#include <iostream>

#include <string>

#include <iomanip>

class Student {

private:

int id;

std::string name;

std::string faculty;

std::string speciality;

int age;

float average;

int course;

public:

// Конструктор по умолчанию

Student() : id(0), name("Unknown"), faculty("Unknown"), speciality("Unknown"), age(0), average(0.0), course(0) {}

// Конструктор с параметрами

Student(int id, std::string name, std::string faculty, std::string speciality, int age, float average, int course)

: id(id), name(name), faculty(faculty), speciality(speciality), age(age), average(average), course(course) {}

// Методы для получения значений полей

int getId() const { return id; }

std::string getName() const { return name; }

std::string getFaculty() const { return faculty; }

std::string getSpeciality() const { return speciality; }

int getAge() const { return age; }

float getAverage() const { return average; }

int getCourse() const { return course; }

// Методы для установки значений полей

void setId(int newId) { id = newId; }

void setName(const std::string& newName) { name = newName; }

void setFaculty(const std::string& newFaculty) { faculty = newFaculty; }

void setSpeciality(const std::string& newSpeciality) { speciality = newSpeciality; }

void setAge(int newAge) { age = newAge; }

void setAverage(float newAverage) { average = newAverage; }

void setCourse(int newCourse) { course = newCourse; }

// Метод для отображения информации о студенте

void display() const {

std::cout << "ID: " << id

<< ", Имя: " << name

<< ", Факультет: " << faculty

<< ", Специальность: " << speciality

<< ", Возраст: " << age

<< ", Средний балл: " << average

<< ", Курс :" << course << std::endl;

}

};

// Функция для создания массива объектов студентов с использованием конструктора по умолчанию

Student\* createArray(int size) {

return new Student[size];

}

// Функция для создания массива объектов студентов с использованием конструктора с параметрами

Student\* createArray(int size, int id, std::string name, std::string faculty, std::string speciality, int age, float average, int course) {

Student\* students = new Student[size];

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

students[i] = Student(id, name, faculty, speciality, age, average, course);

}

return students;

}

// Функция для редактирования свойств конкретного студента в массиве

void editStudent(Student& student) {

int id;

std::string name, faculty, speciality;

int age;

float average;

int course;

std::cout << "Введите новые данные для студента:\n";

std::cout << "ID: ";

std::cin >> id;

std::cout << "Имя: ";

std::cin >> name;

std::cout << "Факультет: ";

std::cin >> faculty;

std::cout << "Специальность: ";

std::cin >> speciality;

std::cout << "Возраст: ";

std::cin >> age;

std::cout << "Средний балл: ";

std::cin >> average;

std::cout << "Курс: ";

std::cin >> course;

student.setId(id);

student.setName(name);

student.setFaculty(faculty);

student.setSpeciality(speciality);

student.setAge(age);

student.setAverage(average);

student.setCourse(course);

}

// Функция для обработки массива объектов (вывод информации)

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

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

students[i].display();

}

}

int main() {

setlocale(LC\_ALL, "");

int size;

std::cout << "Введите количество студентов: ";

std::cin >> size;

// Создание массива объектов с использованием конструктора по умолчанию

Student\* students = createArray(size);

// Редактирование данных первого студента в массиве

if (size > 0) {

std::cout << "Редактирование данных первого студента:\n";

editStudent(students[0]);

}

// Вывод информации о всех студентах

std::cout << "\nИнформация о студентах:\n";

processArray(students, size);

// Освобождение памяти

delete[] students;

return 0;

}