**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №1**

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 2373 |  | Чесноков М. А. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2023

**Цель работы.**

Изучение структур в c++, массивов структур, создание прототипа баз данный в c++ при помощи структур, добавление функционала.

**Основные теоретические положения.**

В языке c++, структура — композитный тип данных, инкапсулирующий без сокрытия набор значений различных типов. Порядок размещения значений в памяти задаётся при определении типа и сохраняется на протяжении времени жизни объектов, что даёт возможность косвенного доступа. Пример объявления структуры: struct str\_name

Структуры можно использовать в качестве параметров функций, как и обычные переменные. Для структур поддерживаются все три механизма передачи данных: по значению, через указатели и по ссылке.

**Постановка задачи**

Необходимо создать программу, позволяющую:

1.   Создание новой записи о студенте.

2.   Внесение изменений в уже имеющуюся запись.

3.   Вывод всех данных о студентах.

4.   Вывод информации обо всех студентах группы *N*. *N* – инициализируется пользователем.

5.   Вывод топа самых успешных студентов с наивысшим по рейтингу средним баллом за прошедшую сессию.

6.   Вывод количества студентов мужского и женского пола.

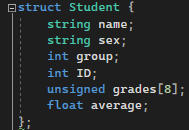
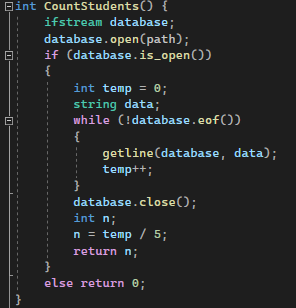
7.   Вывод данных о студентах, которые не получают стипендию; учатся только на «хорошо» и «отлично»; учатся только на «отлично»;

8.   Вывод данных о студентах, имеющих номер в списке – *k*.

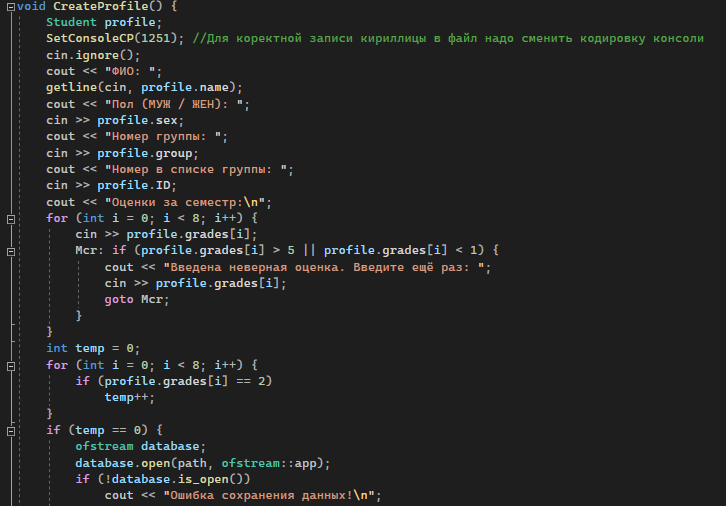
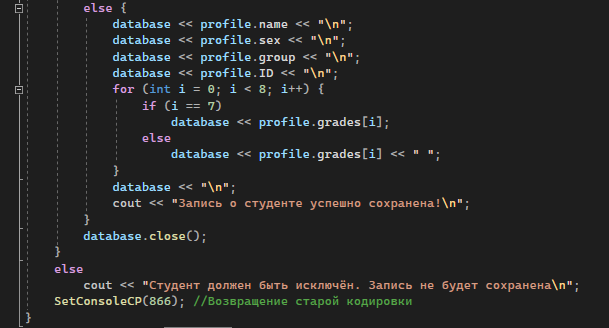
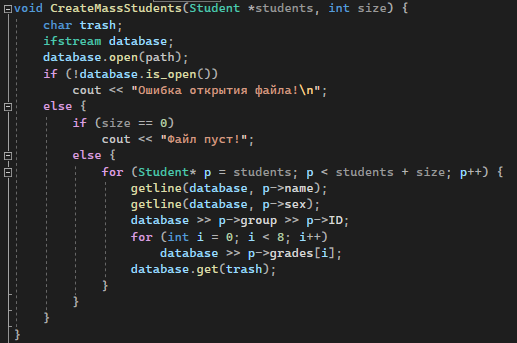
**Выполнение работы.**

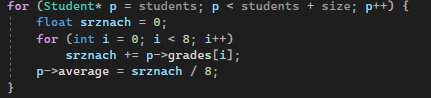
Код программы представлен в приложении А.

**Блок описания кода и использованных алгоритмов:**

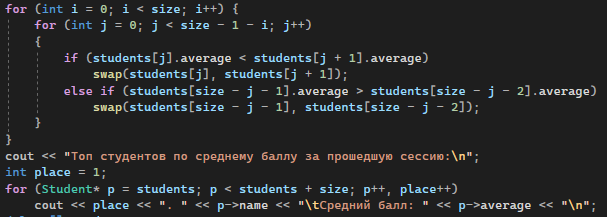
Создаём динамический массив структур, размер которого узнаём, считывая количество строк в изначальном файле Students.txt.

1. Создаётся элемент структуры, данные в который записывает пользователь, и сохраняется в файл.

Перед выполнением любого другого пункта программы создаётся динамический массив, который удаляется после выполнения пункта.

1. Пользователь вводит имя студента, данные которого хочет изменить, а затем меняет их.
2. Циклом for по очереди выводим всех студентов в списке.
3. Пользователь вводит номер группы и перебором выводятся студенты с соответствующим номером группы.
4. Сначала создаём функцию для высчитывания среднего значения оценок обучающегося.

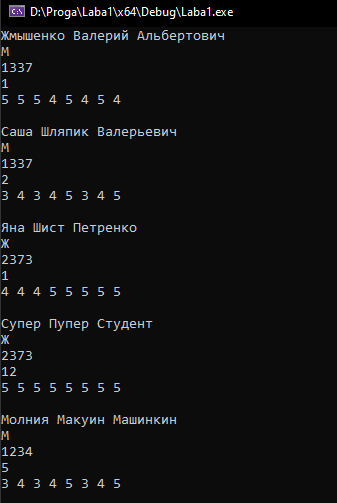
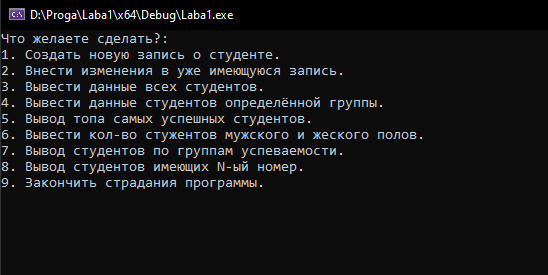
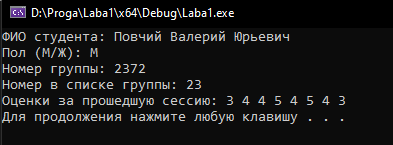
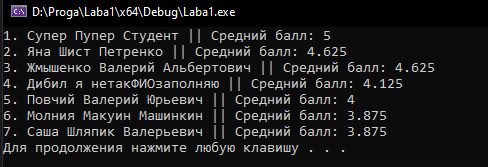
Далее сортируем массив алгоритмом Bubble sort.



1. Перебором проходимся по всему массиву и увеличиваем счётчики обеих полов.
2. Делаем проверку оценок студентов. Если находим соответствие, то выводим в консоль.
3. Пользователь вводит номер и перебором выводятся студенты с соответствующим номеров в группе.

**Выводы.**

Мы научились использовать структуры c++, написали прототип баз данный c++, научились использовать функционал структур.

**Блок скриншотов работы программы**

Приложение А

рабочий код

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

using namespace std;

string path = "C:\\Users\\MAX\\Desktop\\Prog\\students.txt";

struct Student {

string name;

string sex;

int group;

int ID;

unsigned grades[8];

float average;

};

void CreateProfile() {

Student profile;

SetConsoleCP(1251); //Для коректной записи кириллицы в файл надо сменить кодировку консоли

cin.ignore();

cout << "ФИО: ";

getline(cin, profile.name);

cout << "Пол (МУЖ / ЖЕН): ";

cin >> profile.sex;

cout << "Номер группы: ";

cin >> profile.group;

cout << "Номер в списке группы: ";

cin >> profile.ID;

cout << "Оценки за семестр:\n";

for (int i = 0; i < 8; i++) {

cin >> profile.grades[i];

Mcr: if (profile.grades[i] > 5 || profile.grades[i] < 1) {

cout << "Введена неверная оценка. Введите ещё раз: ";

cin >> profile.grades[i];

goto Mcr;

}

}

int temp = 0;

for (int i = 0; i < 8; i++) {

if (profile.grades[i] == 2)

temp++;

}

if (temp == 0) {

ofstream database;

database.open(path, ofstream::app);

if (!database.is\_open())

cout << "Ошибка сохранения данных!\n";

else {

database << profile.name << "\n";

database << profile.sex << "\n";

database << profile.group << "\n";

database << profile.ID << "\n";

for (int i = 0; i < 8; i++) {

if (i == 7)

database << profile.grades[i];

else

database << profile.grades[i] << " ";

}

database << "\n";

cout << "Запись о студенте успешно сохранена!\n";

}

database.close();

}

else

cout << "Студент должен быть исключён. Запись не будет сохранена\n";

SetConsoleCP(866); //Возвращение старой кодировки

}

int CountStudents() {

ifstream database;

database.open(path);

if (database.is\_open())

{

int temp = 0;

string data;

while (!database.eof())

{

getline(database, data);

temp++;

}

database.close();

int n;

n = temp / 5;

return n;

}

else return 0;

}

void CreateMassStudents(Student \*students, int size) {

char trash;

ifstream database;

database.open(path);

if (!database.is\_open())

cout << "Ошибка открытия файла!\n";

else {

if (size == 0)

cout << "Файл пуст!";

else {

for (Student\* p = students; p < students + size; p++) {

getline(database, p->name);

getline(database, p->sex);

database >> p->group >> p->ID;

for (int i = 0; i < 8; i++)

database >> p->grades[i];

database.get(trash);

}

}

}

}

void ChangeData() {

int size;

size = CountStudents();

Student\* students = new Student[size];

CreateMassStudents(students, size);

if (size == 0) {

delete[] students;

system("pause");

return;

}

string chName;

int change;

cin.ignore();

SetConsoleCP(1251);

while (true) {

cout << "Введите ФИО студента, данные о котором вы ходите изменить, или выход, чтобы завершить изменения: ";

getline(cin, chName);

if (chName == "выход")

break;

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

if (students[i].name == chName) {

cout << "Что вы ходите изменить?\n" <<

"1 - ФИО.\n" <<

"2 - Пол.\n" <<

"3 - Номер группы.\n" <<

"4 - Номер в списке группы.\n" <<

"5 - Оценки за семестр.\n" <<

"0 - Отменить изменение.\n" <<

"Ваш выбор: ";

cin >> change;

switch (change) {

case 1:

cout << "Введите новые данные: ";

cin.ignore();

getline(cin, students[i].name);

break;

case 2:

cout << "Введите новые данные: ";

cin >> students[i].sex;

cin.ignore();

break;

case 3:

cout << "Введите новые данные: ";

cin >> students[i].group;

cin.ignore();

break;

case 4:

cout << "Введите новые данные: ";

cin >> students[i].ID;

cin.ignore();

break;

case 5:

cout << "Введите новые данные:\n";

for (int j = 0; j < 8; j++) {

cin >> students[i].grades[j];

Mch: if (students[i].grades[j] > 5 || students[i].grades[j] < 1) {

cout << "Введена неверная оценка. Введите ещё раз: ";

cin >> students[i].grades[j];

goto Mch;

}

}

cin.ignore();

break;

case 0:

cin.ignore();

break;

}

break;

}

if (i == (size - 1))

cout << "Данный студент не найден!\n";

}

}

ofstream database;

database.open(path);

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

cout << "Ошибка открытия файла!\n";

}

else {

for (Student\* p = students; p < students + size; p++) {

database << p->name << "\n" << p->sex << "\n" << p->group << "\n" << p->ID << "\n";

for (int i = 0; i < 8; i++) {

if (i == 7)

database << p->grades[i];

else

database << p->grades[i] << " ";

}

database << "\n";

}

}

database.close();

cout << "Сохранение успешно завершено!\n";

SetConsoleCP(866);

system("pause");

delete[] students;

}

void ShowData() {

int size;

size = CountStudents();

Student\* students = new Student[size];

CreateMassStudents(students, size);

if (size == 0) {

delete[] students;

return;

}

for (Student\* p = students; p < students + size; p++) {

cout << "ФИО: " << p->name << "\nПол: " << p->sex << "\nНомер группы: " <<p->group << "\nНомер в группе: " << p->ID << "\nОценки за семестр: ";

for (int i = 0; i < 8; i++)

cout << p->grades[i] << " ";

cout << "\n\n";

}

delete[] students;

}

void ShowGroup() {

int n;

cout << "Введите номер группы: ";

cin >> n;

int size;

size = CountStudents();

Student\* students = new Student[size];

CreateMassStudents(students, size);

if (size == 0) {

delete[] students;

return;

}

int temp = 0;

for (Student\* p = students; p < students + size; p++)

if (p->group == n) {

temp++;

cout << "ФИО: " << p->name << "\nПол: " << p->sex << "\nНомер в группе: " << p->ID << "\nОценки за семестр: ";

for (int i = 0; i < 8; i++)

cout << p->grades[i] << " ";

cout << "\n\n";

}

delete[] students;

if (temp == 0)

cout << "Студентов данной группы не найдено!\n";

}

void CountAverage(Student\* students, int size) {

for (Student\* p = students; p < students + size; p++) {

float srznach = 0;

for (int i = 0; i < 8; i++)

srznach += p->grades[i];

p->average = srznach / 8;

}

}

void ShowTop() {

int size;

size = CountStudents();

Student\* students = new Student[size];

CreateMassStudents(students, size);

if (size == 0) {

return;

delete[] students;

}

CountAverage(students, size);

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

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

{

if (students[j].average < students[j + 1].average)

swap(students[j], students[j + 1]);

else if (students[size - j - 1].average > students[size - j - 2].average)

swap(students[size - j - 1], students[size - j - 2]);

}

}

cout << "Топ студентов по среднему баллу за прошедшую сессию:\n";

int place = 1;

for (Student\* p = students; p < students + size; p++, place++)

cout << place << ". " << p->name << "\tСредний балл: " << p->average << "\n";

delete[] students;

}

void CountSex() {

int size;

size = CountStudents();

Student\* students = new Student[size];

CreateMassStudents(students, size);

int male = 0, female = 0;

if (size == 0) {

delete[] students;

return;

}

for (Student\* p = students; p < students + size; p++) {

if (p->sex == "МУЖ")

male++;

else

female++;

}

delete[] students;

cout << "Студентов мужского пола: " << male << "\nСтудентов женского пола: " << female << "\n";

}

void ShowByGrades() {

int size;

size = CountStudents();

Student\* students = new Student[size];

CreateMassStudents(students, size);

if (size == 0) {

delete[] students;

return;

}

string\* GreatStudents = new string[size];

string\* FineStudents = new string[size];

string\* SatisStudents = new string[size];

int g = 0, f = 0, s = 0; //отслеживаем кол-во студентов в каждой группе

bool four = 0, three = 0;

for (Student\* p = students; p < students + size; p++) {

for (int i = 0; i < 8; i++) {

if (p->grades[i] == 3) {

SatisStudents[s] = p->name; //без стипендии

s++;

three = 1;

break;

}

if (p->grades[i] == 4) {

four = 1;

}

}

if (!four && !three) {

GreatStudents[g] = p->name; //отличники

g++;

}

else if (four && !three) {

FineStudents[f] = p->name; //"хорошо" и "отлично"

f++;

}

four = 0;

three = 0;

}

delete[] students;

cout << "Студенты, которые не получают стипендию: ";

if (s == 0)

delete[] SatisStudents;

else {

for (string\* ss = SatisStudents; ss < SatisStudents + s; ss++) {

cout << \*ss;

if (ss != SatisStudents + s - 1)

cout << ", ";

}

delete[] SatisStudents;

}

cout << "\n";

cout << "Студенты, которые учатся только на «хорошо» и «отлично»: ";

if (f == 0)

delete[] FineStudents;

else {

for (string\* fs = FineStudents; fs < FineStudents + f; fs++) {

cout << \*fs;

if (fs != FineStudents + f - 1)

cout << ", ";

}

delete[] FineStudents;

}

cout << "\n";

cout << "Студенты, которые учатся только на «отлично»: ";

if (g == 0)

delete[] GreatStudents;

else {

for (string\* gs = GreatStudents; gs < GreatStudents + g; gs++) {

cout << \*gs;

if (gs != GreatStudents + g - 1)

cout << ", ";

}

delete[] GreatStudents;

}

cout << "\n";

}

void ShowID() {

int k;

cout << "Введите номер в списке группы: ";

cin >> k;

int size;

size = CountStudents();

Student\* students = new Student[size];

CreateMassStudents(students, size);

if (size == 0) {

delete[] students;

return;

}

int temp = 0;

for (Student\* p = students; p < students + size; p++)

if (p->ID == k) {

temp++;

cout << "ФИО: " << p->name << "\nПол: " << p->sex << "\nНомер группы: " << p->group << "\nОценки за семестр: ";

for (int i = 0; i < 8; i++)

cout << p->grades[i] << " ";

cout << "\n\n";

}

delete[] students;

if (temp == 0)

cout << "Студентов с данным номером в списке группы не найдено!\n";

}

int main() {

setlocale(0, "");

char menu;

while (true) {

cout << "\t\tМеню\n"

<< "1 - Создание новой записи о студенте.\n"

<< "2 - Внесение изменений в уже имеющуюся запись.\n"

<< "3 - Вывод всех данных о студентах.\n"

<< "4 - Вывод информации обо всех студентах группы N.\n"

<< "5 - Вывод топа самых успешных студентов с наивысшим по рейтингу средним баллом за прошедшую сессию.\n"

<< "6 - Вывод количества студентов мужского и женского пола.\n"

<< "7 - Вывод данных о студентах, которые не получают стипендию; учатся только на «хорошо» и «отлично»; учатся только на «отлично».\n"

<< "8 - Вывод данных о студентах, имеющих номер в списке - k.\n"

<< "0 - Выйти из программы\n"

<< "Выберите пункт меню: ";

cin >> menu;

switch (menu) {

case '1':

system("cls");

CreateProfile();

system("pause");

system("cls");

break;

case '2':

system("cls");

ChangeData();

system("cls");

break;

case '3':

system("cls");

ShowData();

system("pause");

system("cls");

break;

case '4':

system("cls");

ShowGroup();

system("pause");

system("cls");

break;

case '5':

system("cls");

ShowTop();

system("pause");

system("cls");

break;

case '6':

system("cls");

CountSex();

system("pause");

system("cls");

break;

case '7':

system("cls");

ShowByGrades();

system("pause");

system("cls");

break;

case '8':

system("cls");

ShowID();

system("pause");

system("cls");

break;

case '0':

return 0;

default:

cout << "Пункт меню выбран неправильно!\n";

system("pause");

system("cls");

cin.ignore();

}

}

}