

Файлы

1. Типы файлов
2. Доступ к файлу
3. Текстовые файлы
4. Типизированный файлы
5. Нетипизированные файлы
6. Умная Йохансон

1 Типы файлов

Файл - любая именованная область внешней памяти

Характерные особенности:

- имя;
- содержит компоненты одного типа;
- длина файла не оговаривается при создании.

Связано ли расширение файла с его содержимым? - **НЕТ**

1 Типы файлов

Объявление файловой переменной:

- <имя> = FILE OF <тип>; (типизированный)
- <имя> = TEXT; (текстовый)
- <имя> = FILE. (нетипизированный)

type

```
product = record
```

```
  name: string;
```

```
  code: word;
```

```
  price: real;
```

```
end;
```

```
text80 = file of string[80];
```

var

```
f1: file of char;
```

```
f2: text;
```

```
f3: file;
```

```
f4: text80;
```

```
f5: file of product;
```

Файлы

1. Типы файлов
2. **Доступ к файлу**
3. Текстовые файлы
4. Типизированный файлы
5. Нетипизированные файлы
6. Умная Йохансон

2 Доступ к файлу

Файловая переменная связывается с именем файла:

ASSIGN (<ф.п.>, <имя файла>);

здесь <ф.п.> – файловая переменная (правильный идентификатор объявленный в программе как переменная файлового типа);
<имя файла> – текстовое выражение, содержащее имя файла.

```
const
  name='e:\work\try.txt';
var
  f: Text;
begin
  assign(f,name);
  ...
```

2 Доступ к файлу

После связывания файловой переменной с файлом процедура работы с файлом состоит из следующих этапов:

- инициация файла – указание направление передачи данных;
- чтение и/или запись информации; (*может повторяться сколько угодно раз*)
- закрытие файла.

Чтение файла:

```
RESET(<ф.п.>);
```

Перезапись файла:

```
REWRITE(<ф.п.>);
```

Добавление в файл:

```
APPEND(<ф.п.>);
```

Закрытие файла:

```
CLOSE (<ф.п.>);
```

2 Доступ к файлу

Специальные логические функции:

EOF(<ф.п.>): `boolean` – возвращает `TRUE` , если файловый указатель стоит в конце файла.

EOLN(<ф.п.>): `boolean` – возвращает `TRUE` , если файловый указатель стоит в конце строки

Чтение из текстового файла:

READ (<ф.п.>, <сп. ввода>)

READLN (<ф.п.>, <сп. ввода>)

Запись в текстовый файл:

WRITE (<ф.п.>, <сп. вывода>)

WRITELN (<ф.п.>, <сп. вывода>)

Файлы

1. Типы файлов
2. Доступ к файлу
3. **Текстовые файлы**
4. Типизированный файлы
5. Нетипизированные файлы
6. Умная Йохансон

3 Текстовый файл

Вывод на экран исходного текста программы

```
var
  f: text;
  s: string;
begin
  assign(f,'myprog.pas');
  reset(f);
  while not eof(f) do
    begin
      readln(f,s);
      writeln(s);
    end;
  close(f);
end.
```

3 Текстовый файл

Запись значений массива в файл

```
const N = 5;
var
  f: text;
  m: array[1..N] of real;
  i: integer;
begin
  assign(f,'prog.dat');
  rewrite(f);
  for i := 1 to N do
    begin
      m[i] := random * 10;
      writeln(m[i]); //ВЫВОД на экран
      write(f,m[i]); //ВЫВОД в файл
    end;
  close(f);
end.
```

На

экране

```
5.34014130259871
0.0751469703741125
9.14481708274447
2.01952238661215
6.81229438949949
```

3 Текстовый файл

Содержимое файла prog.dat (в виде текста)

5.340141302598710.07514697037411259.144817082744472.019522386612156.81229438949949

Содержимое файла prog.dat в виде бинарного файла

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	35	2E	33	34	30	31	34	31	33	30	32	35	39	38	37	31
00000010	30	2E	30	37	35	31	34	36	39	37	30	33	37	34	31	31
00000020	32	35	39	2E	31	34	34	38	31	37	30	38	32	37	34	34
00000030	34	37	32	2E	30	31	39	35	32	32	33	38	36	36	31	32
00000040	31	35	36	2E	38	31	32	32	39	34	33	38	39	34	39	39
00000050	34	39														

Таблица ASCII символов

	:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI	
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US	
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL	

3 Текстовый файл

Изменим вывод в файл, вместо `write(f,m[i])`

`fprintf(f,"%f\n",m[i])` файла `prog.dat` (в виде текста)

```
753859810882
9.44486666444916
5.96698050665063
6.08366832886062
3.03903472751334
```

Содержимое файла `prog.dat` в виде бинарного файла

offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	34	2E	37	32	37	35	33	38	35	39	38	31	30	38	38	32
00000010	0D	0A	39	2E	34	34	34	38	36	36	36	36	34	34	34	39
00000020	31	36	0D	0A	35	2E	39	36	36	39	38	30	35	30	36	36
00000030	35	30	36	33	0D	0A	36	2E	30	38	33	36	36	38	33	32
00000040	38	38	36	30	36	32	0D	0A	33	2E	30	33	39	30	33	34
00000050	37	32	37	35	31	33	33	34	0D	0A						

Почему файл занимает 90 байта?

