**8 ЛАБОРАТОРНАЯ РАБОТА №8**

**«ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ РАБОТЫ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ КОНТЕЙНЕРОВ. СТАНДАРТНАЯ БИБЛИОТЕКА ШАБЛОНОВ»**

**8.1 Цель работы**

Изучить способы реализации и особенности работы программ с использованием контейнеров стандартной библиотеки шаблонов.

**8.2 Вариант задания – 8**

Разработать программу «Работа деканата». В базе хранится информация о студентах — ФИО, оценки, группа, курс. Программа должна обеспечивать выбор с помощью меню.

Требуется реализовать выполнение следующих функций:

– загрузка «базы данных» (из файла) или ее создание;

– общий вывод информации о студентах;

– вывод информации о студентах запрошенной группы, курса;

– вычисление и вывод информации о среднем балле групп.

Хранение данных организовать с применением контейнерного класса deque.

**8.3 Ход работы**

8.3.1 В main.cpp осуществлена работа с базой данных с помощью меню. В заголовочном файле strustures.h осуществляется подключение библиотек и описание структур данных. В файлe functions.cpp описаны функции для работы с базой данных. В файле saveAndDownload.cpp описаны функции сохранения и загрузки базы данных в программу.

8.3.2 Написана программа на С++ согласно вышеописанного алгоритма.

Файл **main.cpp**

#include "headers/structures.h"

#include "headers/functions.h"

#include "headers/saveAndDownload.h"

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

system("color B0");

std::deque<Element> database;

char ch;

while (1) {

system("cls");

std::cout << "1 - Создание базы данных (добавление элементов в конец очереди)" << std::endl;

std::cout << "2 - Загрузка базы данных из файла" << std::endl;

std::cout << "3 - Сохранить базу данных в файл" << std::endl;

std::cout << "4 - Вывод базы данных" << std::endl;

std::cout << "5 - Вывод информации о студентах запрошенной группы, курса" << std::endl;

std::cout << "6 - Вычисление и вывод информации о среднем балле групп" << std::endl;

std::cout << "7 - Очистить базу данных" << std::endl;

std::cout << "8 - Выход" << std::endl;

std::cout << " -> "; std::cin >> ch;

system("cls");

switch (ch) {

case '1' : {

do {

database.push\_back( setData() );

} while ( continueOrNot() );

break;

}

case '2' : {

database.clear();

database = downloadDataBase(database);

break;

}

case '3' : {

saveDataBase(database);

break;

}

case '4' : {

showDataBase(database);

break;

}

case '5' : {

showSelectedGroup(database);

break;

}

case '6' : {

averageScoreByGroup(database);

break;

}

case '7' : {

database.clear();

break;

}

case '8' : {

database.clear();

return 0;

}

}

}

}

Файл **structures.h**

#ifndef TEST\_STRUCTURES\_H

#define TEST\_STRUCTURES\_H

#include <iostream>

#include <windows.h>

#include <deque>

#include <iomanip>

#include <fstream>

#include <algorithm>

//==============================================================================

// СТРУКТУРЫ

//==============================================================================

struct FIO {

std::string name;

std::string surname;

std::string patronymic;

};

struct Marks {

int OOP;

int AIP;

int OSIS;

int math;

};

struct Element {

FIO fio;

Marks marks;

std::string group;

int course;

};

#endif //TEST\_STRUCTURES\_H

Файл **functions.cpp**

#include "../headers/functions.h"

// Продолжить или нет?

bool continueOrNot() {

char ch;

while(1) {

system("cls");

std::cout << "Продолжаем действие? y/n или д/н" << std::endl;

std::cout << " -> "; std::cin >> ch;

if ((ch == 'y') || (ch == 'д')) {

return true;

}

else if ((ch == 'n') || (ch == 'н')) {

return false;

}

std::cout << "\a";

}

}

// Ввод информации об одном студенте

Element setData() {

Element temp;

system("cls");

std::cout << "Введите имя, фамилию и отчество студента:" << std::endl;

std::cout << " -> "; std::cin >> temp.fio.name;

std::cout << " -> "; std::cin >> temp.fio.surname;

std::cout << " -> "; std::cin >> temp.fio.patronymic;

std::cout << std::endl;

std::cout << "Введите оценки по ООП, АиП, ОСИСУ и Выш.мату:" << std::endl;

std::cout << " -> "; std::cin >> temp.marks.OOP;

std::cout << " -> "; std::cin >> temp.marks.AIP;

std::cout << " -> "; std::cin >> temp.marks.OSIS;

std::cout << " -> "; std::cin >> temp.marks.math;

std::cout << std::endl;

std::cout << "Введите название группы:" << std::endl;

std::cout << " -> "; std::cin >> temp.group;

std::cout << std::endl;

std::cout << "Введите номер курса:" << std::endl;

std::cout << " -> "; std::cin >> temp.course;

return temp;

