// Structure&DB.cpp :#include <iostream>

#include <fstream>

#include <string>

#include <iomanip>

struct Student {

int id;

std::string name; //имя

std::string faculty; //факультет

std::string speciality; //направление

int age; //возраст

float average; //ссредний балл

int course; //какой курс

};

int loadData(const std::string& filename, Student\*\*& students) { //обрабатываем и считываем бд

std::ifstream file(filename);

int count = 0; //тип число студентов

if (file.is\_open()) {

std::string line;

while (std::getline(file, line)) {

count++;

} //когда цикл прошел мы посчитали всех студентов

file.clear();

file.seekg(0);

students = new Student \* [count]; //выделаяем указатели, зависит от числа студентов

// Считываем данные в массив указателей

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

students[i] = new Student;

file >> students[i]->id >> students[i]->name >> students[i]->faculty >> students[i]->speciality>> students[i]->age >> students[i]->average >> students[i]->course;

} //для каждого студента записали все в ячейеку памяти

file.close();

}

else {

std::cout << "Не удалось открыть файл." << std::endl; //бд нет, работоспособность не проверить, хз че ему не нравитсся

}

return count;

}

void displayData(Student\*\* students, int count) { // выводим все

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

std::cout << "ID: " << students[i]->id

<< ", Имя: " << students[i]->name

<< ", Факультет: " << students[i]->faculty

<< ", Специальность: " << students[i]->speciality

<< ", Возраст: " << students[i]->age

<< ", Средний балл: " << students[i]->average

<< ", Курс:" << students[i]->course << std::endl;

}

}

// Функция для редактирования записи по ID

void editData(Student\*\* students, int count, int id) {

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

if (students[i]->id == id) {

std::cout << "Редактирование студента с ID " << id << ", введите новые данные:" << std::endl;

std::cout << "Имя: "; //если очень коротко, то типа cin - вводим, cout - выводим

std::cin >> students[i]->name;

std::cout << "Факультет: ";

std::cin >> students[i]->faculty;

std::cout << "Специальность: ";

std::cin >> students[i]->speciality;

std::cout << "Возраст: ";

std::cin >> students[i]->age;

std::cout << "Средний балл: ";

std::cin >> students[i]->average;

std::cout << "Курс: ";

std::cin >> students[i]->course;

break;

}

}

}

float calculateAverage(Student\*\* students, int count) { //средний балл посчитать

float sum = 0;

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

sum += students[i]->average;

}

return count > 0 ? sum / count : 0; //сокращенная запись if else

}

void saveData(const std::string& filename, Student\*\* students, int count) { //в ноый файл записываем

std::ofstream file(filename);

if (file.is\_open()) {

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

file << students[i]->id << " "

<< students[i]->name << " "

<< students[i]->faculty << " "

<< students[i]->speciality << " "

<< students[i]->age << " "

<< students[i]->average << " "

<< students[i]->course << std::endl;

}

file.close();

}

else {

std::cout << "Не удалось открыть файл." << std::endl;

}

}

void freeMemory(Student\*\*& students, int count) { //освобождаемм выделенную память

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

delete students[i];

}

delete[] students;

students = nullptr;

}

int main() {

Student\*\* students = nullptr;

std::string filename = "студенты.txt";

int choice, id, count;

count = loadData(filename, students);

do {

std::cout << "1. Просмотр всех данных\n2. Редактирование данных по ID\n3. Рассчет среднего баллаA\n4. Сохранение данных\n5. Выход\nВыберите опцию: ";

std::cin >> choice;

switch (choice) {

case 1:

displayData(students, count);

break;

case 2:

std::cout << "Введите ID для редактирования: ";

std::cin >> id;

editData(students, count, id);

break;

case 3:

std::cout << "Средний балл: " << calculateAverage(students, count) << std::endl;

break;

case 4:

saveData("новый.txt", students, count);

std::cout << "Данные сохранены в файл новый.txt" << std::endl;

break;

case 5:

std::cout << "Выход..." << std::endl;

break;

default:

std::cout << "Неверный выбор. Попробуйте снова." << std::endl;

break;

}

} while (choice != 5);

freeMemory(students, count); //lynyrd skynyrd - free bird

return 0;

}