**Лабораторная работа №7  
Двоичный файл: поиск и редактирование данных  
(4 ак.часа)**

**Задание:**

Создать тип данных (в виде структуры на языке Си) в соответствии с условием задачи своего варианта из нижеприведенного перечня, создав многомодульное консольное приложение для *MS Windows* на языке *C* или *C*++, которая в интерактивном режиме (меню) выполняет следующие команды пользователя:

а) из текстового файла ввести все данные и создать двоичный файл;

б) поиск в двоичном файле по условию задачи;

в) коррекция данных в двоичном файле (не при вводе из текстового, а именно редактирование двоичного файла);

г) просмотр двоичного файла – вывод данных из файла на экран в виде таблицы.

**Указание**. Для ввода строки из *нескольких* слов, используйте *fgets* вместо *fscanf* или *scanf*, например,

//---------------------------------------------------------------------------

#include <conio.h> // getch

#include <stdio.h> // FILE, fgets, fscanf, fflush

#include <string.h> // strchr, strlen

//---------------------------------------------------------------------------

int main()

{ // stdin – стандартный текстовый файл (клавиатура), иначе подставьте

// другое имя переменной (дескриптор своего открытого файла), например, f, если FILE\* f;

char Fam[31]; // сюда **введем одно слово** не длиннее 30 символов

printf("Fam =?");

**fscanf**(stdin, "%30s", Fam); // stdin – стандартный текстовый файл (клавиатура)

fflush(stdin); // если строка была длиннее, остаток удаляем из буфера,

// если его не удалить – перейдет в следующий ввод

printf("Vvedeno '%s'\n", Fam);

char FamIO[61]; // сюда **введем** **несколько слов** не длиннее 60 символов

char \*n; // позиция спец.символа (конца строки) в введенной строке

printf("FamIO =?");

**fgets**(FamIO,60,stdin); // stdin – стандартный текстовый файл (клавиатура), иначе другое имя

fflush(stdin); // если строка была длиннее, остаток удаляем из буфера, а если короче:

n = strchr(FamIO, '\n'); if (n) FamIO[n-FamIO]='\0'; // укорачиваем

printf("Vvedeno '%s'\n", FamIO);

printf("Press any key");

getch();

return 0;

}

**Перечень задач:**

1. *Исходные данные*: Багаж авиапассажира характеризуется фамилией пассажира, номером рейса, датой вылета, количеством вещей и общим весом вещей. *Условие поиска*: Пассажиры, багаж которых состоит из одной вещи весом не менее 30 кг, и вычислить средний вес багажа всех пассажиров. *Коррекция*: Вес и количество вещей неотрицательные и не более 50. При нулевом количестве вещей только нулевой вес.

2. *Исходные данные*: Расписание экзаменов содержит следующую информацию: номер студенческой группы, дата и время экзамена, номер аудитории, наименование дисциплины и фамилия преподавателя. *Условие поиска*: Даты, по которым занят конкретный преподаватель (Запросить фамилию в режиме диалога). *Коррекция*: Номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.

3. *Исходные данные*: Сведения о автомобиле состоят из его марки, номера, фамилии владельца и отметки о прохождении техосмотра. *Условие поиска*: Фамилии владельцев и номера автомобилей определенной марки (Запросить марку в режиме диалога), не прошедших вовремя техосмотр. *Коррекция*: Отметка о прохождении техосмотра включает и дату его прохождения. Техосмотр должен быть пройден не более чем год назад. Номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.

4. *Исходные данные*: В расписании поездов указаны: номер поезда, пункт отправления, пункт назначения, время отправления, время в пути, цена билета. *Условие поиска*: Поезда, следующие из Москвы в Санкт-Петербург, время отправления которых с 7.00 до 9.00. *Коррекция*: Время отправления: минуты 0-59, часы 0-23. Названия пунктов отправления и назначения могут состоят из нескольких слов.

