

Файлы

начало



Вникнем – что такое файлы?

В Паскале существует множество различных типов данных.

Они делятся на простые (состоящие из одного данного) и на сложные (состоящие из набора данных) типы.



До сегодняшнего дня ты все время при запуске программы вводил по новому все данные потому они упорно не желали оставаться в памяти компьютера.

А о том чтобы передать данные из одной программы в другую и речи идти не могло!



Но теперь у тебя появился новый друг и спутник жизни – Файлы.

Как ты наверное уже догадался с помощью файлов ты сможешь:

- Использовать сохраненные данные даже после взрыва вселенной
- Передавать данные из одной программы в другую

прямо как wifi в макдональдсе передает сообщения vk в твой телефон

Стоп! На этом слайде надо задержаться :)

Файл – это упорядоченный поименованный набор элементов одного типа на внешнем носителе.

Определение выше я советую прочитать еще раз, осмыслить и вы зубрить потому что у тебя всю жизнь будут спрашивать его

Будут будить по ночам и спрашивать

Не ответишь - расстрел

(из личного опыта)

Теперь немного о элементах файла

Прежде всего элемент файла в программе это переменная определенного типа

И как любая переменная она должна быть объявлена

Иначе последствия будут катастрофическими

Будет ошибка и насмешки друзей над отличной программой неспособной скомпилироваться :)

Как происходит получение файла в программе

Постарайся уловить мысль в этой под теме поскольку это очень важно

Программа не загружает сразу полностью файл

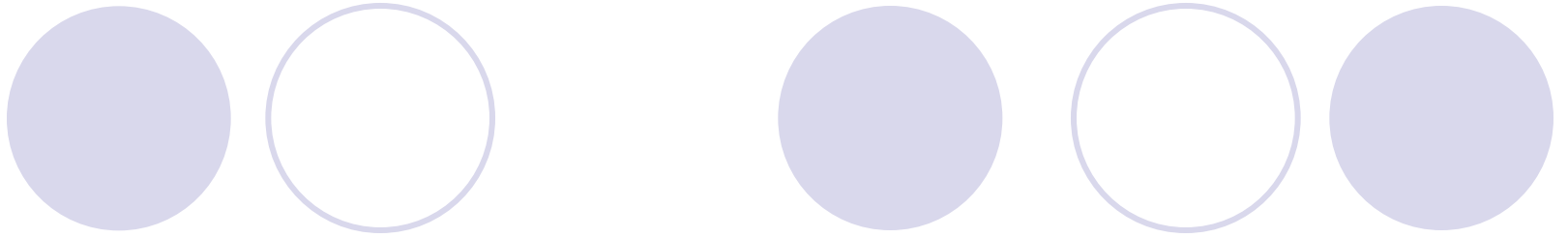
Она загружает его поэлементно

Начинает она разумеется с первого элемента и идет по порядку до последнего

На том элементе который сейчас программа будет считывать стоит так называемое окошко (понять что такое окошко и как оно переключается безумно важно так что вникай)

Есть специальные операторы которые сообщают программе что пора считать новый элемент с того места где стоит окошко (о них позже)

И после применения таких операторов программа переводит окошко на следующий элемент



Все элементы из файла считываются в переменную
Самую обычную переменную, только важно чтобы ее
тип соответствовал типу элементов файла

Объявление файлов

Для того чтобы программа смогла использовать файл надо объявить его и привязать то имя к имени файла на диске

Давайте разберемся с объявлением файла (сразу после этого с привязкой)

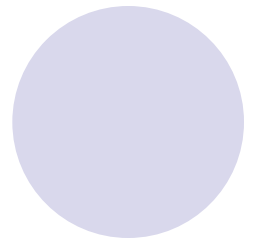
При объявлении файла надо сообщить программе какого типа у нас файл

Есть несколько способов это сделать:

- Тыкнуть носом паскаль в тип
Тогда файл будет считаться типизированным
- Послать паскаль самому определять тип файла (фигово работает с записями)
Тогда файл будет считаться нетипизированным
- Намекнуть что файл текстовый
Тогда файл будет считаться текстовым

Мне больше нравится первый вариант

Так я точно уверен что паскаль не будет меня подкалывать)



Давай объявим разные типы файлов

- Типизированные файлы

```
var f:file of integer;
```

f – это имя файла в программе, ну и ты видишь что здесь у нас файл будет типа integer

Вместо integer можно написать любой другой тип кроме файлового (можешь даже свой тип там указывать)

- Нетипизированные файлы

```
var f:file;
```

- Текстовые файлы

```
var f:text;
```

Привязка, открытие, закрытие

Теперь как я и обещал привязка имя файла на диске к имени файла в программе

Это сделать очень легко

`assing(f, 'file.txt');` где `f` – ранее объявленное нами имя файла, а `file.txt` его название на диске

Теперь чтобы считать информацию из файла его нужно открыть

В паскале есть два режима открытия файла:

- Для ввода – файл полностью обнуляется
`rewrite(f);` где `f` – ранее объявленное нами имя файла
- Для чтения – файл не обнуляется, но ты можешь заменить любой элемент в файле (об этом далее)

`reset(f);` где `f` – ранее объявленное нами имя файла

Так же всегда после работы с файлами их необходимо закрывать чтобы не засорять память может вылететь ошибка

`close(f);` где `f` – угадай что

Считывание и запись в файл

Ну конечно же между открытием и закрытием файла должны быть чтение и запись!

