Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)

**Лабораторная работа № 4  
Использование своих классов в STL (часть 1)**

Работу выполнил:

Конов Михаил, ИУ8-24

**ЦЕЛЬ**

Изучение использования пользовательский классов при работе с контейнерами STL.

**УСЛОВИЕ ЗАДАЧИ**

В приложении организовать контейнер объектов своего класса (использовать шаблоны std::list, std::vector или std::deque в зависимости от варианта, элементы контейнера - объекты класса, **не указатели!!!!**). Варианты заданий заданы в ячейках таблицы 1. Параметры приложений указаны в заголовках строк и столбцов таблицы 1.

Класс должен иметь необходимые конструкторы, конструктор копирования и перемещения при необходимости (обосновать отсутствие или наличие необходимости), перегруженные операции присваивания с копированием и перемещением при необходимости (обосновать отсутствие или наличие необходимости), перегруженную операцию вставки в поток <<.

Обеспечить копирование одного контейнера в другой с помощью алгоритма std::copy. А также сортировку объектов в исходном контейнере, для шаблона list при сортировке использовать метод list::sort без параметров, для шаблона vector или deque при сортировке использовать алгоритм std::sort с двумя параметрами: итератор на начало и итератор на конец контейнера.

Исходные данные прочитать из текстового файла input.txt. Вывести в выходной файл output.txt исходный контейнер, контейнер после сортировки, использовать при этом перегруженную операцию вставки в поток, также вывести в выходной файл контейнер, в который скопирован исходный контейнер.

**ВАРИАНТ 10**

|  |  |  |
| --- | --- | --- |
| **Параметры приложений** | | **Исходный контейнер list, копируем в vector** |
| Объект- студент (поля: ФИО, группа, номер зачетной книжки, массив 4-х оценок за сессию) | Сортировка по ФИО | 10 |

**ТЕКСТ ПРОГРАММЫ**

#include <iostream>

#include <vector>

#include <list>

#include <string>

#include <fstream>

using namespace std;

//Объект - студент (поля: ФИО, группа, номер зачетной книжки, массив 4-х оценок за сессию)

//Сортировка по ФИО

//Исходный контейнер - list, копируем в vector

class Student{

public:

string name;

string group;

string zachetka;

int session[4];

Student(){}

Student(string& name, string& group, string& zachetka, int session[4]){

this->name = name;

this->group = group;

this->zachetka = zachetka;

for (int i = 0; i < 4; ++i){

this->session[i] = session[i];

}

}

Student(const Student& S){

this->name = S.name;

this->group = S.group;

this->zachetka = S.zachetka;

for (int i = 0; i < 4; ++i){

this->session[i] = S.session[i];

}

}

print(){

cout << "\nФИО: " << name << "\nГруппа: " << group << "\nЗачетка: " << zachetka << endl;

for (auto pos: session){

cout << session[pos];

}

}

friend ostream& operator << (ostream& osm, Student& St){

osm << "\nФИО: " << St.name << "\nГруппа: " << St.group << "\nЗачетка: " << St.zachetka << endl;

for (int i = 0; i < 4; ++i){

osm << St.session[i];

}

osm << endl;

return osm;

}

bool operator <(const Student& s1){

if (name < s1.name){

return 1;

}

else{

return 0;

}

}

};

int main(){

ifstream infile("input.txt");

fstream outfile("output.txt");

setlocale(LC\_ALL, "RUS");

cout << "Hello world!" << endl;

//создаем list

list <Student> origin;

string name, group, zachetka;

int session[4];

while(!infile.eof()){

getline(infile, name);

// cout << name << "\t";

infile >> group;

infile.ignore();

// cout << group << "\t";

infile >> zachetka;

infile.ignore();

// cout << zachetka << "\t";

for (int i = 0; i < 4; ++i){

infile >> session[i];

cout << session[i];

}

cout << "\t";

infile.ignore();

origin.push\_back(Student(name, group, zachetka, session));

}

//сортируем с помощью sort без параметров

cout << "Исходный список до сортировки:\n";

outfile << "Исходный список до сортировки:\n";

for (auto iter = origin.begin(); iter != origin.end(); iter++)

{

outfile << \*iter << "\t";

cout << \*iter << "\t";

}

vector <Student> legacy;

legacy.resize(origin.size());

copy(origin.begin(), origin.end(), legacy.begin());

cout << "Результат копирования в другой контейнер:\n";

outfile << "Результат копирования в другой контейнер:\n";

for (auto iter = legacy.begin(); iter != legacy.end(); iter++)

{

outfile << \*iter << "\t";

cout << \*iter << "\t";

}

cout << "Исходный список после сортировки:\n";

outfile << "Исходный список после сортировки:\n";

origin.sort();

for (auto iter = origin.begin(); iter != origin.end(); iter++)

{

outfile << \*iter << "\t";

cout << \*iter << "\t";

}

return 0;

}

**ВЫВОД**

Использование пользовательских классов в контейнерах STL открывает много возможностей для решения задач с разными данными при программировании на языке C++.