5. *Исходные данные*: В анкетах сотрудников некоторого учреждения имеются следующие сведения: фамилия, имя, отчество, год рождения, должность (мастер, инженер, рабочий и т..д.), стаж, образование. *Условие поиска*: Мастера с высшим образованием, и вычислить их средний стаж. *Коррекция*: Стаж 0-60 лет, и не больше возраста-15лет.

6. *Исходные данные*: В сведениях об игрушках указывается название (кукла, кубики, мяч, конструктор и т.д.), цена, возрастные границы ребенка (например, от двух до пяти лет). *Условие поиска*: Игрушки, цена которых не превышает заданную (в режиме диалога) и которые подходят детям пяти лет. *Коррекция*: Возрастные границы: нижняя меньше верхней не менее чем на 2 года, не отрицательные и не более 150 лет.

7. *Исходные данные*: Имеются следующие сведения о книгах: фамилия и инициалы автора, название, год издания. *Условие поиска*: Книги, изданные с 1990 г., в названии которых содержится слово «информатика» в любом регистре. *Коррекция*: Год издания не менее 1800, и не более нынешнего года.

8. *Исходные данные*: О каждой книге имеются следующие сведения: фамилия и инициалы автора, название, год издания и количество запросов за последние три года. *Условие поиска*: Книги, изданные до 1980 года, по которым нет запросов *Коррекция*: Год издания не менее 1800, и не более нынешнего года. Запросов неотрицательное количество.

9. *Исходные данные*: Сведения состоят из фамилий, имен и отчеств сотрудников и их телефонов. *Условие поиска*: Сотрудники, номера телефонов которых начинаются с цифр 361, 362 или 273. *Коррекция*: Номера телефонов только из цифр, и их не менее 6.

10. *Исходные данные*: Имеются сведения о химических веществах: название, удельный вес, проводимость (проводник, полупроводник, изолятор), количество. *Условие поиска*: Полупроводники, имеющиеся в количестве более 5 кг. *Коррекция*: Проводимость только трех указанных видов, количество положительное.

11. *Исходные данные*: Сведения о компьютерах некоторого учрежден ия состоят из марки компьютера, даты приобретения, номера комнаты, где находится компьютер, и фамилии ответственного лица. *Условие поиска*: Компьютеры IBM HC-486, приобретенные в 1995 г. *Коррекция*: В дате номер месяца 1-12, дней не более 28-31 в соответствии с месяцем, номер комнаты положительный.

12. *Исходные данные*: Сведения об изделиях, хранящихся на складе, состоят из порядкового номера, наименования и количества. *Условие поиска*: Изделия, которых меньше заданного (в режиме диалога) количестве. *Коррекция*: Количество, имеющееся в наличии и заданное, - положительные величины.

13. *Исходные данные*: Имеются сведения об экспортируемых товарах: наименование, страна-экспортер, срок поставки и количество товара. *Условие поиска*: Товары, срок поставки которых - до 01 июля текущего года. *Коррекция*: Номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.

14. *Исходные данные*: Сведения о заказах некоторого предприятия состоят из номера заказа и даты (число, месяц, год) его исполнения. *Условие поиска*: Все весенние заказы. *Коррекция*: Номер заказа положительный, номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.

15. *Исходные данные*: Некоторое обслуживающее предприятие хранит сведения о заказах: наименование заказа и время (часы, минуты, секунды) его поступления. *Условие поиска*: Заказы, которые произошли с 15ч 20 мин 35 с до 18ч 50 мин 10с. *Коррекция*: Время поступления: часы 0-23, минуты и секунды 0-59.

16. *Исходные данные*: Сведения о заказе состоят из адреса (улица, дом, квартира) и даты (число, месяц, год) исполнения. *Условие поиска*: Заказы, которые должны были быть выполнены 15 мая 2015 г на Московской улице. *Коррекция*: Номер месяца 1-12, дней не более 28-31 в соответствии с месяцем. Номер квартиры положительное число.

