

Виноградов Владимир Николаевич

Муниципальное бюджетное образовательное учреждение

«Средняя общеобразовательная школа № 5 г. Губкинский, ЯНАО»

КОНСПЕКТ УРОКА
«ТЕКСТОВЫЕ ФАЙЛЫ В PASCAL,
ОПЕРАЦИИ НАД ФАЙЛАМИ»

Тема урока: *«Текстовые файлы в PASCAL, операции над файлами».*

Задачи урока:

1. Образовательные – *получение и систематизация знаний учащихся по теме «Операции над текстовыми файлами»;*
 - a) *Знать и понимать назначение процедуры ASSIGN;*
 - b) *Уметь применять процедуры Reset, Rewrite, Append для записи, открытия и пополнения текстового файла;*
 - c) *Понимать назначение функции EOF и процедуры Close;*
2. Развивающая - *развитие приемов умственной деятельности (обобщение, анализ, синтез), памяти, алгоритмического и логического мышления при решении задач и составлении программ, творческой активности.*
3. Воспитательная - *развитие познавательного интереса учащихся, основ коммуникативного общения, уверенности в своих силах.*

4. **Личностные:**

Проявление творческого отношения к процессу обучения;

Проявление эмоционально-ценностного отношения к учебной проблеме.



5. Метапредметные

Познавательные:

Умение:

перерабатывать полученную информацию, преобразовывать информацию из одной формы в другую; обобщать полученную информацию; давать оценку своим действиям, оценивать результат; находить ответы на вопросы, используя свой жизненный опыт и информацию, полученную на уроке.

6. **Регулятивные**

Умение:

Работать по предложенному учителем плану;

Формулировать вопрос, проблему, затруднение, оценивать сложившуюся учебную ситуацию.

7. **Коммуникативные**

Умение:

оформлять свою мысль в устной речи, высказывать свою точку зрения, формулировать высказывание;

сотрудничать, договариваться о последовательности действий и результате, учиться представлять другим ход работы и ее результат, слушать мнение других;

адекватно использовать речевые средства для дискуссии и аргументации своей позиции.

Тип урока: *комбинированный урок объяснения нового материала и закрепления полученных знаний.*

Время урока: *продолжительность 45 мин.*

Место урока в учебном плане: второй урок по теме файловые типы, файловые переменные.



Оборудование урока:

1. *Компьютеры;*
2. *Карточки с заданиями практической и самостоятельной работы;*
3. *Карточки с вопросами;*
4. *Карточки блок-схемы программы;*
5. *Программа Turbo Pascal, Pscal ABC, NetOp School;*

План урока:

1. Организационный момент (1 мин);
2. Сообщение темы и постановка целей урока (1 мин);
3. Беседа с учащимися по пройденному материалу (2 мин)
4. Объяснение нового материала (10 мин);
5. Практическая работа учащихся (15 мин);
6. Самостоятельная работа учащихся (14 мин);
7. Подведение итогов урока (1 мин);
8. Домашнее задание (1 мин);

Ход урока

Работа с учащимися по пройденному материалу:

Ответить на вопросы:

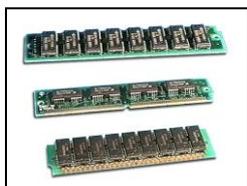
1. *Как необходимо обозначать переменные в Паскале?*
2. *Назовите все известные вам типы переменных, используемых в Паскале.*
3. *Описание типов переменных.*

Объяснение нового материала учителем:

Что такое файл (стр.122, Turbo Pascal учитесь программировать О.А.Меженный)



Учитель: Типы данных, с которыми мы до сих пор имели дело, предназначены для манипулирования информацией, содержащейся в *оперативной памяти* компьютера.



Однако, как известно, *оперативная память* способна хранить данные только временно – пока включен компьютер. Представим себе, что в течение дня производился какой-то сложный расчет, к результатам которого придется обращаться впоследствии? Или создавали базу данных предприятия. Как сохранить эти результаты, чтобы они были в целости и сохранности, постоянно под рукой, и при этом иметь возможность выключать компьютер, когда это необходимо? Очевидно, для эффективной работы нужен какой-то способ продолжительного хранения информации, чтобы ею можно было воспользоваться после выключения и включения компьютера снова. И такой способ существует. Для долговременного хранения информация из *оперативной памяти* переносится в *файлы*.

