**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО»**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 22**

**«Работа с ООП»**

Специальность 09.02.07«Информационные системы и программирование»

Дисциплина «Основы алгоритмов и программирование»

|  |  |
| --- | --- |
| Преподаватель:  Яковлева Н.М.  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_202\_\_г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2231  Петров Н.В. |

Санкт-Петербург

2019/2020

**Цель работы:**

1. Написать программу, которая сортирует по убыванию элементы одномерного массива, который может состоять из различных числовых типов данных. Используйте шаблон функции. Все необходимые данные должны передаваться функции в качестве параметров. Продемонстрировать работу шаблона.
2. Создать шаблон класса “стек” (см. лекцию о динамических структурах данных). У класса должно быть хотя бы два поля различных параметризованных типов.

**Код:**

1. main.cpp

#include <iostream>  
using namespace std;  
  
template<class Type>  
void sort(Type \*data, int size) {  
 bool sorted = false;  
 while (!sorted) {  
 sorted = true;  
 for (int i = 0; i < (size - 1); i++) {  
 if (data[i] > data[i+1]) {  
 sorted = false;  
 Type tmp = data[i];  
 data[i] = data[i+1];  
 data[i+1] = tmp;  
 }  
 }  
 }  
}  
  
template<class Type>  
void print(Type \*data, int size) {  
 cout << data[0];  
 for (int i = 1; i < size; i++)  
 cout << " " << data[i];  
 cout << endl;  
}  
  
int main() {  
 int array[] = {46, 34, 81, 61, 31, 42, 16, 32, 91, 87};  
 float array2[] = {6.5, 9.6, 9.1, 8.2, 5.0, 3.7, 9.4, 4.2, 1.0, 1.5};  
  
 print(array, 10);  
 sort(array, 10);  
 print(array, 10);  
  
 print(array2, 10);  
 sort(array2, 10);  
 print(array2, 10);  
  
 return 0;  
}

1. main.cpp

#include <iostream>  
  
using namespace std;  
  
template<class Type, class Size>  
class Stack {  
 Size size = 0;  
 struct Node {  
 Type data;  
 Node \*p;  
 };  
 Node \*top = nullptr;  
public:  
 void push(const Type data) {  
 Node \*tmp\_p = new Node;  
 (\*tmp\_p).data = data;  
 (\*tmp\_p).p = (\*this).top;  
 (\*this).top = tmp\_p;  
 size++;  
 }  
  
 Type pop() {  
 Type tmp = (\*(\*this).top).data;  
 Node \*tmp\_p = (\*this).top;  
 (\*this).top = (\*this).top->p;  
 delete tmp\_p;  
 size--;  
 return tmp;  
 }  
  
 inline Size get\_size() {  
 return size;  
 }  
};  
  
int main() {  
 Stack<float, unsigned int> stack;  
 float array[] = {76, 59, 28, 52, 91};  
  
 for (auto i : array)  
 stack.push(i);  
 cout << "size: " << stack.get\_size() << endl;  
 for (auto \_ : array)  
 cout << stack.pop() << endl;  
 cout << "size: " << stack.get\_size() << endl;  
  
 return 0;  
}

**Результаты и вывод:**

Вывод программы

1.

46 34 81 61 31 42 16 32 91 87  
16 31 32 34 42 46 61 81 87 91  
6.5 9.6 9.1 8.2 5 3.7 9.4 4.2 1 1.5  
1 1.5 3.7 4.2 5 6.5 8.2 9.1 9.4 9.6

2.

size: 5  
91  
52  
28  
59  
76  
size: 0