17. *Исходные данные*: Сведения о горных вершинах состоят из следующих данных: название, координаты (широта и долгота), высота (в метрах над уровнем моря). *Условие поиска*: Вершины с высотой более 3000м, расположенные в Северном полушарии. *Коррекция*: Широта от -90 до 90, долгота от -180 до 180. Высота положительная.

18. *Исходные данные*: По некоторой группе людей собраны следующие сведения данные: возраст, пол, рост, вес, температура, артериальное давление (верхнее и нижнее). *Условие поиска*: Мужчины моложе 25 лет, имеющих верхнее артериальное давление больше 140. *Коррекция*: Пол только мужской и женский, некорректный сделать мужским. Давление положительное.

19. *Исходные данные*: О детях, поступающих в спортивную секцию, собраны следующие данные: фамилия, имя, год рождения, вес, рост. *Условие поиска*: Дети от 10 до 12 лет, рост которых не менее 155см, а вес не более 45 кг. *Коррекция*: Все дети не младше 5 лет и не старше 14 по году рождения, рост-вес положительные и не более 200.

20. *Исходные данные*: Электронная записная книжка должна содержать следующие сведения: фамилия, имя, отчество, адрес (город, улица, дом, квартира), телефон. *Условие поиска*: Всех, проживающих в г.Саратове на улице Солнечной. *Коррекция*: Дом, квартира - положительные числа.

21. *Исходные данные*: Имеются сведения о районных библиотеках города: номер, тип (взрослая, детская), район, адрес, часы работы. *Условие поиска*: Детские библиотеки заданного (в режиме диалога) районе. *Коррекция*: Часы работы: минуты 0-59, часы 6-21. Начало работы раньше завершения не менее чем на 4 часа.

22. *Исходные данные*: Сведения о магазинах города состоят из названия, типа (книжный, продуктовый, универсам и т.д.), района, адреса и часов работы. *Условие поиска*: Книжные магазины заданного (в режиме диалога) района, работающие после 18 часов, и число таких магазинов. *Коррекция*: Начало работы раньше завершения не менее чем на 4 часа.. Часы работы: минуты 0-59, часы 0-23.

23. *Исходные данные*: Имеются следующие данные о товарах обувного магазина: наименование (ботинки, сапоги, босоножки и т.д.), тип (мужская, женская или детская), сезон, цвет, размеры, цена. *Условие поиска*: Летняя детская обувь и ее средняя цена. *Коррекция*: Размеры с 10 до 45, типов только три.

24. *Исходные данные*: Имеются сведения о кошках, участвующих в конкурсе: порода, кличка, окрас, возраст, Фамилия хозяина, место в конкурсе. *Условие поиска*: Кошки сиамской породы, вошедшие в призовую десятку, и их средний возраст. *Коррекция*: Возраст от 0 до 15, место положительное.

25. *Исходные данные*: Сведения о маршрутах городских автобусов содержат: номер маршрута, начальный пункт, конечный пункт, описание маршрута (промежуточные остановки), среднее время движения по маршруту, интервал в час пик. *Условие поиска*: Маршруты с заданным (в режиме диалога) конечным пунктом и интервалом в часы пик, меньшим 10 мин, и число таких маршрутов. *Коррекция*: Интервалы положительные и не более часа, среднее время не более двух часов и не менее 30 минут.

26. *Исходные данные*: Имеются сведения о пациентах поликлиники: фамилия, имя, отчество, год рождения, номер полиса ОМС, адрес, основное заболевание, дата последнего посещения лечащего врача. *Условие поиска*: Больные диабетом, не посещавшие лечащего врача более трех месяцев, и общее число больных диабетом. *Коррекция*: В дате посещения номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.

27. *Исходные данные*: Имеются сведения о прохождении детьми прививок: фамилия, имя, год рождения, адрес, отметки о прививках против туберкулеза, полиомиелита, кори, коклюша, столбняка. *Условие поиска*: Деи семи лет, у которых отсутствует хотя бы одна прививка. *Коррекция*: В дате рождения номер месяца 1-12, дней не более 28-31 в соответствии с месяцем. Отметки только двух видов: сделана по плану или не сделана.

