Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ № 17**

Структуры  
Вариант № 12

Преподаватель \_\_\_\_\_\_\_\_ Пушкарев К. В.

подпись, дата

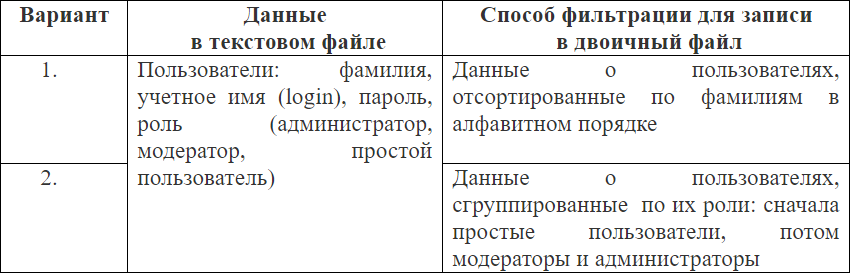
Студент КИ18-09б, 031830645 \_\_\_\_\_\_ 18.02.2019 Котов С.А.

подпись

Красноярск 2019

**1 Дополнительное упражнение № 2**

Разработать, отладить, продемонстрировать и защитить преподавателю графическую схему алгоритма и программу для решения следующей задачи: прочитать из текстового файла данные в массив структур, отфильтровать данные в соответствии с указаниями в таблице №17.3.2 и сохранить отфильтрованные данные в двоичном файле (имя файла вводится с клавиатуры). Вывести содержимое текстового и двоичного файла на монитор в виде таблицы.



**2 Цель работы**

Получить практические навыки решения задач с использованием структур.

**3 Графическая схема алгоритма**

Рисунок 1 – Блок-схема дополнительного упражнения № 2.

**4 Код программы**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129 | #include "pch.h"  #include <iostream>  #include <clocale>  #include <fstream>  #include <cstring>  #include <vector>  #include <iomanip>  using namespace std;  enum enum\_user{  users,  moder,  admin  };  struct user {  char surname[20];  char log[20];  char pass[20];  enum\_user user\_role;  };  // Заполнение структуры  void fill\_user(ifstream &fin, vector<user> &buf) {  user buffer;  fin >> buffer.surname >> buffer.log >> buffer.pass;  char role[20];  fin >> role;  if (strcmp(role,"user") == 0) {  buffer.user\_role = users;  }  else if (strcmp(role,"admin") == 0) {  buffer.user\_role = admin;  }  else {  buffer.user\_role = moder;  }  buf.push\_back(buffer);  }  // Сортировка списка пользователей  void sort(vector<user> &buf, int n) {  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (buf[i].user\_role > buf[j].user\_role) {  swap(buf[i], buf[j]);  }  }  }  }  // Вывод данных пользователя  void print\_user\_console(const user \*buf) {  cout.setf(ios::left);  cout << setw(20) << buf->surname;  cout << "\t" << setw(20) << buf->log << "\t" << setw(20) << buf->pass;  cout << "\t" << setw(20);  if (buf->user\_role == 0) {  cout << "пользователь" << endl;  }  else if (buf->user\_role == 1) {  cout << "модератор" << endl;  }  else {  cout << "администратор" << endl;  }  }  // Запись данных пользователя в двоичный файл  void write\_user\_binar(fstream &fout, user &s) {  fout.write((char\*)&s, sizeof(s));  }  int main() {  setlocale(LC\_ALL, "");  ifstream fin("in.txt");  cout << "Программа сортирует данные пользователей по их роли." << endl << endl;  if (!fin.is\_open()) {  cout << "Не удалось открыть файл!" << endl;  return -1;  }  char files\_name[255];  cout << "Введите название файла и расширение: ";  cin.getline(files\_name, 255);  fstream fout(files\_name, ios::binary | ios::in | ios::out | ios::trunc);  cout << "Список пользователей: ";  vector<user> buf;  int n;  fout.seekp(0, ios\_base::beg);  for (n = 0; !fin.eof(); ++n) {  fill\_user(fin, buf);  write\_user\_binar(fout, buf.back());  }  cout << endl;  user buffer;  fout.seekp(0, ios\_base::beg);  for (int i = 0; i < n; i++) {  fout.read((char \*)&buffer, sizeof(buffer));  print\_user\_console(&buffer);  }  sort(buf, n);    cout << endl << "Список пользователей после сортировки: " << endl;  fout.seekp(0, ios\_base::beg);    for (int i = 0; i < n; i++) {  fout.write((char \*)&buf[i], sizeof(buf[i]));  }  fout.seekg(0, ios\_base::beg);  for (int i = 0; i < n; i++) {  fout.read((char \*)&buffer, sizeof(buffer));  print\_user\_console(&buffer);  }  cout << endl << "Пришло время проверить файл!" << endl;  fout.close();  fin.close();  return 0;  } |

**5 Результат выполнения экспериментальной части работы.**

Результаты запуска программы с различными входными значениями приведены в таблице 1.

Таблица 1 – Работа программы на тестовых наборах данных

| Входные данные | Результат |
| --- | --- |
| Kotov kot ser admin  Ovs Ovs vladqrty moder  Ovsfds vlfsdad ovfss123 moder  Mashfsd Natfsda Smasdfsh admin  Mfash Mfata Smfsdfash user  Mash Nata Smash user  Котов Котов кот123 admin | Список пользователей:  Kotov kot ser администратор  Ovs Ovs vladqrty модератор  Ovsfds vlfsdad ovfss123 модератор  Mashfsd Natfsda Smasdfsh администратор  Mfash Mfata Smfsdfash пользователь  Mash Nata Smash пользователь  Котов Котов кот123 администратор  Список пользователей после сортировки:  Kotov kot ser администратор  Mashfsd Natfsda Smasdfsh администратор  Котов Котов кот123 администратор  Ovs Ovs vladqrty модератор  Ovsfds vlfsdad ovfss123 модератор  Mash Nata Smash пользователь  Mfash Mfata Smfsdfash пользователь |