Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ № 23**

Указатели на функции  
 Вариант № 2

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

подпись, дата

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

подпись, дата

Студент КИ18-09б, 031831293 \_\_\_\_\_\_\_\_ Овсянников В.А.

подпись, дата

Студент КИ18-09б, 031831050 \_\_\_\_\_\_\_\_ Машковская Н.В.

подпись, дата

Красноярск 2019

**1 Дополнительное упражнение № 1 (вариант 2)**

В двоичном файле хранится информация о пользователях: фамилия, имя, учетное имя (login), пароль, роль (администратор, модератор, простой пользователь), статус (активен, заблокирован). С клавиатуры пользователь вводит учетное имя (login) и пароль. В ответ пользователь получает соответствующее приветствие и если пользователь является администратором, то предлагаются в виде текстового меню следующие функциональные возможности: список активных модераторов, список всех пользователей в алфавитном порядке, список заблокированных пользователей.

**2 Цель работы**

Получить практические навыки использования указателей на функции

для организации текстового меню в интерфейсе программы, получить опыт командной работы.

**3 Графическая схема алгоритма**

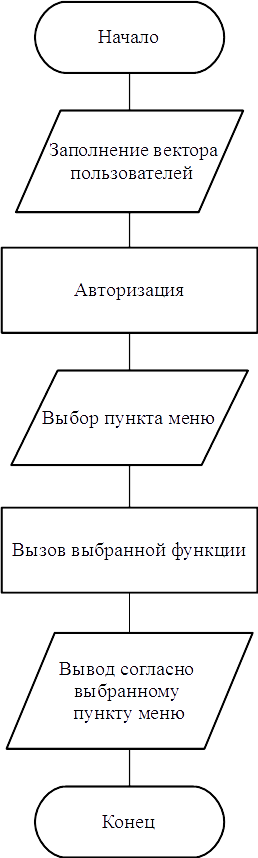


Рисунок 1 – Графическая схема алгоритма

**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  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157 | #include "pch.h"  #include <iostream>  #include <clocale>  #include <fstream>  #include <vector>  #include <iomanip>  #include <cstring>  using namespace std;  enum role\_user {  role\_users,  role\_moder,  role\_admin  };  struct user {  char user\_name[20];  char user\_surname[20];  char user\_log[20];  char user\_pass[20];  role\_user user\_role;  bool user\_status;  };  typedef void(\*menu)(const vector<user>&);  // Заполнение пользователя  void fill\_user(fstream& fout, vector<user>& users\_list) {  user buffer;  if (fout.read((char\*)& buffer, sizeof(buffer))) {  users\_list.push\_back(buffer);  }  }  // Вывод пользователя на экран  void print\_user(const user\* users\_list) {  cout << users\_list->user\_name << "\t" << users\_list->user\_surname << "\t" << users\_list->user\_log << "\t" << users\_list->user\_pass << "\t";  if (users\_list->user\_role == role\_users) {  cout << "Пользователь";  }  else if (users\_list->user\_role == role\_moder) {  cout << "Модератор";  }  else {  cout << "Администратор";  }  cout << "\t";  if (users\_list->user\_status) {  cout << "Активен" << endl;  }  else {  cout << "Заблокирован" << endl;  }  }  // Вывод активных модераторов  void active\_moder(const vector<user>& users\_list) {  cout << endl << "Список активных модераторов: " << endl;  cout.setf(ios::left);  for (int i = 0; i < users\_list.size(); i++) {  if (users\_list[i].user\_role == role\_moder) { //если роль - модератор  if (users\_list[i].user\_status) { //если модератор активен  print\_user(&users\_list[i]);  }  }  }  }  // Вывод заблокирвоанных пользователей  void block\_user(const vector<user>& users\_list) {  cout << endl << "Список заблокированных пользователей: " << endl;  cout.setf(ios::left);  for (int i = 0; i < users\_list.size(); i++) {  if (!users\_list[i].user\_status) { //если пользователь заблокирован  print\_user(&users\_list[i]);  }  }  }  // Сортировка пользователей по алфавиту  void sort\_aplh(const vector<user>& users\_list) {  vector<user> buf = users\_list;  cout << endl << "Список всех пользователей в алфавитном порядке: " << endl;  for (int i = 0; i < buf.size(); i++) {  for (int j = 0; j < buf.size(); j++) {  if (buf[i].user\_role > buf[j].user\_role) {  swap(buf[i], buf[j]);  }  }  }  for (int i = 0; i < buf.size(); i++) {  print\_user(&buf[i]);  }  }  // Авторизация пользователя  int auth(const vector<user>& users\_list, const char\* log, const char\* pass) {  int level\_access = -1; //значение уровня доступа  for (int i = 0; i < users\_list.size(); i++) {  if ((strcmp(users\_list[i].user\_log, log) == 0) && (strcmp(users\_list[i].user\_pass, pass) == 0)) {  level\_access = users\_list[i].user\_role; //значение уровня доступа  cout << "Здравствуйте, " << users\_list[i].user\_name << "!" << endl;  return level\_access;  }  }  cout << "Вы ввели неверные логин или пароль!" << endl;  return level\_access;  }  int main() {  setlocale(LC\_ALL, "");  fstream fout("inout.bin", ios::binary | ios::out | ios::in);  if (!fout.is\_open()) {  cout << "Входящий файл не найден! " << endl;  return -1;  }  vector<user> users\_list;  fout.seekg(0, ios\_base::beg);  while (!fout.eof()) {  fill\_user(fout, users\_list);  }  char log[20], pass[20];  cout << "Введите Ваш логин: ";  cin >> log;  cout << "Введите Ваш пароль: ";  cin >> pass;  int level\_access = auth(users\_list, log, pass); //авторизация  if (level\_access == 2) {  const vector<const char\*> main\_menu = { "Список активных модераторов", "Список всех пользователей в алфавитном порядке", "Список заблокированных пользователей" };  const vector<menu> items\_menu = { active\_moder, sort\_aplh, block\_user };  cout << endl << "Навигация по меню: " << endl;  for (int i = 0; i < main\_menu.size(); i++) {  cout << i + 1 << " — " << main\_menu[i] << endl;  }  int userChoice;  cout << "Выберите пункт меню: ";  cin >> userChoice;  if (userChoice >= 1 && userChoice <= items\_menu.size()) {  (\*items\_menu[userChoice - 1])(users\_list); //вызов выбранной функции  }  else {  cout << "Вы выбрали некорректный пункт меню. До свидания. Я отключаюсь. " << endl;  return 1;  }  }  else {  return 0;  }  return 0;  } |

**5 Результаты выполнения экспериментальной части работы**

Результаты запуска программы с различными входными значениями приведены ниже.