 (Английское слово *file* переводится как архив, подшивка, картотека – иными словами – хранилище информации).

Файл представляет собой некоторое поименованное место на внешнем носителе. В качестве внешнего носителя могут служить различного рода диски (гибкие, жесткие, магнитооптические, компакт – диски) и микросхемы (Flash-память).





Наконец, в качестве такого «условного» внешнего носителя могут рассматриваться виртуальные диски, которые можно создавать (с помощью команды VDISK) в оперативной памяти компьютеров, работающих под управлением операционной системы MS DOS.

Файловые типы и файловые переменные в Pascal.

Все сказанное выше относится к долговременному хранению информации в компьютерах вообще. А как это реализовано в *Turbo Pascal*?

```
program luk;
var
  fl:text;
  s:string;
begin
  assign(fl,'c:\ASPUSKIN.txt');
  rewrite(fl);
  writeln('Введите стихи построчно');
  readln(s);
  while s<>'' do
  begin
    writeln(fl,s);
    readln(s);
  end;
  close(fl);
end.
```

Для сохранения информации в *Turbo Pascal* предусмотрена возможность определять *файловые типы и файловые переменные*. После этого, информацию, которая может потребоваться в последствии, можно перенести в *файл* на диске.

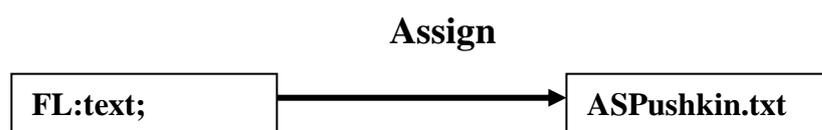
Существует три способа определения файлового типа (или файловой переменной). Эти три способа соответствуют трем видам файлов, которыми

позволяет оперировать Pascal. Речь идет о *типизированных, текстовых и нетипизированных файлах*.

Сегодня мы будем говорить о *текстовых файлах*. Текстовые файлы хранят информацию в символьном виде, и эта информация разбита на строки. С текстовым файлом на магнитном диске связывается файловая переменная в Паскале, тип которой объявляется как *text*.

Организация доступа к файлам.

Однако при объявлении файловой переменной не указывается ни имя файла, ни в каком каталоге он содержится. Для того чтобы конкретный файл (например, на диске С текстовый файл **ASPushkin.txt**) стал доступным, его необходимо как-то связать с ранее объявленной *файловой переменной* (например, *FL*). Такое **связывание** осуществляется с помощью процедуры **Assign**, которая является стандартной процедурой Pascal.



Заголовок процедуры Assign выглядит следующим образом:

Assign (var FL: file_name:string);

Здесь FL – имя файловой переменной любого вида, с которой ассоциируется файл с диска, имеющий имя *file_name* (параметр *file_name* принадлежит типу *string*).

1. Открытие файла для чтения, записи, чтения-записи одновременно.

Открыть файл в Pascal можно для *чтения*, для *записи*, а также для *чтения и записи* одновременно. До открытия файл должен быть связан (с помощью процедуры *Assign*) с ранее объявленной файловой переменной.

При открытии файла ищется уже существующий файл на диске, либо

создается новый (если файл открывается для записи) и текущий указатель ставится в начало файла.

Для открытия файлов предназначены процедуры *Reset* (только для чтения), *Rewrite* (только для записи), *Append* (для изменения или пополнения). Причем, если первые две процедуры подходят для открытия файла любого вида, то с помощью процедуры *Append* можно открыть только текстовый файл.

2. Алгоритм чтения, создания и записи текстового файла программой (презентация блок-схемы демонстрируется на интерактивную доску и мониторы учащихся посредством программы NET Op School):