Кстати, сейчас надо вспомнить про то самое окошко о котором я рассказывал)

Чтение элемента файла:

`read(f,info);` где `f` – мы уже выучили, `info` – переменная в которую присваивается элемент файла
оператор работает только если мы открываем файл для чтения

Запись элемента файла:

`write(f,info);` где `f` - ... , а `info` – то что мы хотим записать в файл
оператор работает как в режиме чтения, так и в режиме записи

И не забывайте! Использовали мы один из этих операторов и окошко само переключилось на следующий элемент файла. 70% студентов забывают об этом по началу

Как пройтись по файлу?

Как ты уже запомнил мы идем поэлементно по файлу

И раз надо пройтись поэлементно значит нужен цикл

Но как выйти из цикла как только файл закончится?

Для этого существует `eof(f)` ну это просто дебилизм будет писать что такое f :)

Чтобы активировать это фигню надо написать в самом начале программы `checkeof:=true;`

Пока файл не закончится `eof(f)` будет возвращать `false`

Как только закончится вернет `true`

Вот пример цикла прохода по файлу

```
while not(eof(f)) do begin
    read(f,info);
end;
```

Стандартные файлы в Паскале

А ты знаешь что почти во всех своих программах ты уже работал с файлами?)

Есть так называемые стандартные файлы в паскале

Они служат для ввода/вывода информации

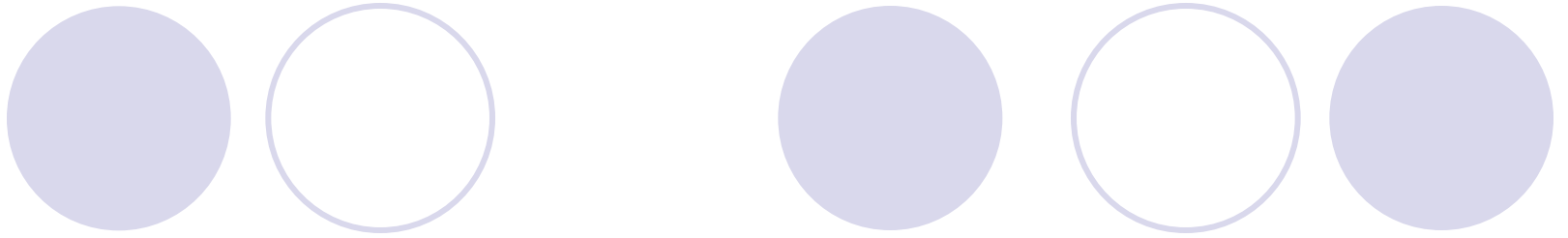
Ведь не случайно `read(info)` и `read(f,info)`, `write(info)` и `write(f,info)` похожи по написанию

Считывая переменную `info` программа по умолчанию устанавливает работу с файлом ввода/вывода (при условии если ты сам не указал файл с которым работать)

А если все что мы набираем на клавиатуре считается выходными данными из файл значит можно и определить конец выходного потока

```
checkeof:=true;
while not(eof) do begin
    read(info);
end;
```

И пока мы не нажмем Esc мы будем снова и снова вводить переменную `info`



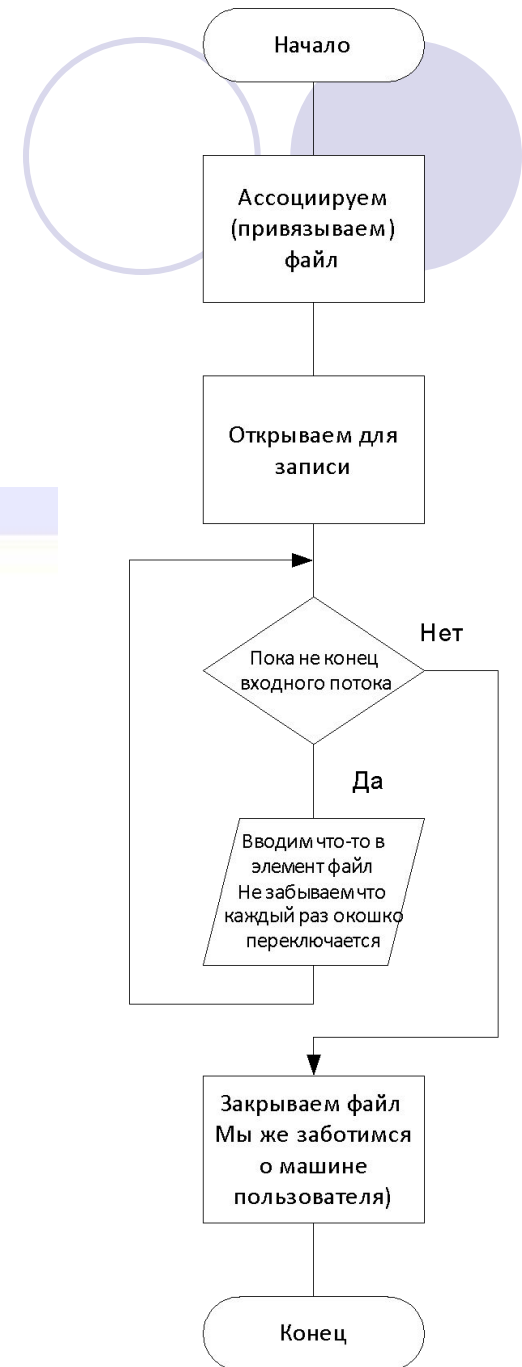
Теперь сделай перерыв
Отдохни, подумай, перевари прочитанное

Если что не понятно спроси у преподавателя
или еще раз перечитай предыдущие слайды

А теперь к практике

Запись в файл

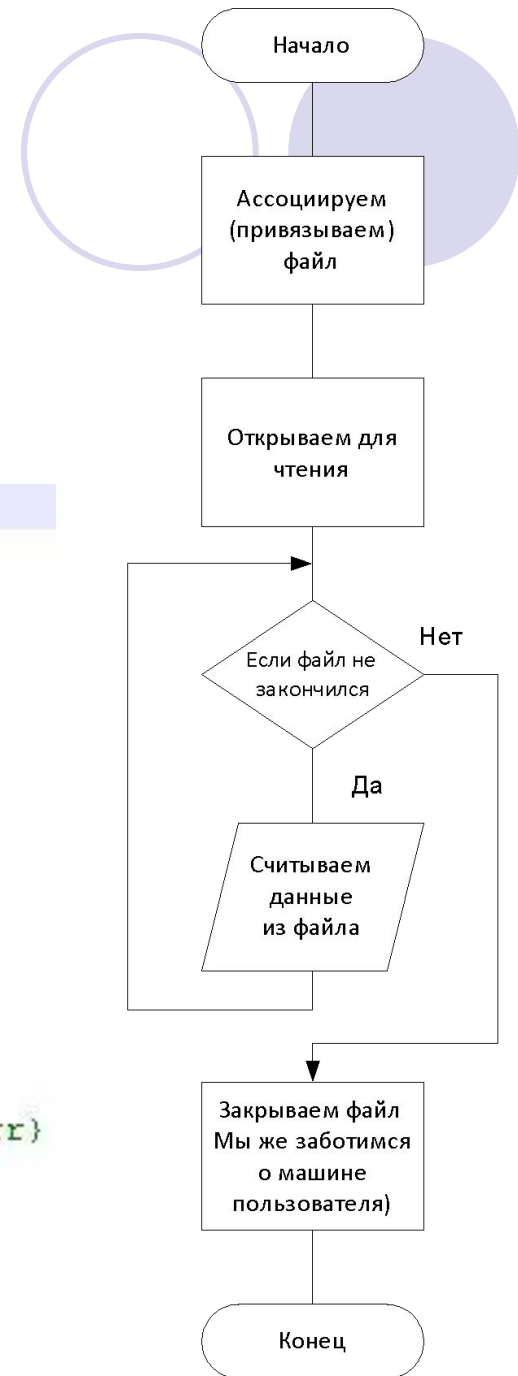
```
1  program la_pr001;  
2  uses crt;  
3  var f:file of string; info:string;  
4  begin  
5    checkeof:=true;  
6    assign(f,'file.txt'); {привязываем файл}  
7    rewrite(f); {открываем файл для записи}  
8  while not(eof) do begin  
9      read(info); {запрашиваем у пользователя info}  
10     write(f,info); {записываем info в файл}  
11 end;  
12 close(f); {закрываем файл}  
13 end.
```



А теперь к практике

Считывание из файла

```
1  program la_pr001;
2  uses crt;
3  type t=record
4      fio:string;
5      date:integer;
6  end;
7  var q:file of t; arr:t;
8  begin
9      checkeof:=true;
10     assign(q,'file.txt'); {привязываем файл}
11     reset(q); {открываем файл для чтения}
12     while not(eof(q)) do begin
13         read(q,arr); {считываем arr из файла}
14         writeln(arr.fio,' ',arr.date); {красиво выводим arr}
15     end;
16     close(q); {закрываем файл}
17 end.
```





Ну вот и все
Ты освоил основы файлов и теперь
напишешь крутую прогу ;)

Удачи!

С тобой был
LIGHT ALEX