**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: **«Структуры»**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 0324 |  | Костюхина Э.В. |
| Преподаватель |  | Глущенко А.Г |

Санкт-Петербург

2021

**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.

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.

Кроме базовых примитивных типов данных как int или char, массивов и указателей в качестве элементов структуры можно использовать другие структуры.

Для того чтобы записать данные в структурную переменную, необходимо каждому полю структуры присвоить определенное значение. Для этого необходимо использовать оператор ‘.’ («точка»):

**Постановка задачи.**

Необходимо создать массив структур, содержащий информацию о студентах: ФИО, пол, номер группы, номер в списке группы, оценки за прошедшую сессию (всего 3 экзамена и 5 дифференцированных зачетов), форма обучения\*, отметка времени о внесении или изменении данных\*. Ввод и изменение данных обо всех студентах должен осуществляться в файл students\*.

**Выполнение работы.**

Для выполнения поставленной задачи была написана программа на языке C++.

Программа разбита на функции:

menu() – выводит список заданий для выбора.

1. Создание новой записи о студенте.

AddStudent(vector<Students>&stud, int n)

2. Внесение изменений в уже имеющуюся запись.

EditStudent(vector<Students>&stud, int n)

3. Вывод всех данных о студентах.

Info(vector<Students>&stud, int n)

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

func\_4(vector<Students>&stud, int n, int N)

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

top(vector<Students>&stud, int n)

6. Вывод количества студентов мужского и женского пола.

gender(vector<Students>&stud, int n)

7. Определение количества студентов, которые будут получать стипендию (стипендия начисляется, если у студента нет троек и очная форма обучения).\*

stipend(vector<Students>&stud, int n)

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

funk\_8(vector<Students>&stud, int n)

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

studNum(vector<Students>&stud, int n, int N)

10. Вывод всех записей, сделанных в день, который введет пользователь. Вывод всех записей, сделанных после полудня. Вывод всех записей, сделанных до полудня.\*

time\_day(vector<Students>&stud, int n)

time\_afternoon(vector<Students>&stud, int n)

time\_untilnoon(vector<Students>&stud, int n)

**Результаты работа программы.**



