28. *Исходные данные*: Сведения о товарах магазина одежды: наименование (костюм, пальто и т.д.), размер, рост, цвет, номер модели. *Условие поиска*: Костюмы, начиная с 52 размера. *Коррекция*: Размеры с 20 по 60, рост 100-200.

29. *Исходные данные*: Имеются сведения о выпускниках учебного заведения: фамилия, имя, отчество, год поступления, год окончания, специальность, отметка о работе по специальности. *Условие поиска*: Выпускники, окончившие учебное заведение в течение последних трёх лет и не работающие по специальности. *Коррекция*: Год окончания больше года поступления не менее чем на 4 года, но не больше нынешнего.

30. *Исходные данные*: Имеются сведения о программных продуктах: наименование, автор, соавторы, версия, год создания. *Условие поиска*: Вывести сведения об авторах, создавших программный продукт без соавторов в заданном (в режиме диалога) году. *Коррекция*: Год создания и заданный год не раньше 1950 и не больше нынешнего.

**Пример решения задачи** на двух языках приведен и рассмотрен на лекции.

**Контрольные вопросы**

1. Можно ли считать структуру/запись целиком из текстового файла? Двоичного? Типизированного?
2. Как указать тип файла? Имя файла? //Типизированного в c++ нет
3. Как указать длину записи в нетекстовом файле? //sizeof(“запись”)
4. Как узнать удачей ли закончилась попытка чтения или записи в двоичный/бестиповой файл? //В поток выдается ошибка, которую можно считать функцией ferror
5. С помощью какого оператора выполняется чтение из двоичного/бестипового файла? fread
6. С помощью какого оператора выполняется запись в двоичный/бестиповой файл? fwrite
7. Как узнать текущую позицию в файле? Как изменить текущую позицию в нетекстовом файле?