}

// Вывод шапки таблицы

void headerDataBase() {

std::cout << "База данных:" << std::endl << std::endl;

std::cout << "\t" << std::left;

std::cout << std::setw(10) << "ИМЯ" << " | ";

std::cout << std::setw(20) << "ФАМИЛИЯ" << " | ";

std::cout << std::setw(15) << "ОТЧЕСТВО" << " | ";

std::cout << std::setw(4) << "OOP" << " | ";

std::cout << std::setw(4) << "AIP" << " | ";

std::cout << std::setw(4) << "OSIS" << " | ";

std::cout << std::setw(4) << "MATH" << " | ";

std::cout << std::setw(15) << "ГРУППА" << " | ";

std::cout << std::setw(4) << "КУРС" << " | " << std::endl;

std::cout << std::setw(107) << std::setfill('x') << "\tx" << std::endl;

std::cout << std::setfill(' ');

}

// Показать информацию о студенте

void showStudent(std::deque<Element>::iterator i) {

std::cout << "\t" << std::left;

std::cout << std::setw(10) << (\*i).fio.name << " | ";

std::cout << std::setw(20) << (\*i).fio.surname << " | ";

std::cout << std::setw(15) << (\*i).fio.patronymic << " | ";

std::cout << std::setw(4) << (\*i).marks.OOP << " | ";

std::cout << std::setw(4) << (\*i).marks.AIP << " | ";

std::cout << std::setw(4) << (\*i).marks.OSIS << " | ";

std::cout << std::setw(4) << (\*i).marks.math << " | ";

std::cout << std::setw(15) << (\*i).group << " | ";

std::cout << std::setw(4) << (\*i).course << " | " << std::endl;

}

// Показать базу данных

void showDataBase(std::deque<Element> database) {

if (database.empty()) {

std::cout << "База данных пуста\a" << std::endl;

}

else {

headerDataBase();

for (std::deque<Element>::iterator i = database.begin(); i != database.end(); i++) {

showStudent(i);

}

}

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

system("pause");

}

// Показать студентов в выбранной группе

void showSelectedGroup(std::deque<Element> database) {

if (database.empty()) {

std::cout << "База данных пуста\a" << std::endl;

}

else {

std::string group;

int course;

int counter = 0;

std::cout << "Введите группу:" << std::endl;

std::cout << " -> "; std::cin >> group;

std::cout << std::endl;

std::cout << "Введите номер курса:" << std::endl;

std::cout << " -> "; std::cin >> course;

std::cout << std::endl;

headerDataBase();

for (std::deque<Element>::iterator i = database.begin(); i != database.end(); i++) {

if (((\*i).group == group) && ((\*i).course == course)) {

counter++;

showStudent(i);

}

}

if (counter == 0) {

std::cout << std::endl << "Чё то мы не нашли студентов в указанной группе, с указанным курсом\a" << std::endl;

}

}

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

system("pause");

}

// Средний балл по группам

void averageScoreByGroup(std::deque<Element> database) {

if (database.empty()) {

std::cout << "База данных пуста\a" << std::endl << std::endl << std::endl;

system("pause");

return;

}

std::sort(database.begin(), database.end(), [](const Element& el1, const Element& el2) -> bool {

return el1.group < el2.group;

}

);

// Можно было перегрузить "<"

// bool operator < (const Element& el1, const Element& el2) {

// return el1.group < el2.group;

// }

// и сделать так: std::sort(database.begin(), database.end());

std::deque<Element>::iterator i = database.begin();

std::string currentGroup = (\*i).group;

float currentOOP = 0;

float currentAIP = 0;

float currentOSIS = 0;

float currentMATH = 0;

int counter = 0;

while (i != database.end() + 1) {

if ((\*i).group == currentGroup) {

counter++;

currentOOP += (\*i).marks.OOP;

currentAIP += (\*i).marks.AIP;

currentOSIS += (\*i).marks.OSIS;

currentMATH += (\*i).marks.math;

}

else {

std::cout << "\tВ группе: " << currentGroup << " средний балл" << std::endl;

std::cout << "\tпо ООП = " << (currentOOP / counter) << std::endl;

std::cout << "\tпо АиП = " << (currentAIP / counter) << std::endl;

std::cout << "\tпо ОСИС = " << (currentOSIS / counter) << std::endl;

std::cout << "\tпо выш.мату = " << (currentMATH / counter) << std::endl;

std::cout << "\tсредний балл по всем предметам = " << (((currentOOP + currentAIP + currentOSIS + currentMATH)) / (counter\*4)) << std::endl << std::endl;

currentGroup = (\*i).group;

currentOOP = (\*i).marks.OOP;

currentAIP = (\*i).marks.AIP;

currentOSIS = (\*i).marks.OSIS;

currentMATH = (\*i).marks.math;

counter = 1;

}

i++;

}

system("pause");

}

Файл **functions.h**

#ifndef TEST\_FUNCTIONS\_H

#define TEST\_FUNCTIONS\_H

#include "structures.h"

//==============================================================================

// ФУНКЦИИ

// ПРОТОТИПЫ

//==============================================================================

bool continueOrNot(); // Продолжить или нет?