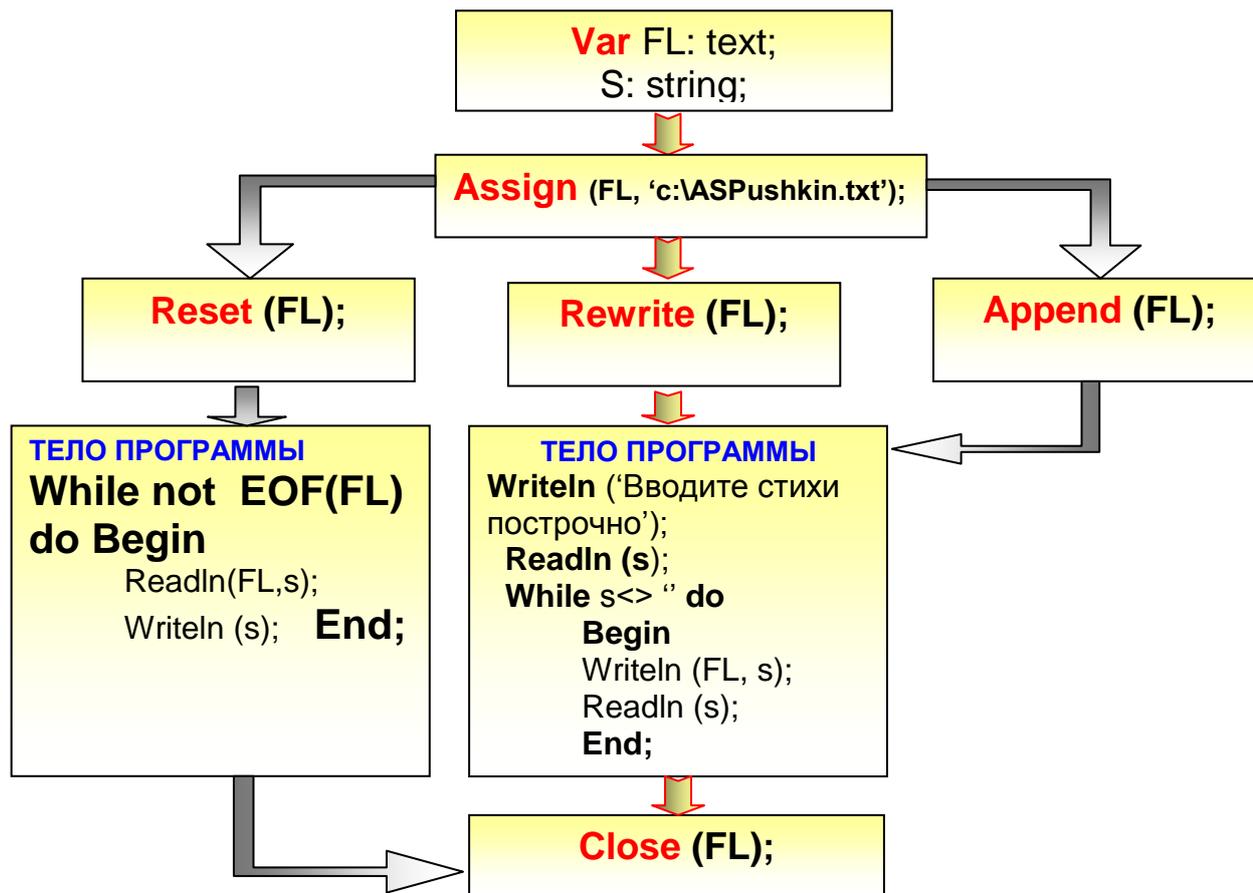
Блок- схема алгоритма.



ПРАКТИЧЕСКОЕ ЗАДАНИЕ:

Используя Pascal создать текстовый файл c:\AS Pushkin.txt, записать в него известное четверостишие знаменитого поэта из стихотворения «Руслан и Людмила».

Блок-схема решения задачи.



Алгоритм создания программы:

1. Ввести переменную файлового типа **FL**:
2. Установить связь между этой переменной **FL** и текстовым файлом **AS Pushkin.txt** с помощью процедуры `Assign` (`Assign, FL, 'c:\ AS Pushkin.txt'`);).
3. Открыть текстовый файл для *записи* процедурой `Rewrite` (`Rewrite (FL);`), для *чтения* процедурой `Reset`(`Reset(FL);`), для изменения (пополнения) файла процедурой `Append` (`Append(FL);`). В файл, открытый для записи или пополнения, можно только писать, из файла, открытого для чтения – можно только считывать данные. Для дополнения информации в существующий текстовый файл его открывают процедурой *Append*. При этом файл откроется для записи, а указатель позиции установится в конец

файла, после чего в файл можно писать – информация будет добавляться в его конец.