|  |  |
| --- | --- |
| С/С++ (двоичный файл) | Аналог на Delphi (бестиповой файл) |
| #include <windows.h>  #include <stdio.h>  #include <conio.h>  //------------------------------------------------------------  struct TStud {  struct {  char F[31], I[31], O[31];  } FIO;  int o1,o2,o3;  };  // ---------создание копии строки, но с символами  //---------------в верхнем регистре в Win1251  char \*ANSIUpperCase(char \*s, char \*S){  S = strcpy(S, s);  char \*ch=S;  while (\*ch) {  if (\*ch>='a' && \*ch<='z' || \*ch>='а' && \*ch<='я') \*ch= \*ch-32;  if (\*ch =='ё') \*ch='Ё';  ch++;  }  return S;  }  //----------первая часть: создание двоичного из текстового  void CreateBinaryFile(int argc, char\* argv[]){  if (argc<3) {  printf("Мало параметров\nPress any key");  getch();  return;  }  FILE \*ft = fopen(argv[2], "rt");  if (ft==NULL) {  printf("Error: не удалось открыть файл с \  исходными данными %s\n", argv[2]);  printf("Press any key "); getch();  return;  }  FILE \*fb = fopen(argv[1], "wb");  if (fb==NULL) {  fclose(ft);  printf("Error: не удалось создать \  двоичный файл %s\n", argv[1]);  printf("Press any key"); getch();  return;  }  TStud Stud; // либо struct TStud Stud; в С  int kol=0, nw=1;  char s[3];  while (nw){  fscanf(ft,"%30s", Stud.FIO.F); // ввод слова  //признак конца файла - фамилия \*\* или конец файла  if (strcmp(Stud.FIO.F,"\*\*")==0 || feof(ft)) break;  //для ввода строки из нескольких слов см.Указание выше  fscanf(ft,"%30s", Stud.FIO.I);  fscanf(ft,"%30s", Stud.FIO.O);  fscanf(ft,"%s", s); Stud.o1=atoi(s);  fscanf(ft,"%s", s); Stud.o2=atoi(s);  fscanf(ft,"%s", s); Stud.o3=atoi(s);  nw=fwrite(&Stud, sizeof(Stud), 1, fb);  kol++;  }  if (nw!=1) printf("Error: Ошибка при записи");  fclose(ft);  fclose(fb);  printf("Создан двоичный файл из %d записей \  по %d байт\n", kol, sizeof(TStud));  printf("Press any key to continue");  getch();  return;  }  //-------------- вторая часть: поиск в двоичном файле ----  void FindIVAN(int argc, char\* argv[]){  if (argc<2) {  printf("Мало параметров\nPress any key");  getch();  return;  }  FILE \*fb = fopen(argv[1], "rb");  if (fb==NULL) {  printf("Error: не удалось открыть двоичный \  файл %s\n", argv[1]);  printf("Press ENTER");  getch();  return;  }  TStud Stud;  char SUp[31];  int kol=0, nr=1;  while (nr){  nr=fread(&Stud, sizeof(Stud), 1, fb);  if (nr>0)  if (Stud.o1==5 && Stud.o2==5 && Stud.o3==5 &&  strcmp(ANSIUpperCase(Stud.FIO.I, SUp),"ИВАН")==0) {  printf("Найден %s %s %s\n", Stud.FIO.I, Stud.FIO.O, Stud.FIO.F);  kol++;  }  }  if (kol==0)  printf("Данные, соответствующие запросу,не найдены\n");  else  printf("Всего найдено: %d\n", kol);  fclose(fb);  printf("Press any key to continue");  getch();  return;  }  //-------третья часть: корректировка в двоичном файле ----  void CorrectFile(int argc, char\* argv[]){  if (argc<2) {  printf("Мало параметров\nPress any key");  getch();  return;  }  FILE \*fb = fopen(argv[1], "rb+");  if (fb==NULL) {  printf("Error: не удалось открыть двоичный файл %s\n", argv[1]);  printf("Press ENTER");  getch();  return;  }  TStud Stud;  int kol=0, nr=1, nw=1, flag;  nr=fread(&Stud, sizeof(Stud), 1, fb);  while (nr==1 && nw==1){  printf("Считана запись:\n");  