

Лекция 9. Двоичный (бинарный) файл, структуры и объединения

Напоминаю, что в языке Си есть два вида файлов: текстовый и бинарные. Файловая переменная – дескриптор файла – для любого из этих двух типов описывается с помощью типа *FILE* из модуля *stdio.h*, который необходимо подключить к своей программе.

До конца семестра для работы с файлами использовать только тип *FILE*, а не классы (*fstream*, *ifstream*, *ofstream*). А также не использовать классы *string*, *list*, *stack* и т.д. и вообще не использовать классы – не свои, не чужие; по плану дисциплины – изучаем процедурное программирование на двух языках высокого уровня. Объектно-ориентированное программирование в следующем семестре.

Порядок работы с бинарными файлами:

Способ	Действие
	Создание нового потока ввода/вывода и определения дескриптора файла. Можно использовать несколько потоков.
Требуется подключить модуль	<code>#include <stdio.h></code>
Описание переменной	<code>FILE *fb;</code>
Открытие существующего для чтения (rb) или редактирования (r+b) * **	<code>fb = fopen("File1.dat", "r+b");</code>
Открытие для записи с созданием (wb или плюс чтение w+b) * **	<code>fb = fopen("d:\\File1.dat", "w+b");</code>
Чтение данных по размеру порции данных и количеству таких порций (применима и к текстовому)	<code>int nr, chislo=255, mas[3]={48,49,50}; nr=fread(&chislo, sizeof(int), 1, fb); nr=fread(mas, sizeof(int), 3, fb);</code>
Запись данных по размеру порции данных и количеству таких порций. nw – сколько реально записалось (применима и к текстовому)	<code>int nw, chislo, mas[10]; nw=fwrite(&chislo, sizeof(int), 1, fb); nw=fwrite(mas, sizeof(int), 10, fb);</code>
Запись строк (до символа '\0') (многие функции для чтения/записи строк и символов из/в текст.файл применимы, не применимы: <code>fscanf</code> , <code>fprintf</code> с форматированием и изменением типа и систем счисления)	<code>fputs("Строка", fb);</code>
Узнать текущую позицию в файле	<code>ftell(fb)</code>
Перейти в указанную позицию	<code>fseek(fb, sizeof(int), SEEK_SET);</code>
От начала файла	<code>fseek(fb, sizeof(int), SEEK_END);</code>
От конца файла	<code>fseek(fb, 0-sizeof(int), SEEK_CUR);</code>
От текущей позиции	<code>fseek(fb, 0-sizeof(int), SEEK_CUR);</code>
Закрытие файла	<code>fclose(f)</code>
Дополнительные функции	if (feof(f)) конец файла, <code>rename</code> , <code>remove</code> , ...
Узнать номер файла Изменить размер файла (обрезать) Закрыть по номеру (полный пример ниже)	<code>#include <io.h> #include <fcntl.h> #include <sys\stat.h> void main(){ int hndl hndl = open("test1.txt", O_RDWR, S_IREAD S_IWRITE); chsize(hndl, 3*sizeof(int)); close(hndl); return; }</code>

* Букву *b* (бинарный/двоичный) надо указывать обязательно, по умолчанию – текстовый.

**(*rb* или *r+b* и *wb* или *w+b*). *t* – *text*, *b* – *binary*, *r* – *read*, *w* – *create+write*, *r+* – *read+write*, *w+* – *create+write+read*. *a* – тоже можно, но не удобно, т.к. нумерация будет не с начала файла.

** *Относительный путь* к файлу указывается от расположения *exe*-файла. Например, в папке **Debug**, **Release** или папке проекта в зависимости от версии среды программирования и режима компиляции.

Слеш (\) в имени файла удваивается, так как является служебным символом в строке, например, `\n |t|0|\" ||`

В качестве двоичного можно открыть и текстовый файл, чтобы его отредактировать, например,

Создайте текстовый файл с фразой:

```
This is an apple.
```

И именем `example.txt`.

Создайте консольное приложение с кодом:

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    FILE * pFile;
    pFile = fopen ( "example.txt" , "r+b" );
    if (!pFile) { printf("file example.txt not found"); getch();
return;}
    fputs ( "This is an apple." , pFile );
    fseek ( pFile , 9 , SEEK_SET );
    fputs ( " sam" , pFile );
    fclose ( pFile );
    return;
}
```

После сохранения программы и файла, успешной компиляции и выполнения этого кода файл `example.txt` будет содержать текст:

```
This is a sample.
```

Поскольку поверх четырёх символов, начиная с 10-го, будут записаны символы « `sam`».

Пример создания и обрезки файла в *C* (в *Delphi* обрезать файл – *truncate*) (может понадобится для типового расчета)

```
#include <conio.h>
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <stdio.h>
void main(){

FILE *f;
f=fopen("fortrunc.txt", "w+b"); //создаем и наполняем файл
int i;
for (i=48; i<58; i++)
    fwrite(&i, sizeof(i), 1, f);
int size=ftell(f); // узнаем размер по текущей позиции
```