Появились символы с 13 и 10

13 - CR (перевод каретки). Курсор перемещается к левому краю поля, не переходя на другую строку.

10 - LF (перевод строки). Курсор перемещается на следующую строку.

3 Текстовый файл

Запись текста в текстовый файл

```
var
  f: text;
  s: string;
begin
  assign(f,'MyText.txt');
  rewrite(f);
  repeat
    readln(s);
    if length(s)>0 then
      writeln(f,s);
  until length(s)=0;
  close(f);
end.
```

Файлы

1. Типы файлов
2. Доступ к файлу
3. Текстовые файлы
4. **Типизированный файлы**
5. Нетипизированные файлы
6. Умная Йохансон

4 Типизированный файл

Работа с типизированными файлами несколько отличается от работы с текстовыми файлами, а именно:

- уже существующий файл может быть инициирован для чтения и записи процедурой RESET.
- для чтения и записи используются соответственно процедуры READ и WRITE.
- для работы с типизированными файлами могут использоваться процедуры и функции **SEEK, FILESIZE, FILEPOS**

4 Типизированный файл

Процедура SEEK. Смещает указатель файла к требуемому компоненту. Формат обращения:

SEEK(<ф.п.>, <№ компонента>)

Здесь <№ компонента> – выражение типа LONGINT, указывающее новый номер компонента файла.

Функция FILESIZE. Возвращает значение типа LONGINT, которое содержит количество компонентов файла. Формат обращения:

FILESIZE(<ф.п.>)

Функция FILEPOS. Возвращает значение типа LONGINT, содержащее порядковый номер компонента файла, который будет обрабатываться следующей операцией ввода-вывода. Формат обращения:

FILEPOS(<ф.п.>)

3 Текстовый файл

Заполнение типизированного файла

type

```
Person = record  
    name: string;  
    year: integer;  
end;
```

var

```
f: file of Person;  
Rec: Person;  
yes: integer;
```

begin

```
assign(f,'List.dat');  
rewrite(f); {для добавления в файл reset(f); seek(f, filesize(f)); }  
repeat  
    write('Введи фамилию    : '); readln(Rec.name);  
    write('Введи год рождения : '); readln(Rec.year);  
    write(f, Rec);  
    writeln('Добавить ещё одну запись? (1 - да, 0 - нет)'); readln(yes);  
until yes<>1;  
close(f);
```

end.

4 Типизированный файл

Вывод на экран содержимого типизированного файла

type

```
Person = record
    name: string;
    year: integer;
end;
```

var

```
f: file of Person;
Rec: Person;
yes: integer;
```

begin

```
assign(f,'List.dat');
reset(f);
while not eof(f) do
    begin
        read(f, Rec);
        writeln(Rec.name, Rec.year:6);
    end;
close(f);
```

end.

Файлы

1. Типы файлов
2. Доступ к файлу
3. Текстовые файлы
4. Типизированный файлы
5. **Нетипизированные файлы**
6. Умная Йохансон

5 Нетипизированный файл

Длина, байт	Название	Диапазон значений
1	Byte	0..255
1	Shortint	-128..+127
2	Word	0..65535
2	Smallint	-32768..32767
4	Integer, Longint	-2147483648..2147483647

Длина, байт	Название	Диапазон значений	Значащих цифр
6	Real	$2,9 \times 10^{-39} - 1,7 \times 10^{38}$	11-12
4	Single	$1,5 \times 10^{-45} - 3,4 \times 10^{38}$	7-8
8	Double	$5 \times 10^{-324} - 1,7 \times 10^{308}$	15-16
10	extended	$3,4 \times 10^{-4932} - 1,1 \times 10^{4932}$	19-20