printf("%25s%20s%25s%2d%2d%2d\n", Stud.FIO.F,  Stud.FIO.I, Stud.FIO.O, Stud.o1,Stud.o2,Stud.o3);  flag=0;  if (Stud.o1<2) {Stud.o1=2; flag=1;}  if (Stud.o2<2) {Stud.o2=2; flag=1;}  if (Stud.o3<2) {Stud.o3=2; flag=1;}  if (Stud.o1>5) {Stud.o1=5; flag=1;}  if (Stud.o2>5) {Stud.o2=5; flag=1;}  if (Stud.o3>5) {Stud.o3=5; flag=1;}  if (flag) {  kol++;  fseek(fb,0-sizeof(Stud),SEEK\_CUR);  nw=fwrite(&Stud, sizeof(Stud), 1, fb);  fseek(fb,0,SEEK\_CUR);  printf("Сделана корректировка:\n");  printf("%25s%20s%25s%2d%2d%2d\n", Stud.FIO.F,  Stud.FIO.I,Stud.FIO.O,Stud.o1,Stud.o2, Stud.o3);  }  nr=fread(&Stud, sizeof(Stud), 1, fb);  }  if (kol==0)  printf("Ни одной корректировки\n");  else  printf("Всего корректировок: %d\n", kol);  fclose(fb);  printf("Press any key to continue");  getch();  return;  }  //------------------главная функция-----------------------  int main(int argc, char\* argv[])  {  char ch;  SetConsoleOutputCP(1251);  do{  printf("\nN - создать новый тип.файл; F - сразу поиск;\  C - коррекция; E - конец.\nВаш выбор?");  ch=getchar(); fflush(stdin);  ch=toupper(ch);  switch (ch) {  //----------первая часть: создание двоичного из текстового  case 'N': CreateBinaryFile(argc, argv); break;  //-------------- вторая часть: поиск в двоичном файле ----  case 'F': FindIVAN(argc, argv); break;  //-------третья часть: корректировка в двоичном файле ----  case 'C': CorrectFile(argc, argv); break;  //-----------выход----------------------------------------  case 'E': return 0;  //--------------------------------------------------------  default:  printf("Нет такой команды\nPress any key");  getch();  }  } while (ch!='E');  return 0;  }  //----------------------------------------------------------- | program Project1;  {$APPTYPE CONSOLE}  uses  SysUtils, Windows;  //-----------------------------------------------------------  Type  TStud = record  FIO: record  F, I, O: string[30];  end;  o1,o2,o3: byte;  end;  //----------первая часть: создание двоичного из текстового  Procedure CreateBinaryFile();  var  Stud: TStud;  ft: TextFile;  fb: file; // бестиповой (двоичный) файл  kol,nw: integer;  begin  if ParamCount<2 then  begin  writeln('Мало параметров '#13#10'Press enter');  readln; exit  end;  AssignFile(ft, ParamStr(2));  {$I-} Reset(ft); {$I+}  if IOResult<>0 then  begin  writeln('Error: не удалось открыть файл ',  'с исходными данными ',ParamStr(2));  write('Press ENTER'); readln; exit  end;  AssignFile(fb, ParamStr(1));  {$I-} ReWrite(fb,sizeof(TStud));{$I+}  // или при открытии 1, а при вводе,выводе SizeOf  if IOResult<>0 then  begin  CloseFile(ft);  writeln('Error: не удалось создать двоичный ',  'файл ',ParamStr(1));  write('Press ENTER'); readln; exit  end;  kol:=0; nw:=1;  while not seekeof(ft) and (nw=1) do  begin  readln(ft, Stud.FIO.F);  //признак конца файла - фамилия \*\* или конец файла  if Stud.FIO.F = '\*\*' then break;  readln(ft, Stud.FIO.I);  readln(ft, Stud.FIO.O);  readln(ft, Stud.o1,Stud.o2,Stud.o3);  BlockWrite(fb, Stud, 1, nw);  inc(kol)  end;  if nw<>1 then writeln('Error: Ошибка при записи');  CloseFile(ft);  CloseFile(fb);  writeln('Создан двоичный файл из ',kol,  ' записей по ',sizeof(TStud),' байт');  write('Press ENTER'); readln;  end;  //-------------- вторая часть: поиск в двоичном файле ----  Procedure FindIVAN();  var  Stud: TStud;  fb: file;  kol, nr: integer;  begin  if ParamCount<1 then  begin  writeln('Мало параметров '#13#10'Press enter');  readln; exit  end;  AssignFile(fb, ParamStr(1));  {$I-} ReSet(fb, sizeof(TStud));{$I+}  // или здесь при открытии 1, а при вводе,выводе SizeOf  if IOResult<>0 then  begin  writeln('Error: не удалось открыть двоичный файл ', ParamStr(1));  write('Press ENTER'); readln; exit  end;  kol:=0; nr:=1;  while not eof(fb) and (nr=1) do  begin  BlockRead(fb, Stud, 1, nr);  if nr>0 then  if (Stud.o1=5) and (Stud.o2=5) and (Stud.o3=5) and  (ANSIUpperCase(Stud.FIO.I) = 'ИВАН') then  begin  writeln('Найден ', Stud.FIO.I,' ',Stud.FIO.O,' ', Stud.FIO.F);  inc(kol)  end;  end;  if nr<>1 then writeln('Error: ошибка при считывании');  if kol=0 then  writeln('Данные, соответствующие запросу, не найдены')  else  writeln('Всего найдено: ',kol);  CloseFile(fb);  write('Press ENTER'); readln;  end;  //-------третья часть: корректировка в двоичном файле ----  Procedure CorrectFile();  var  Stud: TStud; fb: file; flag: boolean;  kol, nr,nw,so: integer;  begin  if ParamCount<1 then  begin  writeln('Мало параметров '#13#10'Press enter');  readln; exit  end;  AssignFile(fb, ParamStr(1));  {$I-} ReSet(fb, 1);{$I+}  // или как выше здесь sizeof, а при чтении/записи 1  if IOResult<>0 then  begin  writeln('Error: не удалось открыть типизированный файл ', ParamStr(1));  write('Press ENTER'); readln; exit  end;  kol:=0; so:=sizeof(TStud); nr:=so; nw:=so;  while not eof(fb) and (nr=so) and (nw=so) do  begin  BlockRead(fb, Stud, so, nr);  writeln('Считана запись:');  writeln(Stud.FIO.F:25, Stud.FIO.I:20, Stud.FIO.O:25, Stud.o1:2,Stud.o2:2,Stud.o3:2);  flag:=false;  if (Stud.o1<2) then begin Stud.o1:=2; flag:=true end;  if (Stud.o2<2) then begin Stud.o2:=2; flag:=true end;  if (Stud.o3<2) then begin Stud.o3:=2; flag:=true end;  if (Stud.o1>5) then begin Stud.o1:=5; flag:=true end;  if (Stud.o2>5) then begin Stud.o2:=5; flag:=true end;  if (Stud.o3>5) then begin Stud.o3:=5; flag:=true end;  if flag then  begin  inc(kol);  seek(fb, FilePos(fb)-so);  BlockWrite(fb, Stud, nr, nw);  writeln('Сделана корректировка:');  writeln(Stud.FIO.F:25, Stud.FIO.I:20, Stud.FIO.O:25, Stud.o1:2,Stud.o2:2,Stud.o3:2);  end;  end;  if kol=0 then  writeln('Ни одной корректировки')  else  writeln('Всего корректировок: ',kol);  CloseFile(fb);  write('Press ENTER'); readln;  end;  //--------------------главная программа--------------------  var  ch: char;  begin  SetConsoleOutputCP(1251);  repeat  write('N - создать новый тип.файл; F - сразу поиск;',  ' C - коррекция; E - конец.',  #13#10'Ваш выбор?');  readln(ch);  ch:= UpCase(ch);  case ch of  //----------первая часть: создание двоичного из текстового  'N': CreateBinaryFile;  //----------вторая часть: поиск в двоичном файле ---------  'F': FindIVAN;  //----------третья часть: корректировка в двоичном файле -  'C': CorrectFile;  //-----------выход----------------------------------------  'E': exit;  //--------------------------------------------------------  else  begin  writeln('Нет такой команды');  write('Press ENTER'); readln;  end;  end;  until ch='E';  end. |

Со следующей страницы аналогичный код на **Delphi с типизированным файлом**:

|  |
| --- |
| Аналог на Delphi с типизированным файлом |
| program Project1;  {$APPTYPE CONSOLE}  uses  SysUtils, Windows;  //--------------------------------------------------------------  Type  TStud = record  FIO: record  F, I, O: string[30];  end;  o1,o2,o3: byte;  end;  //----------первая часть: создание типизированного из текстового  Procedure CreateTypedFile();  var  Stud: TStud;  ft: TextFile;  fb: file of TStud; // типизированный файл – файл записей  kol: integer;  begin  if ParamCount<2 then  begin  writeln('Мало параметров '#13#10'Press enter');  readln; exit  end;  AssignFile(ft, ParamStr(2));  {$I-} Reset(ft); {$I+}  if IOResult<>0 then  begin  writeln('Error: не удалось открыть файл с исходными данными ',ParamStr(2));  write('Press ENTER'); readln; exit  end;  AssignFile(fb, ParamStr(1));  {$I-} ReWrite(fb);{$I+}  if IOResult<>0 then  begin  CloseFile(ft);  writeln('Error: не удалось создать типизированный файл ',ParamStr(1));  write('Press ENTER'); readln; exit  end;  kol:=0;  while not seekeof(ft) do  begin  readln(ft, Stud.FIO.F);  readln(ft, Stud.FIO.I);  readln(ft, Stud.FIO.O);  readln(ft, Stud.o1,Stud.o2,Stud.o3);  write(fb, Stud);  inc(kol)  end;  CloseFile(ft);  CloseFile(fb);  writeln('Создан типизированный файл из ',kol,' записей');  write('Press ENTER');  readln;  end;  //--------------------------------------------------------------  //-------------- вторая часть: поиск в типизированном файле ----  Procedure FindIVAN();  var  Stud: TStud;  fb: file of TStud;  kol: integer;  begin  if ParamCount<1 then  begin  writeln('Мало параметров '#13#10'Press enter');  readln; exit  end;  AssignFile(fb, ParamStr(1));  {$I-} ReSet(fb);{$I+}  if IOResult<>0 then  begin  writeln('Error: не удалось открыть типизированный файл ',ParamStr(1));  write('Press ENTER'); readln; exit  end;  kol:=0;  while not eof(fb) do  begin  read(fb, Stud);  if (Stud.o1=5) and (Stud.o2=5) and (Stud.o3=5) and  (ANSIUpperCase(Stud.FIO.I) = 'ИВАН') then  begin  writeln('Найден ', Stud.FIO.I,' ',Stud.FIO.O,' ',Stud.FIO.F);  inc(kol)  end;  end;  if kol=0 then  writeln('Данные, соответствующие запросу, не найдены')  else  writeln('Всего найдено: ',kol);  CloseFile(fb);  write('Press ENTER'); readln;  end;  //--------------------------------------------------------------  //-------третья часть: корректировка в типизированном файле ----  Procedure CorrectFile();  var  Stud: TStud;  fb: file of TStud;  kol: integer;  flag: boolean;  begin  AssignFile(fb, ParamStr(1));  {$I-} ReSet(fb);{$I+}  if IOResult<>0 then  begin  writeln('Error: не удалось открыть типизированный файл ',ParamStr(1));  write('Press ENTER'); readln;  exit  end;  kol:=0;  while not eof(fb) do  begin  read(fb, Stud);  writeln('Считана запись:');  writeln(Stud.FIO.F:25, Stud.FIO.I:20, Stud.FIO.O:25,  Stud.o1:2,Stud.o2:2,Stud.o3:2);  flag:=false;  if (Stud.o1<2) then begin Stud.o1:=2; flag:=true end;  if (Stud.o2<2) then begin Stud.o2:=2; flag:=true end;  if (Stud.o3<2) then begin Stud.o3:=2; flag:=true end;  if (Stud.o1>5) then begin Stud.o1:=5; flag:=true end;  if (Stud.o2>5) then begin Stud.o2:=5; flag:=true end;  if (Stud.o3>5) then begin Stud.o3:=5; flag:=true end;  if flag then  begin  inc(kol);  seek(fb, FilePos(fb)-1);  write(fb, Stud);  writeln('Сделана корректировка:');  writeln(Stud.FIO.F:25, Stud.FIO.I:20, Stud.FIO.O:25,  Stud.o1:2,Stud.o2:2,Stud.o3:2);  end;  end;  if kol=0 then  writeln('Ни одной корректировки')  else  writeln('Всего корректировок: ',kol);  CloseFile(fb);  write('Press ENTER'); readln;  end;  //--------------------главная программа-------------------------  var  ch: char;  begin  SetConsoleOutputCP(1251);  repeat  write('N - создать новый тип.файл; F - сразу поиск; C - коррекция; E - конец.',  #13#10'Ваш выбор?');  readln(ch);  ch:= UpCase(ch);  case ch of  //--------------------------------------------------------------  //----------первая часть: создание типизированного из текстового  'N': CreateTypedFile;  //----------вторая часть: поиск в типизированном файле ---------  'F': FindIVAN;  //----------третья часть: корректировка в типизированном файле -  'C': CorrectFile;  //-----------выход----------------------------------------------  'E': exit;  //--------------------------------------------------------------  else  begin  writeln('Нет такой команды');  write('Press ENTER'); readln;  end;  end;  until ch='E';  end. |