**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**"ЛЭТИ" ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра информационных систем и технологий**

**ОТЧЁТ**

**по практической работе №1**

**по дисциплине "Программирование"**

**Тема:** **Типы данных, определяемые пользователем**

Студент гр. 0324 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гордиенко Т.Е.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Глущенко А.Г.

Санкт-Петербург

2020

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

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\*.

Написать функции, реализующие операции со структурами (ввод данных с клавиатуры):

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

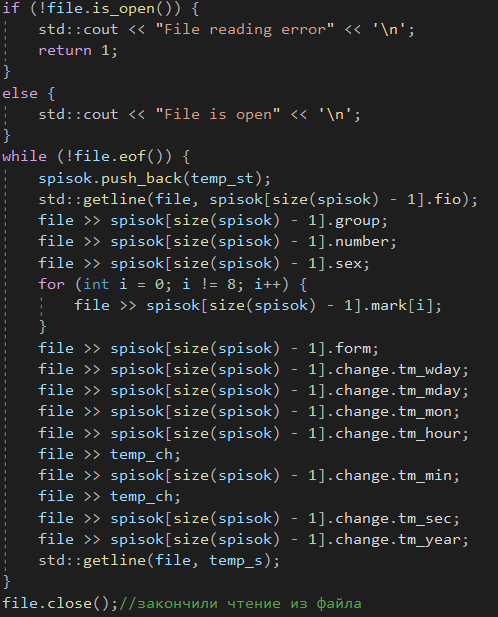
8.   Вывод данных о студентах, которые не получают стипендию; учатся только на «хорошо» и «отлично»; учатся только на «отлично»;

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

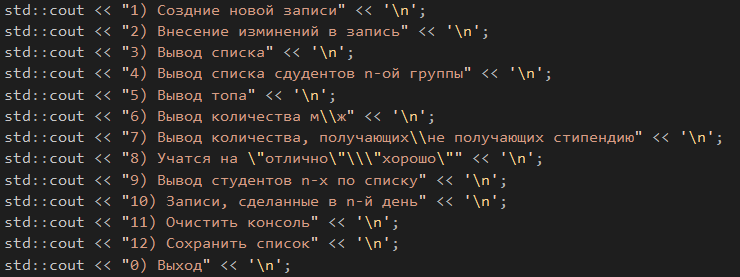
10.   Вывод всех записей, сделанных в день, который введет пользователь. Вывод всех записей, сделанных после полудня. Вывод всех записей, сделанных до полудня.\*

Выполнение:

При запуске программы идёт считывание данных из текстового файла и заполнение ими вектора «spisok».

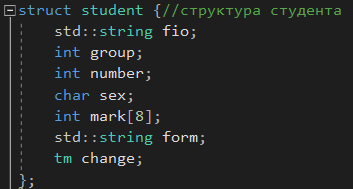


После достижения конца файла, он «закрывается», и пользователю предлагается ряд действий.

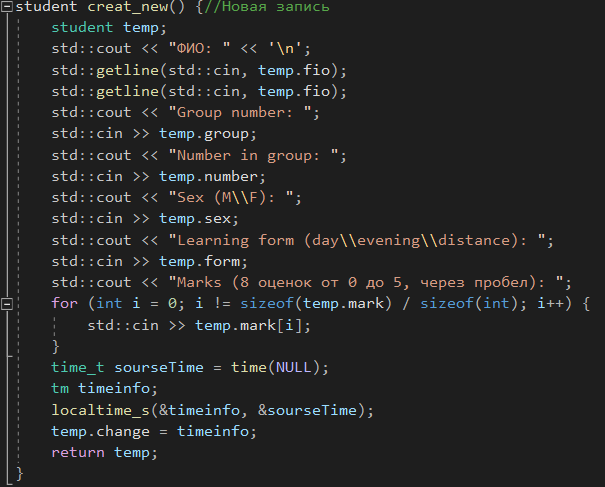


После выбора соответствующего пункта, выполняется соответствующая описанию, заранее написанная функция. Не вижу смысла описывать каждую. Остановлюсь только на нужных моментах.

Ниже представлена структура, которая предназначена для хранении информации об одном студенте. «tm» - это структура из библиотеки time.h для хранения информации о времени.



Благодаря localtime\_s получаем локальное время системы и записываем его в создаваемую запись. При «редактировании» происходит вызов этой же функции.



Для перевода tm на человеческий язык было написано:

switch (chelik.change.tm\_wday) {

case 0:

std::cout << "Sun ";

break;

case 1:

std::cout << "Mon ";

break;

case 2:

std::cout << "Tue ";

break;

case 3:

std::cout << "Wed ";

break;

case 4:

std::cout << "Thu ";

break;

case 5:

std::cout << "Fri ";

break;

case 6:

std::cout << "Sat ";

break;

default:

std::cout << "error ";

break;

}

std::cout << chelik.change.tm\_mday << " ";

switch (chelik.change.tm\_mon) {

case 0:

std::cout << "Jan ";

break;

case 1:

std::cout << "Feb ";

break;

case 2:

std::cout << "Mar ";

break;

case 3:

std::cout << "Apr ";

break;

case 4:

std::cout << "May ";

break;

case 5:

std::cout << "Jun ";

break;

case 6:

std::cout << "Jul ";

break;

case 7:

std::cout << "Aug ";

break;

case 8:

std::cout << "Sep ";

break;

case 9:

std::cout << "Oct ";

break;

case 10:

std::cout << "Nov ";

break;

case 11:

std::cout << "Sep ";

break;

default:

std::cout << "error ";

break;

}

std::cout << " " << chelik.change.tm\_hour << ":" << chelik.change.tm\_min << ":" << chelik.change.tm\_sec << " " << chelik.change.tm\_year + 1900 << '\n' << '\n';

Благодаря этому информация выводилась в более понятном для пользователя информация о времени.

Остальные задания отличались только условиями в if, так что не вижу смысла их объяснять. Исходник вы можете найти по ссылке https://github.com/TimPingvin/Work\_1\_2sem/tree/main.

Вывод: Мы научились пользоваться структурами. Они полезны при необходимости хранить информацию нестандартного вида. Единственный недостаток, который я заметил, это то, что в большинстве случаев придётся самостоятельно писать функции для вывода.