Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление: 710400 «Программная инженерия»

Дисциплина: « Объектно-ориентрованное проектирование / Объектно-ориентрованное программирование»

**ОТЧЕТ**

По лабораторной работе №8

Тема: «Шаблоны»

Выполнила: студентка группы

ПИ(б)-3-21 Пак Ксения

Проверил: Мусабаев Э. Б.

Бишкек – 2024

Задания

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.

#include <iostream>

template<typename T, typename M>

T findMax(const T\* arr, M m,int size) {

std::cout << std::endl<<m;

T max = arr[0];

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

if (arr[i] > max) {

max = arr[i];

}

}

return max;

}

template<typename T>

void sortArray(T\* arr, int size) {

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;

}

}

}

}

int main() {

int intArr[] = { 0, 7, 1, -9, 5 };

int intSize = sizeof(intArr) / sizeof(int);

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

std::cout << intArr[i] << " ";

}

std::cout << std::endl << "Max of intArr: " << findMax(intArr, 2, intSize) << std::endl;

sortArray(intArr, intSize);

std::cout << "Sorted intArr: ";

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

std::cout << intArr[i] << " ";

}

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

double doubleArr[] = { 3.5, 7.2, 1.8, 9.1, 5.4 };

int doubleSize = sizeof(doubleArr) / sizeof(double);

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

std::cout << doubleArr[i] << " ";

}

std::cout << std::endl << "Max of doubleArr: " << findMax(doubleArr, "2",doubleSize) << std::endl;

sortArray(doubleArr, doubleSize);

std::cout << "Sorted doubleArr: ";

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

std::cout << doubleArr[i] << " ";

}

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

char charArr[] = { 'z', 'a', 'b', 'c', 'x' };

int charSize = sizeof(charArr) / sizeof(char);

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

std::cout << charArr[i] << " ";

}

std::cout << std::endl << "Max of charArr: " << findMax(charArr, 1.2, charSize) << std::endl;

sortArray(charArr, charSize);

std::cout << "Sorted charArr: ";

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

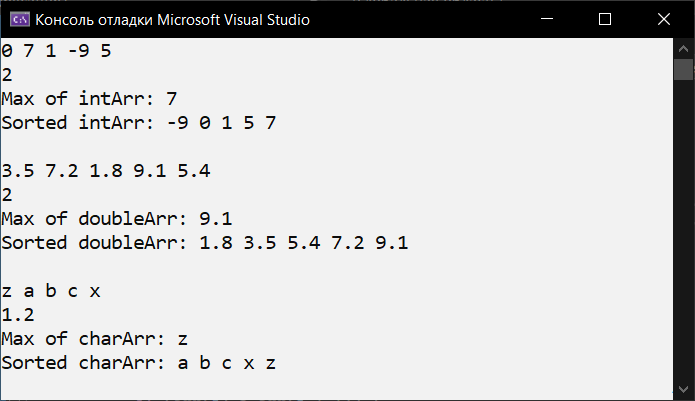
std::cout << charArr[i] << " ";

}

std::cout << std::endl;

return 0;

}



1. 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.

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

template<typename T, size\_t N>

class Array {

private:

T array1[N];

T array2[N];

public:

Array() {

//srand(time(0));

for (size\_t i = 0; i < N; ++i) {

array1[i] = static\_cast<T>(1 + rand() % 100);

array2[i] = static\_cast<T>(1 + rand() % 100);

}

}

T sum() const {

T sum = 0;

for (size\_t i = 0; i < N; ++i) {

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

}

return sum;

}

void printArrays() const {

std::cout << "Array 1: ";

for (size\_t i = 0; i < N; ++i) {

std::cout << array1[i] << " ";

}

std::cout << std::endl;

std::cout << "Array 2: ";

for (size\_t i = 0; i < N; ++i) {

std::cout << array2[i] << " ";

}

std::cout << std::endl;

}

};

int main() {

Array<int, 5> intArray;

intArray.printArrays();

std::cout << "Sum of integers: " << intArray.sum() << std::endl;