Element setData(); // Ввод информации об одном студенте

void headerDataBase(); // Вывод шапки таблицы

void showStudent(std::deque<Element>::iterator i); // Показать информацию о студенте

void showDataBase(std::deque<Element> database); // Показать базу данных

void showSelectedGroup(std::deque<Element> database); // Показать студентов в выбранной группе

void averageScoreByGroup(std::deque<Element> database); // Средний балл по группам

#endif //TEST\_FUNCTIONS\_H

Файл **saveAndDownload.cpp**

#include "../headers/saveAndDownload.h"

// Сохранить базу данных

void saveDataBase(std::deque<Element> database) {

std::string filename;

std::cout << "Введите имя текстового файла:" << std::endl;

std::cout << " -> "; std::cin >> filename;

std::cout << std::endl;

// Открыть файл для записи

std::ofstream fout(filename);

if (!fout.is\_open()) {

std::cout << "ERROR: ошибка открытия файла: \a" << filename << std::endl << std::endl;

system("pause");

return;

}

for(std::deque<Element>::iterator i = database.begin(); i != database.end(); i++) {

fout << (\*i).fio.name << std::endl;

fout << (\*i).fio.surname << std::endl;

fout << (\*i).fio.patronymic << std::endl;

fout << (\*i).marks.OOP << std::endl;

fout << (\*i).marks.AIP << std::endl;

fout << (\*i).marks.OSIS << std::endl;

fout << (\*i).marks.math << std::endl;

fout << (\*i).group << std::endl;

fout << (\*i).course << std::endl;

}

fout.close();

std::cout << "База данных успешно сохранена в файл " << filename << std::endl << std::endl;

system("pause");

}

// Загрузить базу данных

std::deque<Element> downloadDataBase(std::deque<Element> database) {

std::string filename;

std::cout << "Введите имя текстового файла:" << std::endl;

std::cout << " -> "; std::cin >> filename;

std::cout << std::endl;

// Открыть файл для чтения

std::ifstream fin(filename);

if (!fin.is\_open()) {

std::cout << "ERROR: ошибка открытия файла: \a" << filename << std::endl << std::endl;

system("pause");

return database;

}

Element temp;

while (1) {

fin >> temp.fio.name;

fin >> temp.fio.surname;

fin >> temp.fio.patronymic;

fin >> temp.marks.OOP;

fin >> temp.marks.AIP;

fin >> temp.marks.OSIS;

fin >> temp.marks.math;

fin >> temp.group;

fin >> temp.course;

if (fin.eof()) break;

database.push\_back( temp );

}

fin.close();

std::cout << "База данных успешно загружена из файла " << filename << std::endl << std::endl;

system("pause");

return database;

}

Файл **saveAndDownload.h**

#ifndef TEST\_SAVEANDDOWNLOAD\_H

#define TEST\_SAVEANDDOWNLOAD\_H

#include "structures.h"

#include "functions.h"

//==============================================================================

// ФУНКЦИИ СОХРАНЕНИЯ И ЗАГРУЗКИ ТАБЛИЦЫ

// ПРОТОТИПЫ

//==============================================================================

void saveDataBase(std::deque<Element> database); // Сохранить базу данных

std::deque<Element> downloadDataBase(std::deque<Element> database); // Загрузить базу данных

#endif //TEST\_SAVEANDDOWNLOAD\_H

8.3.3 Выполнена отладка программы.

Результаты тестирования отображены на рисунках 8.1– 8.6. На рисунке 8.1 представлена реализация меню. На рисунке 8.2 осуществляется ввод данных об одном студенте и последующее добавление этого студента в базу данных. На рисунке 8.3 представлена загрузка базы данных из текстового файла. На рисунке 8.4 отображено выполнение функции демонстрации базы данных в виде таблицы. На рисунке 8.5 осуществляется ввод названия группы и номера курса, согласно которым студенты из этой группы выводятся на экран, если они есть. На рисунке 8.6 представлен вывод информации о средних баллах по группам.