5 Нетипизированный файл

//запись в файл 5 чисел типа shortint (1 байт -128..127)

```
const N = 5;
var
  f1 : file;
  arr : array [1..N] of shortint ;
  i : byte;
begin
  arr[1] := 0;
  arr[2] := 127;
  arr[3] := -128;
  arr[4] := -1;
  arr[5] := 1;
  assign(f1, 'test.dat');
  rewrite(f1);
  for i := 1 to N do
    write(f1, arr[i]);
  close(f1);
end.
```

test.dat						
Offset	0	1	2	3	4	5
00000000	00	7F	80	FF	01	

hex 00 – bin 0000 0000 dec 0
hex 7F – bin 0111 1111 dec 127
hex 80 – bin 1000 0000 dec 128
hex FF – bin 1111 1111 dec 255
hex 01 – bin 0000 0001 dec 1

5 Нетипизированный файл

Десятичное представление	Двоичное представление (8 бит)		
	прямой	обратный	дополнительный
127	01111111	01111111	01111111
1	00000001	00000001	00000001
0	00000000	00000000	00000000
-0	10000000	11111111	---
-1	10000001	11111110	11111111
-2	10000010	11111101	11111110
-3	10000011	11111100	11111101
-4	10000100	11111011	11111100
-5	10000101	11111010	11111011
-6	10000110	11111001	11111010
-7	10000111	11111000	11111001
-8	10001000	11110111	11111000
-9	10001001	11110110	11110111
-10	10001010	11110101	11110110
-11	10001011	11110100	11110101
-127	11111111	10000000	10000001
-128	---	---	10000000

1 байт со знаком:
//опер. в доп.

коде

$127 + 1 = ?$

127: 0111 1111

1: 0000 0001

+: 1000 0000

обр: 1111 1110

прям: 1000 0001

ответ: -1

5 Нетипизированный файл

```
//запись в файл 5 чисел типа smallint(2 байт -32768..32767)
const N = 4;
var
  f1 : file;
  arr : array [1..N] of smallint;
  i : byte;
begin
  arr[1] := 0;
  arr[2] := 32767;
  arr[3] := -1;
  arr[4] := 16;
  assign(f1, 'test.dat');
  rewrite(f1);
  for i := 1 to N do
    write(f1, arr[i]);
  close(f1);
end.
```

5 Нетипизированный файл

test.dat								
Offset	0	1	2	3	4	5	6	7
00000000	00	00	FF	7F	FF	FF	10	00

```
arr[1] := 0;  
arr[2] := 32767;  
arr[3] := -1;  
arr[4] := 16;
```

hex 00 00 – bin 0000 0000 0000 0000 dec 0

hex 7F FF – bin 0111 1111 1111 1111 dec 32767

hex FF FF – bin 1111 1111 1111 1111 dec 65535

hex 00 10 – bin 0000 0000 0001 0000 dec 16

 Младший
байт
Старший
байт

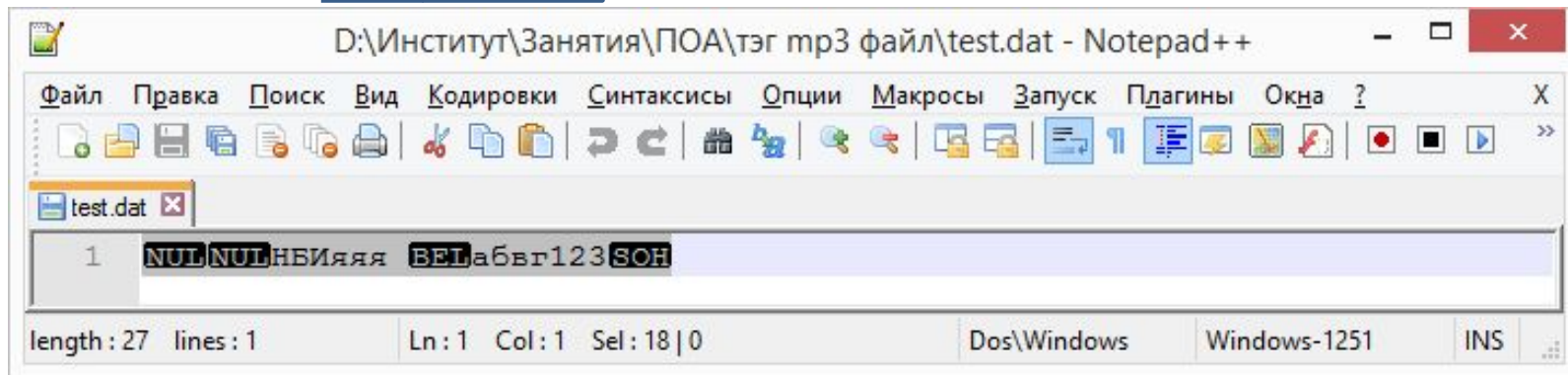
5 Нетипизированный файл

```
//запись в файл переменных разного типа
var
  f1 : file;
  vr1 : single ;
  vr2 : integer;
  vr3 : byte;
  vr4 : string;
  vr5 : boolean;
begin
  vr1 := -12.5;  vr2 := -56;  vr3 := 32;
  vr4 := 'абвр123';  vr5 := true;
  assign(f1, 'test.dat');
  rewrite(f1);
  write(f1, vr1);  write(f1, vr2);  write(f1, vr3);
  write(f1, vr4);  write(f1, vr5);
  close(f1);
end.
```

