**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №1**

**по дисциплине «Программирование»**

Тема: Типы данных, определяемые пользователем

|  |  |  |
| --- | --- | --- |
| Студент гр. |  | Михеева Э. Д. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2023

**Цель работы.**

Изучение и организация структур. Получение практических навыков работы со структурами. Определение преимуществ и недостатков использования структур.

**Основные теоретические положения.**

Структура – группа связанных между собой, как правило, разнотипных переменных, объединенных в единый объект, в отличие от массива, все элементы которого строго однотипны. В языке программирования C++ структура является видом класса и обладает всеми его свойствами.

Для определения структуры применяется ключевое слово *struct*. Каждая входящая в структуру переменная называется членом (полем, элементом) структуры и описывается типом данных и именем. Поля структуры могут быть любого типа данных.

На структуры во многом похожи объединения. Объединения также хранят набор элементов, но в отличие от структуры все элементы объединения имеют нулевое смещение. То есть разные элементы занимают в памяти один и тот же участок.

Объединения объявляются ключевым словом *union*. В памяти члены объединения располагаются по одному адресу. Объединения не могут хранить одновременно несколько различных значений, они позволяют интерпретировать несколькими различными способами содержимое одной и той же области памяти.

Перечисления служат для определения пользовательских типов данных, значения которых принадлежат одному из элементов в некотором списке именованных целочисленных констант. Перечисление задается ключевым словом *enum*.

Для упрощения чтения программы, можно задать некоторому типу данных новое имя с помощью ключевого слова *typedef*. Объявления *typedef* можно использовать для создания более коротких или более понятных имен для типов, уже определенных в языке C/C++ или объявленных пользователем.

**Постановка задачи.**

Необходимо создать массив структур, содержащий информацию о студентах: ФИО, пол, номер группы, номер в списке группы, оценки за прошедшую сессию.

Написать функции, реализующие операции со структурами:

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.
5. Вывод 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.

**Выводы.**

В ходе работы были изучены структур, получены практические навыки работы с ними. Также были определены преимущества и недостатки использования структур.

Приложение А

рабочий код

#include <iostream>  
#include <fstream>  
#include <Windows.h>  
#include <iomanip>  
#include <vector>  
using namespace std;  
  
const string subjects[] = {"АиГ", "матанализ", "программирование", "философия", "физика", "английский", "история", "экология"};  
  
void line(){  
 //cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";  
 cout << "——————————————————————————————";  
}  
  
struct Student {  
 string fullName;  
 string sex;  
 int group{};  
 int listNum{};  
 int grades[8]{};  
 float average{};  
};  
  
void loadFromFile(Student students[], int& numOfStud){  
 ifstream read;  
 read.open("C:/Users/abram/CLionProjects/work1/stud.txt");  
 while (!read.eof()) {  
 string name, surname, patronymic, full;  
 read >> surname;  
 read >> name;  
 read >> patronymic;  
 full = surname + ' ' + name + ' ' + patronymic;  
 students[numOfStud].fullName = full;  
 read >> students[numOfStud].group;  
 read >> students[numOfStud].listNum;  
 for (int i = 0; i < 8; ++i) read >> students[numOfStud].grades[i];  
 numOfStud++;  
 }  
 numOfStud--;  
 read.close();  
}  
  
void getSex(Student students[], int& numOfStud){  
 for (int i = 0; i < numOfStud; i++){  
 string theLastElem;  
 theLastElem = students[i].fullName[students[i].fullName.length()-1];  
 if (theLastElem == "а") students[i].sex = "жен";  
 else students[i].sex = "муж";  
 }  
}  
  
void loadToFile(Student students[], int& numOfStud){  
 ofstream write;  
 write.open("C:/Users/abram/CLionProjects/work1/stud.txt");  
 for (int i = 1; i < numOfStud; i++){  
 write << students[i].fullName << endl << students[i].sex << endl << students[i].group << endl << students[i].listNum << endl;  
 for (int j = 0; j < 8; j++) {  
 write << students[i].grades[j];  
 }  
 write << endl;  
 }  
 write.close();  
}  
  
