**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: «**Изучение и организация структур».**

|  |  |  |
| --- | --- | --- |
| Студент(ка) гр. 4372 |  | Вердян А.Р. |
| Преподаватель |  | Глущенко А. Г. |

г. Санкт-Петербург

2025 г.

**Цель работы:** изучение и организация структур; получение практических навыков работы со структурами; определение преимуществ и недостатков использования структур.

**Основные теоретические положения:**

**Структуры** представляют собой группы связанных между собой, как правило, разнотипных переменных, объединенных в единый объект, в отличие от массива, все элементы которого однотипны. В языке C++ структура является видом класса и обладает всеми его свойствами.

Для определения структуры применяется ключевое слово **struct**, а сам формат определения выглядит следующим образом:

struct [имя\_типа] {

тип\_1 элемент\_1;

тип \_2 элемент\_2;

…

тип\_k элемент\_k;

} [ список\_описателей ];

Каждая входящая в структуру переменная называется членом (полем, элементом) структуры и описывается типом данных и именем. Поля структуры могут быть любого типа данных. Их количество не лимитировано.

**Инициализация структур** аналогична инициализации массивов: в фигурных скобках передаются значения для элементов структуры по порядку. Так как в структуре **student** первым определено свойство, которое представляет тип **int**- число, то в фигурных скобках вначале идет число. И так далее для всех элементов структуры по порядку.

При определении переменной структуры ее можно сразу инициализировать, присвоив какое-нибудь значение:

struct Student ivan = {23, "Ivan", 0393};

Инициализировать структуру можно сразу после её объявления. Для инициализации структуры значения ее элементов перечисляют в фигурных скобках в порядке их описания:

struct Complex {

float real, im;

} data [2][2] = {{{1,1}, {2,2}}, {{3,3}, {4,4}}};

Все поля структурных переменных располагаются в непрерывной области памяти одно за другим. Общий объем памяти, занимаемый структурой, равен сумме размеров всех полей структуры. Для определения размера структуры следует использовать инструкцию **sizeof().**

В программировании очень часто используются такие конструкции, как массивы структур. Все форматы определения массива структур будут аналогичны определению массивов других типов:

struct Students students[30];

Любая структурная переменная занимает в памяти определенное положение, характеризующееся конкретным адресом. Для работы с адресами структурных переменных (как и для простых переменных) можно использовать указатели. **Указатели** на структурные переменные определяются точно так же, как и для обычных переменных. Разыменование указателя (обращение к данным по адресу, хранящемуся в указателе) осуществляется также обычным образом.

Через указатели можно работать с отдельными полями структур. Для доступа к полю структуры через указатель используется оператор «🡪» («стрелка»), а не «точка».

**Постановка задачи:** необходимо разработать алгоритм и написать программу, которая реализует следующие задачи:

1. Создаёт новую запись 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.

Описание кода и использованных алгоритмов:

При запуске программы перед пользователем появляется окно, в котором на экран выводится:

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*.

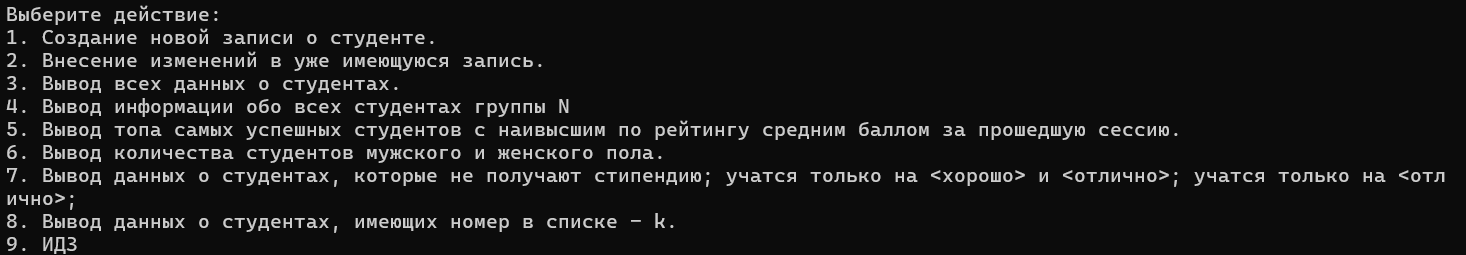
Работа алгоритма и вывод на экран:

Рис.1 Меню для выбора действия.

