – 2024

**Задание №1**

Создать шаблон функции, которая находит значение максимального элемента массива. Аргументами функции должны быть имя и размер массива (типа int). Создать шаблон функции, осуществляющей сортировку данных массива. Аргументами функции должны быть имя и размер массива (типа int). Продемонстрировать работу шаблонов на данных различных типов.

**Код программы:**

#include <iostream>

using namespace std;

template<typename T>

T findMax(const T arr[], int size) {

T max = arr[0];

for (int i = 1; i < size; ++i) {

if (arr[i] > max) {

max = arr[i];

}

}

return max;

}

template<typename T>

void bubbleSort(T arr[], int size) {

cout << "\n";

cout << "Отсортированный массив:" << endl;

for (int i = 0; i < size - 1; ++i) {

for (int j = 0; j < size - i - 1; ++j) {

if (arr[j] > arr[j + 1]) {

T temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

template<typename T>

void display(const T arr[], int size) {

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

cout << arr[i] << " ";

}

cout << endl;

}

int main() {

setlocale(LC\_ALL, "rus");

const int n = 10;

int Array1[n] = { 3, 1, 4, 1, 5, 9, 2, 6, 11, 2 };

display(Array1, n);

cout << "Максимальный элемент в Массиве Array1: " << findMax(Array1, n) << endl;

bubbleSort(Array1, n);

display(Array1, n);

cout << "\n----------------------------------------------------------------------------------------------" << endl;

float Array2[] = { 3.14, 1.23, 4.56, 1.67, 5.89, 2.22, 6.66, 1,05, 5,98, 3,22 };

display(Array2, n);

cout << "Максимальный элемент в Массиве Array2: " << findMax(Array2, n) << endl;

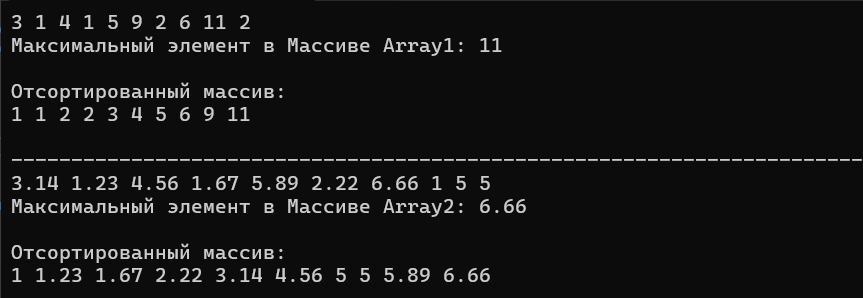
bubbleSort(Array2, n);

display(Array2, n);

return 0;

}

**Результат:**



**Задание №2**

Создать шаблон класса, у которого есть 2 поля в виде одномерных массивов одного размера. Заполнить массивы с помощью генератора случайных чисел. Найти сумму элементов этих массивов. Продемонстрировать работу шаблонов на данных различных типов.

**Код программы:**

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

template<typename T, int Size>

class ArraySum {

private:

T array1[Size];

T array2[Size];

public:

ArraySum() {

srand(time(nullptr));

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

array1[i] = static\_cast<T>(rand() % 100);

array2[i] = static\_cast<T>(rand() % 100);

}

}

T sum() {

T totalSum = 0;

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

totalSum += array1[i] + array2[i];

}

return totalSum;

}

};

int main() {

setlocale(LC\_ALL, "rus");

ArraySum<int, 5> intArraySum;

cout << "Сумма int: " << intArraySum.sum() << endl;

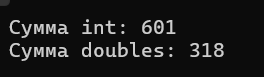
ArraySum<double, 3> doubleArraySum;

cout << "Сумма doubles: " << doubleArraySum.sum() << endl;

return 0;

}

**Результат:**

****

**Задание №3**

Создать шаблон класса «стек». Использовать его при решении задачи 3 из лабораторной № 1.

**Код программы:**

#include <iostream>

#include <stack>

#include <string>

using namespace std;

struct Employee {

string Name;

int age;

string specialty;

double averageSalary;

};

template <typename T>

class Stack {

private:

stack<T> data;

public:

void push(const T& item) {

data.push(item);

}

template <typename U>

U maxElement(U arr[], int size) {

U max = arr[0];

for (int i = 1; i < size; ++i) {

if (arr[i] > max) {

max = arr[i];

}

}

return max;

}

T pop() {

T top = data.top();

data.pop();

return top;

}

bool isEmpty() const {

return data.empty();

}

};

int main() {

setlocale(LC\_ALL, "rus");

Stack<Employee> workersStack;

workersStack.push({ "Смит", 35, "токарь", 2500 });

workersStack.push({ "Джонсон", 28, "токарь", 2200 });

workersStack.push({ "Уильямс", 40, "слесарь", 2800 });

workersStack.push({ "Браун", 32, "слесарь", 2600 });

workersStack.push({ "Эрлан", 35, "токарь", 2500 });

/\* double doubleArr[] = { 3.5, 1.2, 6.7, 2.9, 8.1 };

cout << "Максимальный элемент массива doubleArr: " << workersStack.maxElement(doubleArr, 5) << endl;\*/

int locksmithsCount = 0;

int turnersCount = 0;

Stack<Employee> tempStack;

while (!workersStack.isEmpty()) {

Employee worker = workersStack.pop();

tempStack.push(worker);

if (worker.specialty == "слесарь") {

locksmithsCount++;

}

else if (worker.specialty == "токарь") {

turnersCount++;

}

}

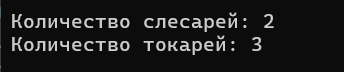
cout << "Количество слесарей: " << locksmithsCount << endl;

cout << "Количество токарей: " << turnersCount << endl;

return 0;

}

**Результат:**

****