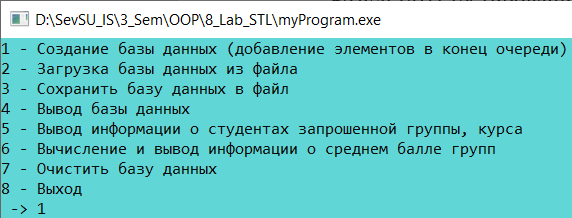


Рисунок 8.1 – Меню

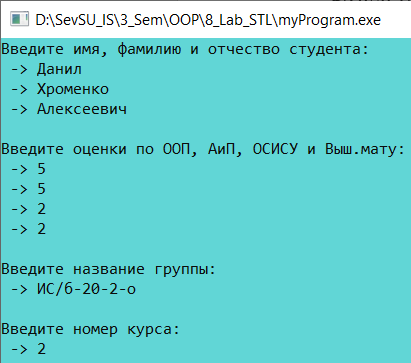


Рисунок 8.2 – Ввод данных о студенте

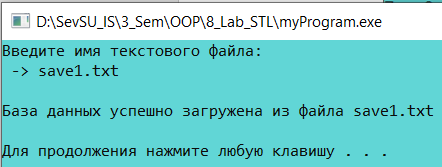


Рисунок 8.3 – Загрузка базы данных из текстового файла

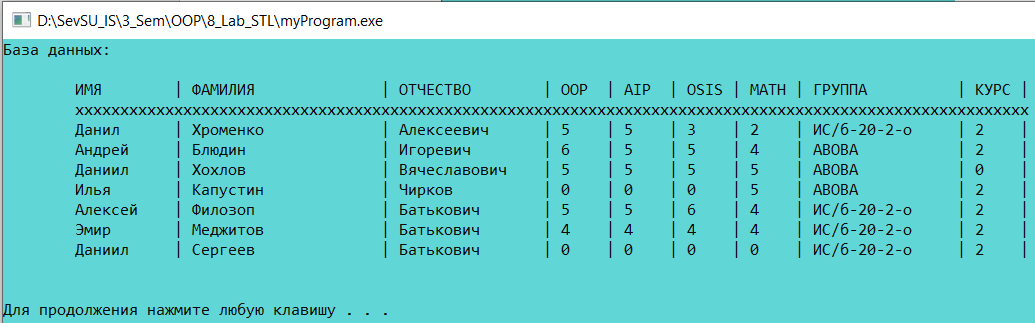


Рисунок 8.4 – Демонстрация базы данных

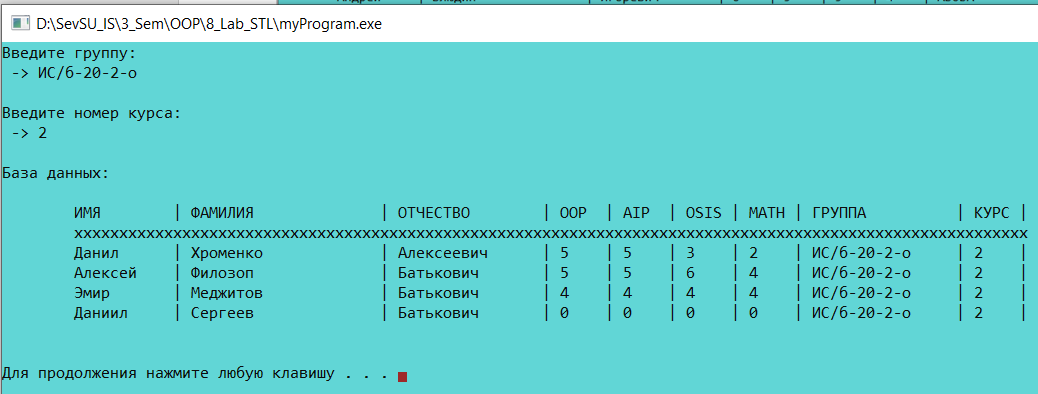


Рисунок 8.5 – Вывод студентов из конкретной группы

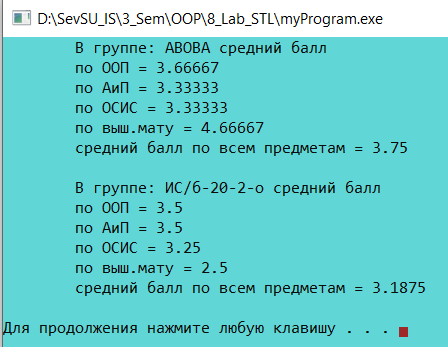


Рисунок 8.6 – Вывод информации о средних баллах по группам

Результаты тестирования полностью соответствуют ожиданиям.

**Выводы**

В ходе выполнения данной лабораторной работы были получены навыки разработки программ, использующих контейнеры STL (стандартной библиотеки шаблонов). Были закреплены навыки работы многофайловым проектом, навыки работы с файлами. Полученные во время разработки навыки помогут разрабатывать более сложные программы с использованием классов и объектов, более эффективные по времени выполнения алгоритмы.