5 Нетипизированный файл

Содержимое файла

test.dat	Single -12.5				Integer -56				Byte 32	String абвг123 (кодировка CP1251)						
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00000000	00	00	48	C1	C8	FF	FF	FF	20	07	E0	E1	E2	E3	31	32
00000016	33	01														
	Boolean true															



Код	Название	Назначение
00	NULL	Этот символ ничего не делает
01	START OF HEADING	В настоящее время используется в консоли маршрутизаторов Cisco.
07	BELL	Если этот символ послать на принтер или на терминал, то ничего не напечатается, но послышится звуковой сигнал.

5 Нетипизированный файл

Пример: Прочитать ID3(v1-v1.1) теги из mp3 файла

Поле	Длина	Описание
заголовок	3	«TAG»
название	30	30-символьное название
исполнитель	30	30-символьное имя исполнителя
альбом	30	30-символьное название альбома
год	4	Строковая запись года
комментарий	28 или 30	Комментарий
нулевой байт	1	Если номер трека присутствует, этот байт равен 0.
track	1	Номер трека в альбоме или 0. Учитывается только если предыдущее поле=0.
жанр	1	Индекс в списке жанров или 255

5 Нетипизированный файл

Структура программы для чтения метаданных:

- открыть файл для чтения;
- отступить на 128 байт с конца файла;
- прочитать заголовок;
- если заголовок «TAG» - прочитать поля;
- закрыть файл.