4. Закрывать файл процедурой `Close (Close(FL);)`. Процедура *Close* сообщает операционной системе все сведения о созданном файле (дату его создания, размер и т.д.)

6. Запись в текстовый файл (Программа «У Лукоморья».)

Program LUK_REW;

Var

FL: text; { Описание файловой переменной }

S: string; { имя вспомогательной переменной списка ввода }

Begin

Assign (FL, 'c:\AS Pushkin.txt');{ Устанавливается связь файловой переменной FL с файлом AS Pushkin.txt (это имя файла на диске, которое в программе не появляется, его замещает имя FL). }

Rewrite (FL) ;{ открытие файла AS Pushkin.txt для записи, если в нем уже было что-то записано, все будет уничтожено }

Writeln ('Вводите стихи построчно');

Readln (s);

While s<> '' do

Begin

Writeln (FL, s);

Readln (s);

End;

Close (FL); {Файл закрыт }

Writeln ('Введенное четверостишие записано в файл AS Pushkin.txt');

Readln;

End.



В этой программе сначала вводится переменная файлового типа **FL**. Затем с помощью процедуры **Assign** устанавливается связь между этой переменной и файлом **AS Pushkin.txt**, который создается (или уже создан) на диске компьютера. После установления связи внешнего файла с файловой переменной внешний файл надо *открыть* для записи или чтения. В нашем примере процедура **Rewrite** выполняет все подготовительные действия, необходимые для записи в файл. Каждая строка, набранная на клавиатуре, считывается в переменную **S** строкового типа, а затем с помощью файловой переменной **FL** передается в файл **AS Pushkin.txt**.

При нажатии клавиши **ENTER** вводятся символы #13 и #10, завершающие строку. Запись в файл будет осуществляться до тех пор, пока не будет введена пустая строка. Процедура **Close** «закрывает» файл. После закрытия файла связанный с ним внешний файл обновляется, затем файловая переменная может быть связана с другим внешним файлом. Работа с файлом – запись в файл или чтение из него – обязательно должны завершаться закрытием файла с помощью процедуры **Close**.

Закрепление нового материала:

7. Практическая работа учащихся(Rewrite):

1. Запустите Turbo Pascal.
2. Наберите предложенный текст программы.
3. Запустите программу.
4. Наберите знаменитое четверостишие, завершите ввод двойным нажатием клавиши ENTER.

*У Лукоморья дуб зеленый,
Златая цепь на дубе том,
И днем, и ночью кот ученый,
Все ходит по цепи кругом.*



5. Сохраните программу под именем **LUK_REW.pas**.
6. Откройте текстовый файл **ASPushkin.txt**, расположенный на диске **C**, просмотрите его, убедитесь в том, что информация хранится в символьном, удобном для просмотра виде.

8. Дополнение текстового файла (Append):

Открыть программу **LUK_REW.pas**.

Внести изменения в программу, необходимые для дополнения в текстовый файл **ASPushkin.txt** **второго** четверостишия.

Program LUK;

Var

FL: text; {Описание файловой переменной}

s: string; {имя вспомогательной переменной списка ввода}

Begin

assign(FL, 'c:\AS Pushkin.txt');{Устанавливается связь файловой переменной **FL** с файлом **AS Pushkin.txt** (это имя файла на диске, которое в программе не появляется, его замещает имя **FL**). }

Append (FL);{ открытие файла **AS Pushkin.txt** для дополнения существующей информации}

Writeln ('Вводите стихи построчно');

Readln (s);

While s<> '' do

Begin

Writeln (FL, s);

Readln (s);

End;

Close (FL); {Файл закрыт}



```
Writeln ('Введенное четверостишие добавлено в файл ASPushkin.txt');  
Readln;  
End.
```

9. Чтение из текстового файла (*Reset*):

1. Открыть программу **LUK_REW.pas**.