void addStudents(Student students[], int& numOfStud){  
 int num;  
 cout << "\n\nвведите количество студентов, которое хотите внести в список: ";  
 cin >> num;  
 for (int i = 1; i <= num; i++){  
 cout << "\nстудент " << i << "\n";  
 cout << "\nФИО: ";  
 getline(cin.ignore(), students[numOfStud].fullName);  
 cout << "\nномер группы: ";  
 cin >> students[numOfStud].group;  
 cout << "\nоценки:\n";  
 for (int j = 0; j < 8; j++){  
 cout << subjects[j] << ": ";  
 cin >> students[numOfStud].grades[j];  
 if (students[numOfStud].grades[j] <= 2 || students[numOfStud].grades[j] > 5){  
 cout << "ошибка записи, введите значение снова\n";  
 j--;  
 }  
 }  
 int numInList = 0;  
 for (int g = 0; g < numOfStud; g++){  
 if (students[numOfStud].group == students[g].group){  
 if (students[g].listNum > numInList) numInList = students[g].listNum;  
 }  
 }  
 students[numOfStud].listNum = numInList+1;  
 numOfStud = numOfStud+1;  
 }  
 getSex(students, numOfStud);  
}  
  
void editStudent(Student students[], int& numOfStud){  
 int group, num;  
 bool YorN;  
 int param;  
 string sub;  
 cout << "\nвведите номер группы студента, данные которого хотите изменить: ";  
 cin >> group;  
 cout << "\nвведите его номер в списке группы: ";  
 cin >> num;  
 for (int i = 0; i < numOfStud; i++){  
 if (group != students[i].group || num != students[i].listNum) continue;  
 if (group == students[i].group && num == students[i].listNum) cout << "\nвы хотите изменить данные студента " << students[i].fullName << "? (1 - да, 0 - нет)" << endl;  
 cin >> YorN;  
 while (YorN){  
 cout << "\nкакую информацию о студенте нужно изменить? (для выхода введите 0)";  
 cout << "\n1. ФИО: " << students[i].fullName << "\n2. пол: " << students[i].sex << "\n3. группа: " << students[i].group << "\n4. оценки\n" << endl;  
 cin >> param;  
 switch(param){  
 case 0: break;  
 case 1:{  
 cout << "\nФИО: ";  
 getline(cin.ignore(), students[i].fullName);  
 } break;  
 case 2:{  
 cout << "\n2. пол (муж/жен): ";  
 cin >> students[i].sex;  
 } break;  
 case 3:{  
 cout << "\n3. группа: ";  
 cin >> students[i].group;  
 } break;  
 case 4: {  
 cout << "\nвот список оценок:" << endl;  
 for (int j = 0; j < 8; j++) cout << subjects[j] << ": " << students[i].grades[j] << endl;  
 cout << "\nоценку по какому предмету требуется изменить?" << endl;  
 cin >> sub;  
 for (int k = 0; k < 8; k++) {  
 if (sub == subjects[k]) {  
 cout << "\nвведите оценку по предмету " << sub << ": ";  
 cin >> students[i].grades[k];  
 if (students[i].grades[k] <= 2 || students[i].grades[k] > 5){  
 cout << "\nошибка записи, введите значение снова\n";  
 cin >> students[i].grades[k];  
 }  
 }  
 }  
 cout << "\nоценка успешно перезаписана!" << endl;  
 } break;  
 }  
 cout << "\n\tперезаписанная информация о студенте";  
 cout << "\n1. ФИО: " << students[i].fullName << "\n2. пол: " << students[i].sex << "\n3. группа: " << students[i].group << "\n4. оценки:" << endl;  
 for (int o = 0; o < 8; o++) cout << "\t" <<subjects[o] << ": " << students[i].grades[o] << endl;  
 cout << "\nтребуется ли изменить ещё что-либо?\n(1 - да, 0 - нет)" << endl;  
 cin >> YorN;  
 }  
 }  
 getSex(students, numOfStud);  
}  
  
void printListOfGroup(Student students[], int& numOfStud){  
 int group;  
 cout << "\nвведите номер группы, для которой нужно вывести информацию о студентах: ";  
 cin >> group;  
 cout << "\n\tданные студентов группы " << group << endl;  
 for (int i = 0; i < numOfStud; i++){  
 if (group != students[i].group) continue;  
 if (group == students[i].group){  
 cout << "\n" << students[i].listNum << ". " << students[i].fullName << "\nпол: " << students[i].sex << "\n\tоценки" << endl;  
 for (int o = 0; o < 8; o++) cout << "\t" << subjects[o] << ": " << students[i].grades[o] << endl;  
 line();  
 }  
 }  
}  
  