Основные сведения | Обложка | ID3v1 | ID3v2

ID3v1

Добавить теги ID3v1 в файл

Номер дорожки

Название

Исполнитель

Альбом

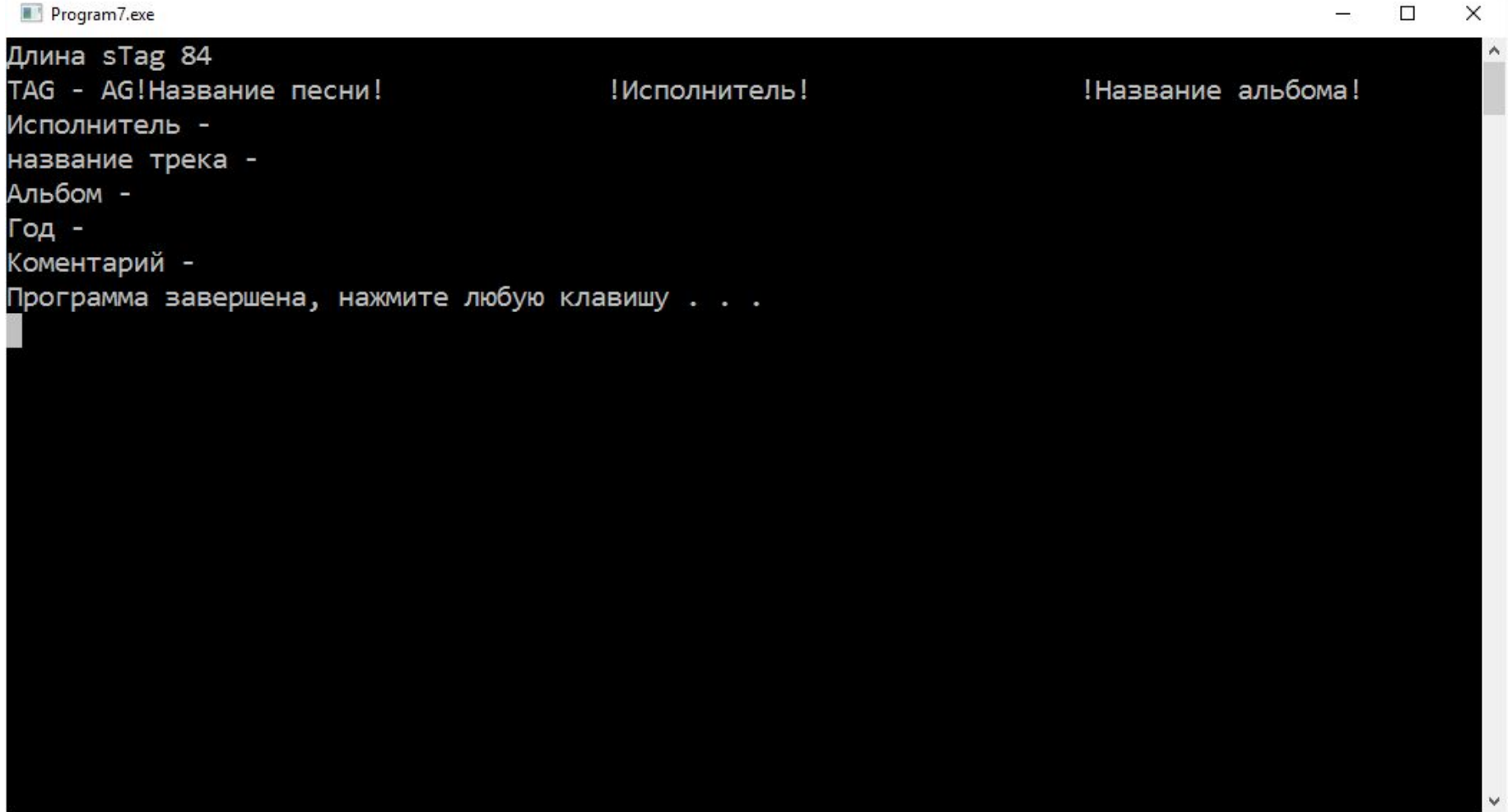
Год Жанр

Примечание

5 Нетипизированный файл

```
var
  f_music : file;
  stag : string[3];
  sTitle,sArtist,sAlbum :string[30];
  sYear : string[4];sComment :string[28];
  fsize : int64; i :integer;
begin
  assign(f_music, 'music.mp3'); reset(f_music);
  fsize := FileSize(f_music); seek(f_music,fsize-128);
  read(f_music,sTag);
  writeln('Длина sTag 'length(sTag)); writeln('TAG - ',sTag);
  read(f_music,sTitle); read(f_music,sArtist);
  read(f_music,sAlbum); read(f_music,sYear);
  read(f_music,sComment);
  close(f_music);
  writeln('Исполнитель - ',sArtist);
  writeln('название трека - ',sTitle);
  writeln('Альбом - ',sAlbum);
  writeln('Год - ',sYear);
  writeln('Коментарий - ',sComment);
end.
```

5 Нетипизированный файл



```
Program7.exe
Длина sTag 84
TAG - AG!Название песни! !Исполнитель! !Название альбома!
Исполнитель -
название трека -
Альбом -
Год -
Коментарий -
Программа завершена, нажмите любую клавишу . . .
```

Почему длина переменной sTag 84 символа?

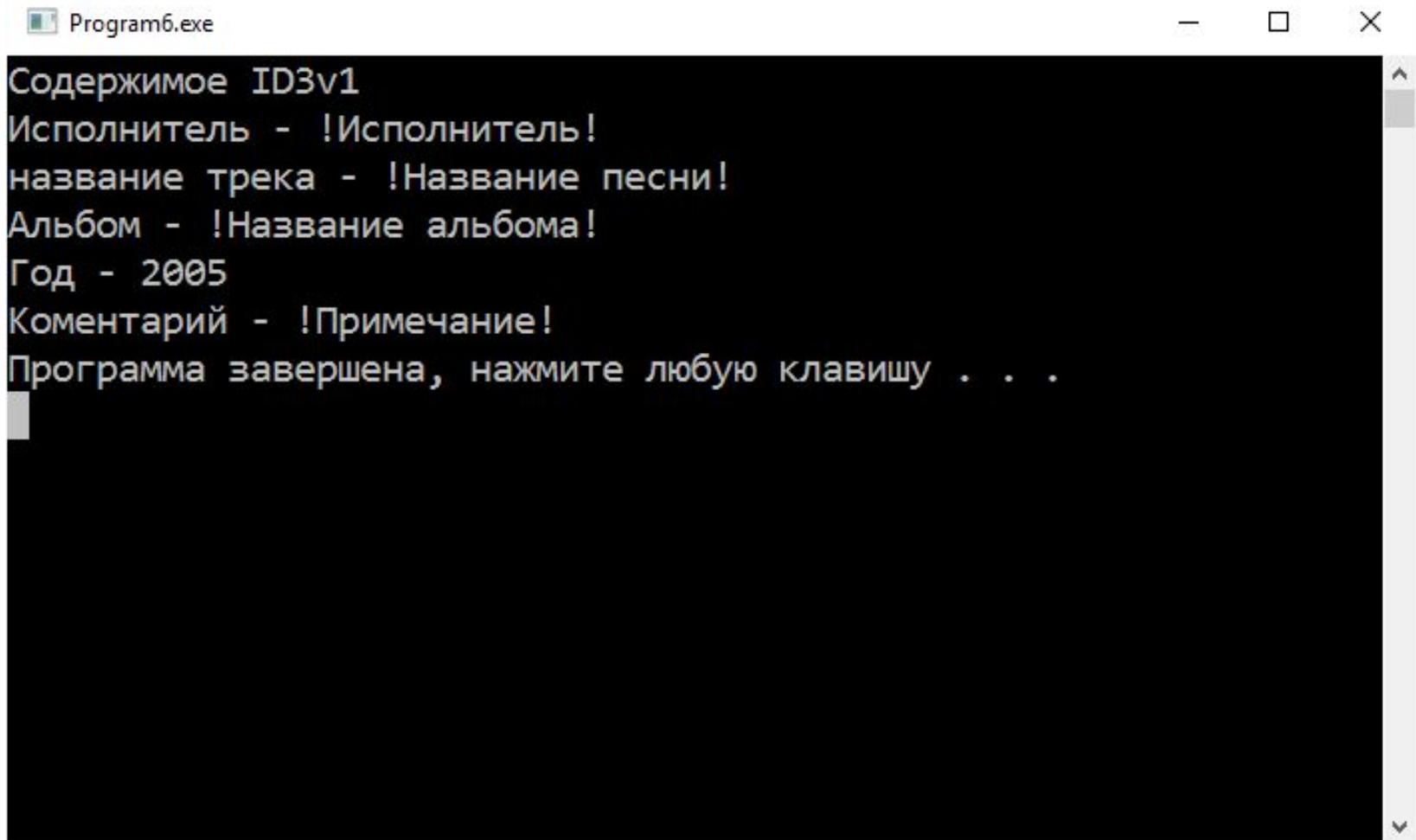
5 Нетипизированный файл

```
var
  f_music : file;
  stag : array [1..3] of byte;
  sTitle, sArtist, sAlbum : array [1..30] of byte;
  sYear : array [1..4] of byte;
  sComment : array [1..28] of byte;
  fsize : int64; i : integer;
begin
  assign(f_music, 'music.mp3');
  reset(f_music);
  fsize := FileSize(f_music);
  seek(f_music, fsize-128);
  read(f_music, sTag);
```