Внести изменения в программу, необходимые для чтения из текстового файла *AS* *Pushkin.txt*

```
Program LUK;
```

```
Var
```

```
  FL:text; {Описание файловой переменной}
```

```
  s: string; {имя вспомогательной переменной списка ввода}
```

```
Begin
```

```
  assign(FL, 'c:\ASPushkin.txt'); {Устанавливается связь файловой  
переменной FL с файлом ASPuchkin.txt (это имя файла на диске,  
которое в программе не появляется, его замещает имя FL ). }
```

```
  Reset (FL) ; {открытие файла ASPushkin.txt для чтения}
```

```
  While not EOF (FL) do {Чтение продолжается пока не достигнут конец  
файла}
```

```
    Begin
```

```
      Readln (FL, s); {Чтение из файла ASPushkin.txt в S  
построчно}
```

```
      Writeln (s); {Вывод прочитанного на экран}
```

```
    End;
```

```
    Close (FL); {Заккрытие файла}
```

```
End.
```



Процедура **Reset** устанавливает указатель текущей позиции файла на начало первой строки текстового файла. Процедура **Readln** считывает строку полностью, сохраняет считанное значение в переменной **S** и передвигает указатель на начало следующей строки. Так продолжается до тех пор, пока указатель не достигнет конца файла. (**EOF- End Of File признак конца файла**).

Сохранить программу под именем **LUK_RES.pas**.

Практическое задание (дописать предложения).

Текст задания передается на компьютеры программой NetOp Shool, сохраняется учащимися в своих папках под именем FAMILIE в формате RTF.

1. Процедура Assign предназначена для ...
2. Для открытия текстовых файлов используют процедуры...
3. Процедура Rewrite предназначена для...
4. Процедура Append предназначена для...
5. Процедура Reset предназначена для...
6. Процедура Close предназначена для...
7. EOF означает ...

Дополнительное задание (№225, Н. Кетков «Turbo Pascal в задачах и примерах»)

1. Написать программу, которая создает на диске C компьютера файл *numbers.txt* и записывает в него 5 введенных пользователем целых чисел.

Program NUMER ;

Var



FL: text;

n: integer; {число, прочитанное из файла}

i: integer;

Begin

Writeln ('Ведите 5 целых чисел');

Assign (FL, 'c:\numbers.txt');

Rewrite(FL);

For i: =1 to 5 do

Begin

Readln (n);

Writeln (FL, n);

End;

Close (FL);

Writeln ('Введенные числа записаны в файл numbers.txt');

Readln;

End.

(Ввести числа 2, 3, 4, 5, 8);

2. Внести изменения в программу, позволяющие пользователю дописать в данный текстовый файл пять новых целых чисел (изменить в предыдущей программе **Rewrite** на **Append**). *(Ввести числа 1,6,7,9,10).*

3. Вычислить среднее арифметическое чисел, находящихся в файле **numbers.txt**.

*(Ввести переменные K, S, Sa соответствующие количеству прочитанных чисел, сумме прочитанных чисел и среднему арифметическому прочитанных чисел. Открыть текстовый файл для чтения процедурой **Reset**)*



Program NUMER;

Var

FL: text;

n: integer; {число, прочитанное из файла}

i, k, s: integer;

sa: real;

Begin

Writeln ('Вычисление среднего арифметического чисел,');

Writeln ('находящихся в файле c:\numbers.txt');

Writeln ('чтение из файла. Подождите...');

s:=0;

k:=0;

sa:=0;

assign (FL, 'c:\numbers.txt');

Reset (FL);

While not EOF (FL) **do**

Begin

Readln (FL,n);

s:=s+n;

k:+k+1;

End;

Close(FL);

sa:=s/k;

Writeln ('Среднее арифметическое:' sa: 9:2);

Readln;

End.

10. Самостоятельная работа учащихся (по времени)



1. Написать программу, которая записывает в текстовый файл **Phone.txt**, находящийся на диске **C** фамилию, имя и номер телефона вашего товарища. Рекомендуемый вид экрана приведен ниже:

Телефонный справочник учащихся 9 а класса.

1. Алтер Денис 3-73-72
2. Бутвин Ростислав 4-10-14
3. Греку Дмитрий 11-41-3
4. Деревщикова Дарья 5-45-43
5. Иванюта Настя 3-16-38
6. Инжебейкин Даниил 3-18-85

2. Сохранить программу в своей папке под именем **Phone_9a.pas**.

3. Используя процедуру Append, измените программу для дополнения текстового файла новыми номерами телефонов.

10. Домашнее задание. (Написать программу, выводящую на экран наибольшее и наименьшее число из текстового файла **numbers.txt**.)

11. Подведение итогов урока.