**Вывод:** В ходе практической работы было изучено написание структур, их инициализация, а также способы работы со структурами.

Приложение А

**РАБОЧИЙ КОД:**

**#include <iostream>**

**#include <string>**

**#include <iomanip>**

**using namespace std;**

**const int MAX\_STUDENTS = 100; // Максимальное количество студентов**

**struct Student {**

**string full\_name\_FIO;**

**string gender;**

**string number\_group;**

**int number\_list;**

**int exam\_grades[3];**

**int test\_grades[5];**

**};**

**Student students[MAX\_STUDENTS];**

**int student\_count = 0;**

**void new\_entry\_student() {**

**if (student\_count >= MAX\_STUDENTS) {**

**cout << "Достигнуто максимальное количество студентов!" << endl;**

**return;**

**}**

**Student newStudent;**

**cout << "Введите ФИО студента: ";**

**cin.ignore();**

**getline(cin, newStudent.full\_name\_FIO);**

**cout << "Введите пол (М/F): ";**

**cin >> newStudent.gender;**

**cout << "Введите номер группы: ";**

**cin >> newStudent.number\_group;**

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

**cin >> newStudent.number\_list;**

**cout << "Введите оценки за экзамены (3 оценки): ";**

**for (int i = 0; i < 3; ++i) {**

**cin >> newStudent.exam\_grades[i];**

**}**

**cout << "Введите оценки за дифференцированные зачеты (5 оценок): ";**

**for (int i = 0; i < 5; ++i) {**

**cin >> newStudent.test\_grades[i];**

**}**

**students[student\_count++] = newStudent;**

**}**

**void change\_student() {**

**int listNumber;**

**cout << "Введите номер в списке группы студента для редактирования: ";**

**cin >> listNumber;**

**for (int i = 0; i < student\_count; ++i) {**

**if (students[i].number\_list == listNumber) {**

**cout << "Редактирование информации о студенте: " << students[i].full\_name\_FIO << endl;**

**cout << "Введите новое ФИО студента: ";**

**cin.ignore();**

**getline(cin, students[i].full\_name\_FIO);**

**cout << "Введите новый пол (М/F): ";**

**cin >> students[i].gender;**

**cout << "Введите новый номер группы: ";**

**cin >> students[i].number\_group;**

**cout << "Введите новые оценки за экзамены (3 оценки): ";**

**for (int j = 0; j < 3; ++j) {**

**cin >> students[i].exam\_grades[j];**

**}**

**cout << "Введите новые оценки за дифференцированные зачеты (5 оценок): ";**

**for (int j = 0; j < 5; ++j) {**

**cin >> students[i].test\_grades[j];**

**}**

**return;**

**}**

**}**

**cout << "Студент с номером в списке " << listNumber << " не найден." << endl;**

**}**

**void all\_data\_students() {**

**for (int i = 0; i < student\_count; ++i) {**

**cout << "ФИО: " << students[i].full\_name\_FIO << endl;**

**cout << "Пол: " << students[i].gender << endl;**

**cout << "Группа: " << students[i].number\_group << endl;**

**cout << "Номер в списке группы: " << students[i].number\_list << endl;**

**cout << "Оценки за экзамены: ";**

**for (int j = 0; j < 3; ++j) {**

**cout << students[i].exam\_grades[j] << " ";**

**}**

**cout << endl;**

**cout << "Оценки за дифференцированные зачеты: ";**

**for (int j = 0; j < 5; ++j) {**

**cout << students[i].test\_grades[j] << " ";**

**}**

**cout << endl;**

**}**

**}**

**void inf\_on\_group\_students\_N() {**

**string groupNumber;**

**cout << "Введите номер группы: ";**

**cin >> groupNumber;**

**for (int i = 0; i < student\_count; ++i) {**

**if (students[i].number\_group == groupNumber) {**

**cout << "ФИО: " << students[i].full\_name\_FIO << endl;**

**cout << "Пол: " << students[i].gender << endl;**

**cout << "Номер в списке группы: " << students[i].number\_list << endl;**

**cout << "Оценки за экзамены: ";**

**for (int j = 0; j < 3; ++j) {**

**cout << students[i].exam\_grades[j] << " ";**

**}**

**cout << endl;**

**cout << "Оценки за дифференцированные зачеты: ";**

**for (int j = 0; j < 5; ++j) {**

**cout << students[i].test\_grades[j] << " ";**

**}**

**cout << endl;**

**}**

**}**

**}**

**void top\_students() {**

**double maxAverage = -1.0;**

**// Находим максимальный средний балл**

**for (int i = 0; i < student\_count; ++i) {**

**double totalGrades = 0.0;**

**for (int j = 0; j < 3; ++j) {**

**totalGrades += students[i].exam\_grades[j];**

**}**

**for (int j = 0; j < 5; ++j) {**

**totalGrades += students[i].test\_grades[j];**

**}**

**double average = totalGrades / 8;**

**if (average > maxAverage) {**

**maxAverage = average;**

**}**

**}**

**// Выводим студентов с максимальным средним баллом**

**cout << "Топ студентов с наивысшим средним баллом:" << endl;**

**for (int i = 0; i < student\_count; ++i) {**

**double totalGrades = 0.0;**

**for (int j = 0; j < 3; ++j) {**

**totalGrades += students[i].exam\_grades[j];**

**}**

**for (int j = 0; j < 5; ++j) {**

**totalGrades += students[i].test\_grades[j];**

**}**

**double average = totalGrades / 8;**

**if (average == maxAverage) {**

**cout << "ФИО: " << students[i].full\_name\_FIO << ", Средний балл: " << fixed << setprecision(2) << average << endl;**

**}**

**}**

**}**

**void count\_gender\_student() {**

**int man\_count = 0, woman\_count = 0;**

**for (int i = 0; i < student\_count; ++i) {**

**if (students[i].gender == "M") {**

**man\_count++;**

**}**

**else if (students[i].gender == "F") {**

**woman\_count++;**

**}**

**}**

**cout << "Количество студентов мужского пола: " << man\_count << endl;**

**cout << "Количество студентов женского пола: " << woman\_count << endl;**

**}**

**void scholarship\_students() {**

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

**for (int i = 0; i < student\_count; ++i) {**

**bool hasScholarship = false;**

**for (int j = 0; j < 3; ++j) {**

**if (students[i].exam\_grades[j] < 4) { // Если есть оценка ниже 4**

**hasScholarship = true;**

**break;**

**}**

**}**

**for (int j = 0; j < 5 && !hasScholarship; ++j) {**

**if (students[i].test\_grades[j] < 4) {**

**hasScholarship = true;**

**break;**

**}**

**}**

**if (!hasScholarship) { // Если все оценки >= 4**

**cout << students[i].full\_name\_FIO << endl;**

**}**

**}**

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

**for (int i = 0; i < student\_count; ++i) {**

**bool only\_good\_and\_great = true;**

**for (int j = 0; j < 3; ++j) {**

**if (students[i].exam\_grades[j] < 4) { // Оценка ниже 4**

**only\_good\_and\_great = false;**

**break;**

**}**

**}**

**for (int j = 0; j < 5 && only\_good\_and\_great; ++j) {**

**if (students[i].test\_grades[j] < 4) {**

**only\_good\_and\_great = false;**

**break;**

**}**

**}**

**if (only\_good\_and\_great) {**

**cout << students[i].full\_name\_FIO << endl;**

**}**

**}**

**cout << "Студенты, учатся только на 'отлично':" << endl;**

**for (int i = 0; i < student\_count; ++i) {**

**bool only\_great = true;**

**for (int j = 0; j < 3; ++j) {**

**if (students[i].exam\_grades[j] != 5) { // Оценка не равна 5**

**only\_great = false;**

**break;**

**}**

**}**

**for (int j = 0; j < 5 && only\_great; ++j) {**

**if (students[i].test\_grades[j] != 5) {**

**only\_great = false;**

**break;**

**}**

**}**

**if (only\_great) {**

**cout << students[i].full\_name\_FIO << endl;**

**}**

**}**

**}**

**void student\_num\_list\_K() {**

**int listNumber;**

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

**cin >> listNumber;**

**for (int i = 0; i < student\_count; ++i) {**

**if (students[i].number\_list == listNumber) {**

**cout << "ФИО: " << students[i].full\_name\_FIO << endl;**

**cout << "Пол: " << students[i].gender << endl;**

**cout << "Группа: " << students[i].number\_group << endl;**

**cout << "Номер в списке группы: " << students[i].number\_list << endl;**

**cout << "Оценки за экзамены: ";**

**for (int j = 0; j < 3; ++j) {**

**cout << students[i].exam\_grades[j] << " ";**

**}**

**cout << "Оценки за дифференцированные зачеты: ";**

**for (int j = 0; j < 5; ++j) {**

**cout << students[i].test\_grades[j] << " ";**

**}**

**cout << endl;**

**return;**

**}**

**}**

**cout << "Студент с номером в списке не найден." << endl;**

**}**

**// idz**

**struct IdzStudent {**

**string full\_name;**

**char gender;**

**int age;**

**char clothing\_size; // Размер одежды (например, 'S', 'M', 'L', 'XL')**

**bool voll\_category; // Наличие разряда (true - есть, false - нет)**

**int number\_course;**

**};**

**void IdzStudents\_has\_voll\_category(IdzStudent IdzStudents[], int count) {**

**cout << "Студенты с разрядом по волейболу:\n";**

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

**if (IdzStudents[i].voll\_category) {**

**cout << IdzStudents[i].full\_name << "\n";**

**}**

**}**

**}**

**void women\_team(IdzStudent IdzStudents[], int count) {**

**cout << "Женская сборная волейбольной секции:\n";**

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

**if (IdzStudents[i].gender == 'F') {**

**cout << IdzStudents[i].full\_name << "\n";**

**}**

**}**

**}**

**void course\_IdzStudents(IdzStudent IdzStudents[], int count, int course) {**

**cout << "Студенты " << course << " курса:\n";**

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

**if (IdzStudents[i].number\_course == course) {**

**cout << IdzStudents[i].full\_name << "\n";**

**}**

**}**

**}**

**void man\_IdzStudents\_with\_a\_larger\_size(IdzStudent IdzStudents[], int count, char size) {**

**cout << "Мужские студенты с размером одежды больше чем S " << size << ":\n";**

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

**if (IdzStudents[i].gender == 'M' && IdzStudents[i].clothing\_size > size) {**

**cout << IdzStudents[i].full\_name << "\n";**

**}**

**}**

**}**

**int launch\_idz() {**

**const int IdzStudent\_count = 5;**

**IdzStudent IdzStudents[IdzStudent\_count] = {**

**{"Шашичев Никита Дмитриевич", 'M', 20, 'M', true, 2},**

**{"Павлова Елизавета Витальевна", 'W', 19, 'S', false, 1},**

**{"Кузнецов Алексей Сидорович", 'M', 21, 'L', true, 3},**

**{"Петрова Анна Сергеевна", 'W', 22, 'M', true, 2},**

**{"Михайлов Сергей Павлович", 'M', 20, 'L', false, 1}**

**};**

**IdzStudents\_has\_voll\_category(IdzStudents, IdzStudent\_count);**

**women\_team(IdzStudents, IdzStudent\_count);**

**course\_IdzStudents(IdzStudents, IdzStudent\_count, 2);**

**man\_IdzStudents\_with\_a\_larger\_size(IdzStudents, IdzStudent\_count, 'S');**

**return 0;**

**}**

**int main() {**

**setlocale(LC\_ALL, "Russian");**

**while (true) {**

**int choice;**

**cout << "Выберите действие:" << endl;**

**cout << "1. Создание новой записи о студенте." << endl;**

**cout << "2. Внесение изменений в уже имеющуюся запись." << endl;**

**cout << "3. Вывод всех данных о студентах." << endl;**

**cout << "4. Вывод информации обо всех студентах группы N" << endl;**

**cout << "5. Вывод топа самых успешных студентов с наивысшим по рейтингу средним баллом за прошедшую сессию." << endl;**

**cout << "6. Вывод количества студентов мужского и женского пола." << endl;**

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

**cout << "8. Вывод данных о студентах, имеющих номер в списке – k." << endl;**

**cout << "9. ИДЗ" << endl;**

**cin >> choice;**

**switch (choice) {**

**case 1:**

**new\_entry\_student();**

**break;**

**case 2:**

**change\_student();**

**break;**

**case 3:**

**all\_data\_students();**

**break;**

**case 4:**

**inf\_on\_group\_students\_N();**

**break;**

**case 5:**

**top\_students();**

**break;**

**case 6:**

**count\_gender\_student();**

**break;**

**case 7:**

**scholarship\_students();**

**break;**

**case 8:**

**student\_num\_list\_K();**

**break;**

**case 9:**

**launch\_idz();**

**break;**

**default:**

**cout << "Неверный выбор! Попробуйте снова." << endl;**

**break;**

**}**

**}**

**return 0;**

**}**