5 Нетипизированный файл

```
if Concat(chr(sTag[1]),chr(sTag[2]),chr(sTag[3]))='TAG' then  
  begin  
    read(f_music,sTitle); read(f_music,sArtist);  
    read(f_music,sAlbum); read(f_music,sYear);  
    read(f_music,sComment);  
    writeln('Содержимое ID3v1');  
    write('Исполнитель - ');  
    for i := 1 to 30 do write(chr(sArtist[i]));  
    writeln; write('название трека - ');  
    for i := 1 to 30 do write(chr(sTitle[i])); writeln;  
    write('Альбом - ');  
    for i := 1 to 30 do write(chr(sAlbum[i])); writeln;  
    write('Год - ');  
    for i := 1 to 4 do write(chr(sYear[i])); writeln;  
    write('Комментарий - ');  
    for i := 1 to 28 do write(chr(sComment[i])); writeln;  
  end  
  else writeln('ID3v1 не заполнен');  
  close(f_music);  
end.
```


5 Нетипизированный файл



```
Program6.exe
Содержимое ID3v1
Исполнитель - !Исполнитель!
название трека - !Название песни!
Альбом - !Название альбома!
Год - 2005
Коментарий - !Примечание!
Программа завершена, нажмите любую клавишу . . .
```

5 Нетипизированный файл

Пример: Вывести графическом режиме изображение, которое хранится в файле

Структура файла

- 0 4 Смещение, с которого начинается само изображение
- 4 3*N Палитра, хранится три составляющие цвета R, G, B
- 3*N+4 M Координаты точечного изображения (x,y)+цвет, 3 байта точку

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	0C	00	00	00	F1	46	17	00	B0	F0	00	B0	50	13	05	00
00000010	1B	05	00	12	06	00	13	06	00	14	06	00	1A	06	00	1B
00000020	06	00	1C	06	00	13	07	00	1B	07	00	16	0A	01	17	0A
00000030	01	18	0A	01	14	0F	02	1A	0F	02	15	10	02	16	10	02
00000040	17	10	02	18	10	02	19	10	02							

5 Нетипизированный файл

```
uses graphABC;
```

```
type
```

```
TColor = record
```

```
  r,g,b : byte;
```

```
end;
```

```
TPoint = record
```

```
  x,y,c : byte;
```

```
end;
```

```
var
```

```
f: file;
```

```
offset: longword;
```

```
Color : array of TColor;
```

```
point : array of TPoint;
```

```
i,j : longword;
```