```

printf("size = %d\n", size);
fclose(f);

int hndl; // номер (handle) файла узнаем при открытии:
hndl = open("fortrunc.txt", O_RDWR, S_IREAD|S_IWRITE);
chsize(hndl, 3*sizeof(int)); // меняем размер
close(hndl); // закрываем по номеру (handle)

f=fopen("fortrunc.txt", "rb"); // опять открываем и смотрим размер
fseek(f, 0, SEEK_END);
size=ftell(f);
printf("size = %d\n", size);
fclose(f);

getch();
return;
}

```

Структуры и объединения

Структура в C (вместе со словом <i>struct</i>)	Запись аналогичная в <i>Delphi</i>
<pre> typedef struct { int a,b; double c; } TMyRec; struct TMyRec MyRec, *PMyRec; </pre>	<pre> Type TMyRec = record a, b: integer; c: double; end; </pre>
Структура в C++	
<pre> struct TMyRec { int a,b; double c; }; TMyRec MyRec, *PMyRec; </pre>	<pre> Var MyRec: TMyRec; PMyRec: ^TMyRec; // объявлены запись и указатель на запись </pre>
Обращение к полям на C	Обращение к полям на <i>Delphi</i>
<pre> MyRec.a = 1; PMyRec = new TMyRec; // или =&MyRec; (*PMyRec).a = 2; PMyRec->a = 3; delete PMyRec; </pre>	<pre> MyRec.a := 1; New(PMyRec); // или PMyRec:=@MyRec; PMyRec[^].a = 2; Dispose(PMyRec); </pre>

Объединение в C/C++	Объединение в C++	Запись аналогичная в <i>Delphi</i>
<pre> typedef union { int a; double c; } TVarRec; union TVarRec VarRec; </pre>	<pre> union TVarRec { int a; double c; }; TVarRec VarRec; </pre>	<pre> Type TVarRec = record Case byte of 0: (a: integer); 1: (c: double); end; Var VarRec: TMyRec; // объявлена запись с вариантами </pre>
Обращение к полям на C		Обращение к полям на <i>Delphi</i>
<pre> VarRec.a=1; VarRec.c = M_PI;//Пи из <math.h> </pre>		<pre> VarRec.a := 1; VarRec.c = pi; </pre>
Выделяется память только для одного поля (a, c) (или набора полей) с наибольшим размером.		

Еще один пример *записи с вариантами* и *перечисляемый тип*:

Запись с вариантами в <i>Delphi</i>	Аналогичная структура и объединение в <i>C++</i> и <i>перечисляемый тип</i>
<pre>Type TSummer = (June, July, August); TSus = record Mon: TSummer; Case TSummer of June: (a, b: integer); July: (c: double); August: (d:byte; r:real) end; Var Perem: TSus; // объявлена запись с вариантами</pre>	<pre>enum TSummer = {June, July, August}; struct TSus{ TSummer Mon; union { struct { int a,b; } StJune; double c; struct { unsigned char d; double r;} StAug; } Un; } Perem;</pre>
Обращение к полям на <i>Delphi</i>	Обращение к полям на <i>C</i>
<pre>Perem.Mon := June; Perem.a := 5; Perem.c := 0.123456789</pre>	<pre>Perem.Mon = June; Perem.Un.StJune.a = 5; Perem.Un.c = 0.123456789;</pre>

Пример решения задачи из Лабораторной №7.

Условие: Имеются сведения о студентах: фамилия, имя, отчество и оценки за три экзамена (физика, мат.анализ и программирование). Найти всех юношей по имени Иван, получивших по всем этим трём экзаменам пятёрки. Предусмотреть коррекцию: оценки лежат в диапазоне 2-5.

Далее приведен **программный код** на Delphi(две версии: для бестипового (двоичного) и типизированного файла) и Си (двоичный файл), поскольку на экзамене будет возможность выбора языка для решения подобной задачи с коррекцией файла. Код на одном языке можно рассматривать как комментарий к другому. Кроме того, эти примеры рассматриваются на лекции.

Задание Лабораторной работы №7:

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:

- из **текстового файла** ввести все данные и создать двоичный файл;
- поиск в двоичном файле по условию задачи;
- коррекция данных в двоичном файле (не при вводе из текстового, а именно редактирование двоичного файла);
- просмотр двоичного файла – вывод данных из файла на экран в виде таблицы. – этой части нет в примере. **Добавьте сами пункт меню и функцию**

C/C++ (двоичный файл)	Аналог на Delphi (бестиповой файл)
<pre> #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) { </pre>	<pre> 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 ', 's исходными данными ',ParamStr(2)); write('Press ENTER'); readln; exit end; AssignFile(fb, ParamStr(1)); {\$I-} Rewrite(fb,sizeof(TStud));{\$I+} // или при открытии l, а при вводе,выводе SizeOf if IOResult<>0 then begin CloseFile(ft); </pre>

```

    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;
    }
    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+}

```

```

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) {

```

```

// 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;
end;

```

```

    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);

```

```

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')

```

```

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;\n
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: коррективровка 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;
}
//-----

```

```

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: коррективровка 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 (как в лабораторной работе №2):

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.

```