void printAllStudents(Student students[], int& numOfStud){  
 cout << "\n\tданные всех студентов\n" << endl;  
 for (int i = 1; i < numOfStud; i++){  
 cout << "\n\tгруппа " << students[i].group << endl << students[i].listNum << ". " << students[i].fullName << endl;  
 cout << "пол: " << students[i].sex << endl << "\tоценки" << endl;  
 for (int o = 0; o < 8; o++) cout << subjects[o] << ": " << students[i].grades[o] << endl;  
 line();  
 }  
}  
  
void averageScore(Student students[], int& numOfStud){  
 for (int i = 0; i < numOfStud; i++){  
 float score = 0;  
 for (int o = 0; o < 8; o++) score += students[i].grades[o];  
 students[i].average = score/8;  
 }  
}  
  
void theBest(Student students[], int& numOfStud, int& top){  
 float allScores[numOfStud];  
 int indexes[numOfStud];  
 for (int i = 0; i < numOfStud; i++){  
 allScores[i] = students[i].average;  
 indexes[i] = i;  
 }  
 for (int i = 0; i < numOfStud; i++){  
 for (int j = i; j < numOfStud - 1 - i; j++){  
 if (allScores[j] > allScores[j+1]){  
 swap(allScores[j], allScores[j+1]);  
 swap(indexes[j], indexes[j+1]);  
 }  
 }  
 for (int j = numOfStud-1; j >= i; j--){  
 if (allScores[j] < allScores[j-1]){  
 swap(allScores[j], allScores[j-1]);  
 swap(indexes[j], indexes[j-1]);  
 }  
 }  
 }  
 cout << "\n\tтоп " << top << endl;  
 for (int t = 1; t <= top; t++) cout << t << ". " << students[indexes[numOfStud-t]].fullName << ": " << allScores[numOfStud-t] << endl;  
}  
  
void sexCount(Student students[], int& numOfStud){  
 int men(0), women(0);  
 for (int i = 0; i < numOfStud; i++){  
 string theLastElem;  
 theLastElem = students[i].fullName[students[i].fullName.length()-1];  
 (theLastElem == "а") ? women++ : men++;  
 }  
 cout << "\nколичество студентов женского пола: " << women << "\nколичество студентов мужского пола: " << men << endl;  
}  
  
void scholarship(Student students[], int& numOfStud){  
 vector<int> grad3, grade4, grade5;  
 bool tri;  
 for (int i = 0; i < numOfStud; i++){  
 tri = false;  
 for (int o = 0; o < 8; o++){  
 if (students[i].grades[o] == 3){  
 grad3.push\_back(i);  
 tri = true;  
 break;  
 }  
 }  
 if (tri != true) {  
 if (students[i].average >= 4 && students[i].average < 5) grade4.push\_back(i);  
 else if (students[i].average == 5) grade5.push\_back(i);  
 }  
 else continue;  
 }  
  
 int choice;  
 cin >> choice;  
 switch(choice){  
 case 1: {  
 cout << "\n\tстуденты, учащиеся на ОТЛИЧНО";  
 for (int i = 0; i < grade5.size(); i++) {  
 cout << "\nФИО: " << students[grade5[i]].fullName << "\nпол: " << students[grade5[i]].sex  
 << "\nгруппа: " << students[grade5[i]].group <<  
 "\nномер в списке группы: " << students[grade5[i]].listNum << "\nоценки:" << endl;  
 for (int o = 0; o < 8; o++)  
 cout << "\t" << subjects[o] << ": " << students[grade5[i]].grades[o] << endl;  
 line();  
 } break;  
 };  
 case 2:{  
 cout << "\n\tстуденты, учащиеся на ХОРОШО и ОТЛИЧНО";  
 for (int i = 0; i < grade4.size(); i++) {  
 cout << "\nФИО: " << students[grade4[i]].fullName << "\nпол: " << students[grade4[i]].sex  
 << "\nгруппа: " << students[grade4[i]].group <<  
 "\nномер в списке группы: " << students[grade4[i]].listNum << "\nоценки:" << endl;  
 for (int o = 0; o < 8; o++)  
 cout << "\t" << subjects[o] << ": " << students[grade4[i]].grades[o] << endl;  
 line();  
 } break;  
 };  
 case 3:{  
 cout << "\n\tстуденты, НЕ получающие стипендию";  
 for (int i = 0; i < grad3.size(); i++) {  
 cout << "\nФИО: " << students[grad3[i]].fullName << "\nпол: " << students[grad3[i]].sex  
 << "\nгруппа: " << students[grad3[i]].group <<  
 "\nномер в списке группы: " << students[grad3[i]].listNum << "\nоценки:" << endl;  
 for (int o = 0; o < 8; o++)  
 cout << "\t" << subjects[o] << ": " << students[grad3[i]].grades[o] << endl;  
 line();  
 } break;  
 };  
 }  
}  
  
void findWithNumber(Student students[], int& numOfStud){  
 int k;  
 vector<int> index;  
 cin >> k;  
 bool find;  
 for (int i = 0; i < numOfStud; i++) if (k == students[i].listNum) index.push\_back(i);  
 switch(index.size()){  
 case 0:{  
 cout << "\nстудентов с таким номером в списке не найдено" << endl;  
 }break;  
 case 1:{  
 cout << "\nнайден 1 студент с номером в списке " << k << endl;  
 }break;  
 default:{  
 cout << "\nстуденты с номером в списке " << k << endl;  
 }break;  
 }  
 for (int i = 0; i < index.size(); i++){  
 cout << "\nФИО: " << students[index[i]].fullName << "\nпол: " << students[index[i]].sex  
 << "\nгруппа: " << students[index[i]].group <<  
 "\nномер в списке группы: " << students[index[i]].listNum << "\nоценки:" << endl;  
 for (int o = 0; o < 8; o++)  
 cout << "\t" << subjects[o] << ": " << students[index[i]].grades[o] << endl;  
 line();  
 }  
}  
  
int main() {  
 SetConsoleCP(1251);  
 SetConsoleOutputCP(1251);  
 int numOfStud;  
 Student students[30];  
 loadFromFile(students, numOfStud);  
 getSex(students, numOfStud);  
 //averageScore(students, numOfStud);  
 cout << "\tменю:\n1. создать новую запись о студенте\n2. изменить имеющуюся запись\n3. вывести список группы\n4. узнать топ студентов"  
 "\n5. узнать колво студентов женского и мужского пола\n6. узнать успеваемость студентов (стипендия)\n7. найти студентов под нужным номером в списке\n8. вывести информацию обо всех студентах\n" << endl;  
 int kek(1), command;  
 while (kek) {  
 cout << "введите номер пункта, который требуется выполнить: ";  
 cin >> command;  
 switch(command){  
 case 1:{  
 addStudents(students, numOfStud);  
 }break;  
  
 case 2:{  
 editStudent(students, numOfStud);  
 }break;  
  
 case 3:{  
 printListOfGroup(students, numOfStud);  
 }break;  
  
 case 4:{  
 averageScore(students, numOfStud);  
 int top;  
 cout << "введите размер топа: ";  
 cin >> top;  
 theBest(students, numOfStud, top);  
 }break;  
  
 case 5:{  
 sexCount(students, numOfStud);  
 }break;  
  
 case 6:{  
 cout << "\nо каких студентах вы хотите получить информацию?\n1) отличники\n2) учащиеся на хорошо и отлично\n3) не получающие стипендию\nвведите: ";  
 scholarship(students, numOfStud);  
 }break;  
  
 case 7:{  
 cout << "\nвведите, под каким номером в списке студенты вас интересуют: ";  
 findWithNumber(students, numOfStud);  
 }break;  
  
 case 8:{  
 printAllStudents(students, numOfStud);  
 cout << endl;  
 }break;  
 }  
 loadToFile(students, numOfStud);  
 cout << "\n продолжить выполнение? (1 - да, 0 - нет)" << endl;  
 cin >> kek;  
 }  
 return 0;  
}