5 Нетипизированный файл

begin

```
assign(f,'pointpict.dat');
```

```
reset(f); //открываем для чтения
```

```
read(f,offset); // откуда начинается изображение
```

```
SetLength(Color,(offset-4) mod 3+1); // кол-во цветов
```

```
for i := 0 to Length(Color)-1 do //заполним палитру
```

```
    read(f,Color[i]);
```

```
i := 0;
```

```
while not eof(f) do //читаем точки
```

```
    begin
```

```
        SetLength(point,i+1);
```

```
        read(f,point[i]);
```

```
        i := i + 1;
```

```
    end;
```

```
close(f);
```

5 Нетипизированный файл

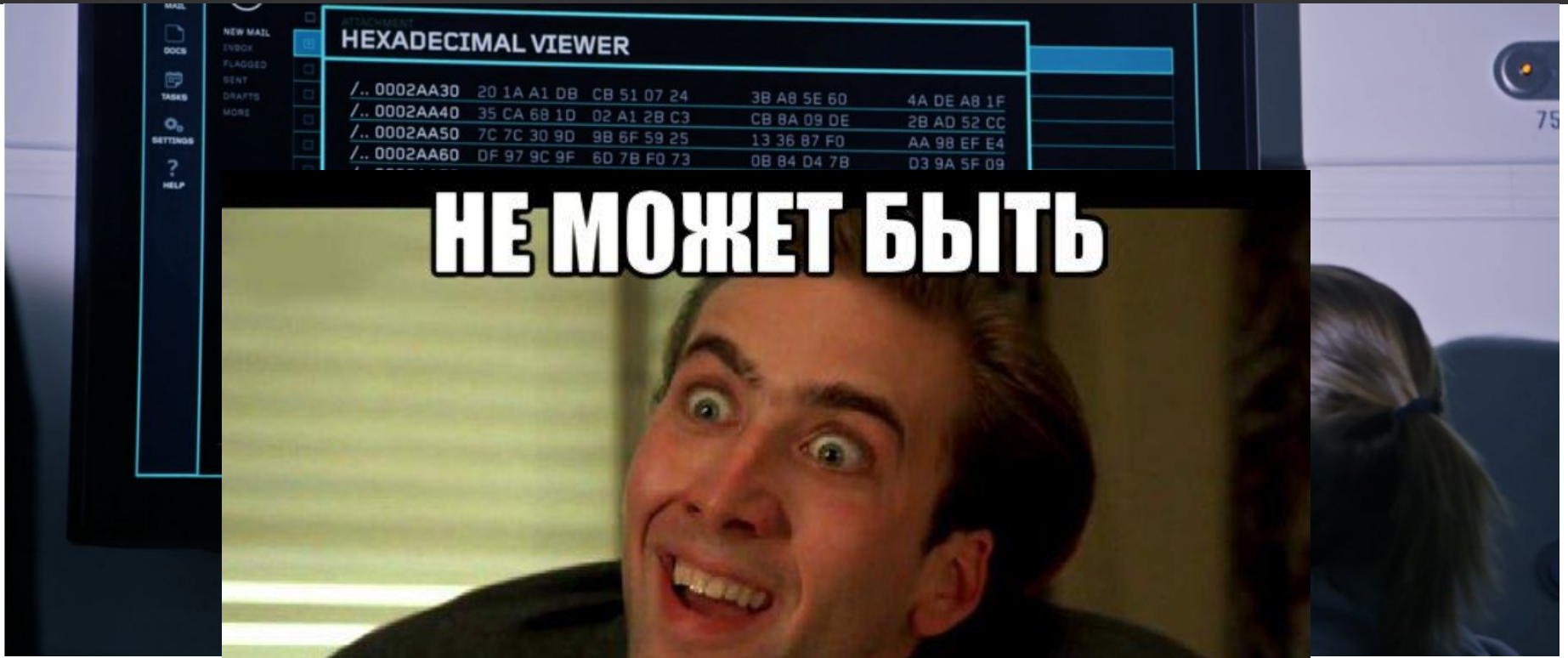
```
//рисуем изображение
for i := 0 to Length(point)-1 do
  begin
    j := point[i].c;
    SetPixel(point[i].x,point[i].y,RGB(color[j].r,color[j].g,color[j].b));
  end;
SetWindowSize(30,30);
SaveWindow('wnd.bmp');
end.
```



Файлы

1. Типы файлов
2. Доступ к файлу
3. Текстовые файлы
4. Типизированный файлы
5. Нетипизированные файлы
6. **Умная Йохансон**

6 Умная Йохансон



НЕ МОЖЕТ БЫТЬ

- Это не

КАК ТЫ ДОГАДАЛАСЬ ???!!!

risovach.ru

Выдумка
сценаристов?