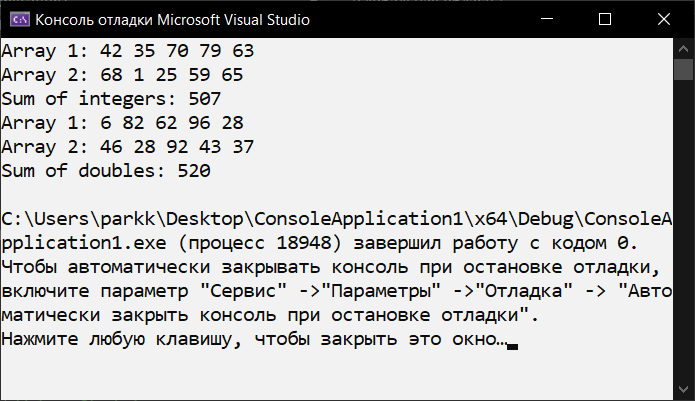
Array<double, 5> doubleArray;

doubleArray.printArrays();

std::cout << "Sum of doubles: " << doubleArray.sum() << std::endl;

return 0;

}



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

#include <iostream>

#include <string>

#include <vector>

using namespace std;

template<typename t>

class stack {

private:

std::vector<t> elements;

public:

void push(const t& element) {

elements.push\_back(element);

}

t pop() {

if (elements.empty()) {

throw std::runtime\_error("stack is empty");

}

t topelement = elements.back();

elements.pop\_back();

return topelement;

}

size\_t size() const {

return elements.size();

}

};

struct worker {

std::string surname;

int age;

std::string position;

double salary;

};

int main() {

stack<worker> workersstack;

int factorycount;

std::cout << "enter the number of factories: ";

std::cin >> factorycount;

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

int workercount;

std::cout << "enter the number of workers in factory " << i + 1 << ": ";

std::cin >> workercount;

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

worker worker;

std::cout << "enter worker's surname, age, specialization, and salary: ";

std::cin >> worker.surname >> worker.age >> worker.position >> worker.salary;

workersstack.push(worker);

}

}

int locksmithcount = 0;

int turnercount = 0;

stack<worker> tempstack;

while (workersstack.size() > 0) {

worker worker = workersstack.pop();

tempstack.push(worker);

if (worker.position == "locksmith") {

++locksmithcount;

}

else if (worker.position == "turner") {

++turnercount;

}

}

while (tempstack.size() > 0) {

worker worker = tempstack.pop();

workersstack.push(worker);

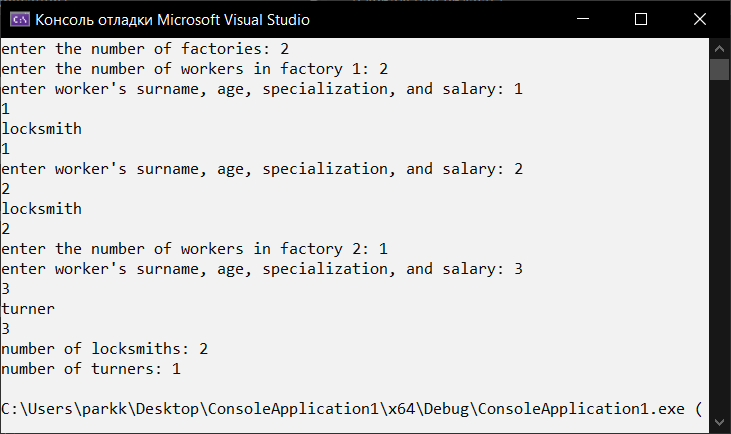
}

std::cout << "number of locksmiths: " << locksmithcount << std::endl;

std::cout << "number of turners: " << turnercount << std::endl;

return 0;

}



Ответы на вопросы

1. Что такое обобщенная функция?

**Шаблон функции (обобщенные функции) –** это функции, в которых тип данных передается, как формальный параметр, называется шаблоном функции.

2. Шаблоны функции позволяют удобным способом создать семейство

а) переменных; б) функций; в) классов; г) программ.

Ответ: б.

1. Истинно ли утверждение, что шаблонная функция может иметь несколько аргументов?

Истинно.

4. Реальный код шаблонной функции генерируется при:

а) объявлении функции в исходном коде;

б) создании объекта;

в) запуске функции из её исходного кода.

Ответ: а

1. Что такое обобщенный (шаблонный) класс?

**Шаблон класса (обобщенный класс) – это** класс в которых тип данных передается как формальный параметр.

1. В каком случае его имеет смысл применять?

При работе с разными типами данных, при написании алгоритмов, не зависящих от конкретного типа данных, при необходимости повышения гибкости кода.

1. Шаблонный класс:

а) работает с разными типами данных;

б) генерирует идентичные объекты;

в) генерирует классы с различным числом методов

Ответ: а.