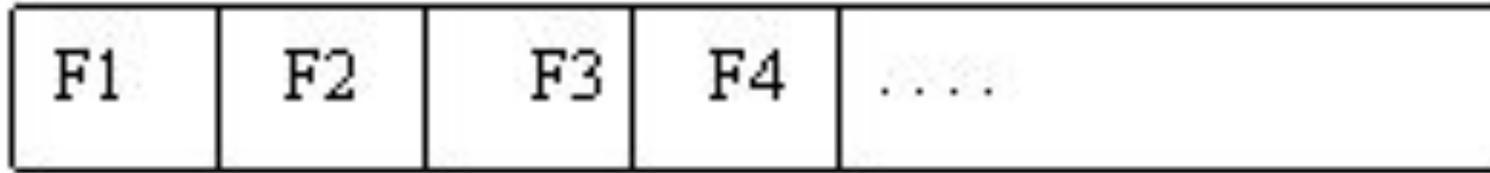


ФАЙЛЫ В ЯЗЫКЕ ПАСКАЛЬ

Элементы файла записываются на эту ленту

F



где F – имя файла, а F1, F2, F3, F4

–

его элементы.

Файловый тип или переменную файлового типа в Паскале

можно

задать одним из трех способов:

Type

<имя_ф_типа>=file of<тип_элементов>;

<имя_ф_типа>=text;

<имя_ф_типа>=file;

Здесь <имя_ф_типа> – имя файлового типа (правильный идентификатор); File, of –

зарезервированные слова (файл, из); <тип_элементов> – любой тип Паскаля, кроме файлов.

Пример описания файлового типа в Паскале

Type

```
Product= record
```

```
  Name: string;
```

```
  Code: word;
```

```
End;
```

```
Text80= file of string[80];
```

Var

```
F1: file of char;
```

```
F2: text;
```

```
F3: file;
```

```
F4: Text80;
```

```
F5: file of Product;
```

В зависимости от способа объявления можно выделить

три вида файлов Паскаля:

- типизированные файлы Паскаля(задаются предложением file of.);
- текстовые файлы Паскаля(определяются типом text);
- нетипизированные файлы Паскаля(определяются типом file).

Пример процедуры Assign в Паскале:

```
Assign (chf, 'G:\Home\ Student\ Lang\ Pascal\ primer.dat');
```

Основные процедуры и функции для работы с файлами

До начала работы с файлами в Паскале необходимо установить

связь между

файловой переменной и именем физического дискового файла:

```
Assign(<файловая_переменная>,  
<имя_дискового_файла>)
```

После окончания работы с файлами на Паскале, они
должны
быть закрыты.

Close(<список файловых переменных>);

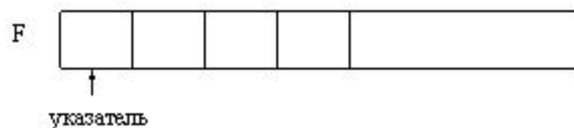
Подготовка к записи в файл Паскаля

Rewrite(<имя_ф_переменной>);

Процедура **Rewrite(f)** (где f – имя файловой переменной)

устанавливает файл с именем f в начальное состояние режима записи, в результате чего указатель устанавливается на первую позицию файла. Если ранее в этот файл были записаны какие-либо элементы, то они становятся недоступными.

Результат выполнения процедуры **rewrite(f)**; выглядит следующим образом:



Состояние файла *f* после выполнения процедуры

• Запись в файл Паскаля

Write(<имя_ф_переменной>, <список записи>);

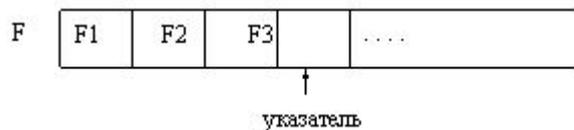
При выполнении процедуры **write(f, x)** в позицию, на которую показывает указатель, записывается очередная компонента, после чего указатель смещается на следующую позицию.

Естественно,

тип выражения *x* должен совпадать с типом компонент файла.

Результат действия процедуры **write(f, x)** можно изобразить так:

Состояние файла *f* до выполнения процедуры



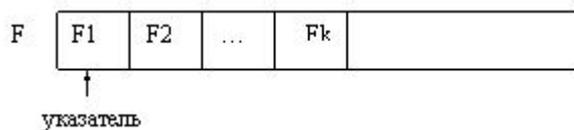
Подготовка файла к чтению Паскаля

Reset(<имя_ф_переменной>);

Эта процедура ищет на диске уже существующий файл и переводит его в режим чтения, устанавливая указатель на первую позицию файла.

Результат выполнения этой процедуры можно изобразить следующим образом:

Если происходит попытка открыть для чтения не существующий еще на диске файл, то возникает ошибка ввода/вывода, и выполнение программы будет прервано.



• Чтение из файла в Паскале

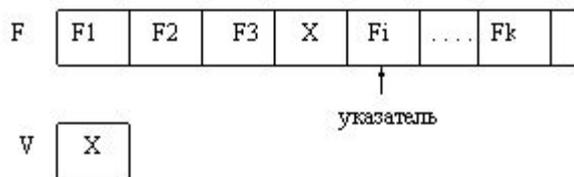
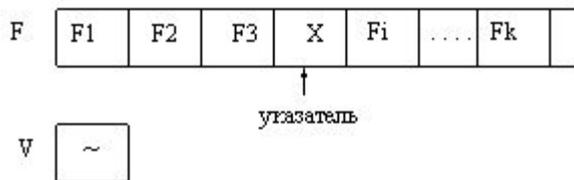
Read(<имя_ф_переменной>, <список переменных>);

Рассмотрим результат действия процедуры **read(f, v):**

Состояние файла **f** и переменной **v** до выполнения процедуры:

Состояние файла **f** и переменной **v** после выполнения процедуры:

Для типизированных файлов при



- Функция определения достижения конца файла в Паскале

Eof(<имя_ф_переменной>);

Название этой функции является сложносокращенным словом от end of file.

- Изменение имени файла в Паскале

Rename(<имя_ф_переменной>, <новое_имя_файла>);

Здесь новое_имя_файла – строковое выражение, содержащее новое имя файла, возможно с указанием пути доступа к нему.

- Уничтожение файла в Паскале

Erase(<имя_ф_переменной>);

Перед выполнением этой процедуры необходимо

закрыть файл, если он ранее был открыт.

Уничтожение части файла от текущей позиции указателя до конца в Паскале

Truncate(<имя_ф_переменной>);

Файл Паскаля может быть открыт для добавления

записей в конец файла

Append(<имя_ф_переменной>);