6 Умная Йохансон

JPEG файл 1

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	48
00000010	00	48	00	00	FF	E1	3D	8E	45	78	69	66	00	00	4D	4D
00000020	00	2A	00	00	00	08	00	0D	01	00	00	04	00	00	00	01
00000030	00	00	0A	00	01	01	00	04	00	00	00	01	00	00	07	80
00000040	01	03	00	03	00	00	00	01	00	06	00	00	01	0F	00	02
00000050	00	00	00	08	00	00	00	AA	01	10	00	02	00	00	00	09
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

JPEG файл 2

00000000	FF	D8	FF	E1	5D	09	45	78	69	66	00	00	4D	4D	00	2A
00000010	00	00	00	08	00	0D	01	00	00	04	00	00	00	01	00	00
00000020	0A	00	01	01	00	04	00	00	00	01	00	00	07	80	01	03
00000030	00	03	00	00	00	01	00	06	00	00	01	0F	00	02	00	00
00000040	00	08	00	00	00	AA	01	10	00	02	00	00	00	09	00	00
00000050	00	B2	01	12	00	03	00	00	00	01	00	01	00	00	01	1A
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

JPEG файл 3

00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	60
00000010	00	60	00	00	FF	E1	60	1A	45	78	69	66	00	00	4D	4D
00000020	00	2A	00	00	00	08	00	0E	01	00	00	04	00	00	00	01
00000030	00	00	0A	00	01	01	00	04	00	00	00	01	00	00	07	80
00000040	01	03	00	03	00	00	00	01	00	06	00	00	01	0F	00	02
00000050	00	00	00	08	00	00	08	C2	01	10	00	02	00	00	00	09
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Что общего у этих файлов?

-
маркеры

Маркеры JPEG файла

Маркер	Байты	Длина	Назначение
SOI	0xFFD8	<i>нет</i>	Начало изображения
SOF0	0xFFC0	<i>переменный размер</i>	Начало фрейма (базовый, ДКП)
SOF1	0xFFC1	<i>переменный размер</i>	Начало фрейма (расширенный, ДКП, код Хаффмана)
SOF2	0xFFC2	<i>переменный размер</i>	Начало фрейма (прогрессивный, ДКП, код Хаффмана)
DHT	0xFFC4	<i>переменный размер</i>	Содержит таблицы Хаффмана
DQT	0xFFDB	<i>переменный размер</i>	Содержит таблицы квантования
DRI	0xFFDD	4 байта	Указывает длину рестарт-интервала
SOS	0xFFDA	<i>переменный размер</i>	Начало сканирования
RSTn	0xFFD n	<i>нет</i>	Перезапуск
APPn	0xFFE n	<i>переменный размер</i>	Задаётся приложением
COM	0xFFFE	<i>переменный размер</i>	Комментарий
EOI	0xFFD9	<i>нет</i>	Конец закодированной части изображения.

Вывод: по двоичному представлению можно определить это JPEG или нет.

А является ли он текстовым в кодировке ASCII?

Особенности текстового файла:

- наличие пробелов (dec 32, hex 20)
- CR+ LF (перевод каретки dec 13, hex 0D; перевод строки dec 10, hex 0A)
- A-Z (hex 41 – 5A), a-z (hex 61 – 7A), 0 – 9 (hex 30 – 39)

6 Умная Йохансон

```
/.. 0002AA30 20 1A A1 DB CB 51 07 24 3B AB 5E 60 4A DE AB 1F
/.. 0002AA40 35 CA 68 1D 02 A1 2B C3 CB BA 09 DE 2B AD 52 CC
/.. 0002AA50 7C 7C 30 9D 9B 6F 59 25 13 36 87 F0 AA 98 EF E4
/.. 0002AA60 DF 97 9C 9F 6D 7B F0 73 0B 84 D4 7B D3 9A 5E 09
```

ЭТО НЕ ТЕКСТОВЫЙ ФАЙЛ В КОДИРОВКЕ

AS

```
/.. 0002AA70 65 CB 7B 79 9C 15 44 68 6D A1 2E 61 48 6C 00 A0
/.. 0002AA80 27 FE DD 65 68 DB 13 3B 09 17 A9 AB 9C 22 63 0B
/.. 0002AA90 44 B8 65 B9 60 D0 3F D6 37 35 CC 72 80 04 62 6A
/.. 0002AAC0 15 31 B6 E3 FC ED 85 A3 75 B4 41 5B 61 8A 2C C4
/.. 0002AAD0 81 90 E2 77 31 A0 1C A3 D6 83 2F C3 50 ED 9F 11
/.. 0002AAE0 5C 5D D5 B3 E7 BC CB 82 1B F5 AE 5B FB 47 8E 16
```

Таблица ASCII символов

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL