

ФАЙЛОВЫЕ ТИПЫ ДАННЫХ. ВИДЫ ФАЙЛОВ

КОГДА И ЗАЧЕМ НУЖНО ИСПОЛЬЗОВАТЬ ФАЙЛЫ?

- При составлении и отладки программы, требующей ввода большого количества данных.
- При необходимости сохранения промежуточных данных при длительной работе программы.
- Когда данные востребованы множеством разных программ.

ВИДЫ ФАЙЛОВ

- ⦿ Типизированный файл
- ⦿ Нетипизированный файл
- ⦿ Текстовый файл

ОБЩИЕ ОСОБЕННОСТИ ВСЕХ ВИДОВ ФАЙЛОВ

- Записи (данные) располагаются последовательно друг за другом как элементы одномерного массива; отличие в том, что размеры файла (количество записей) могут меняться.
- Способы обращения к записи другой, чем к элементам массива.
- Количество записей в файле не ограничивается ничем.
- Длиной файла называется число находящихся в нем записей.

Пустым файлом называется файл, не содержащий запись. Длина пустого файла равна 0.

ТИПИЗИРОВАННЫЙ ФАЙЛ

- Типизированный файл - это последовательность записей одного типа.
- Типизированные файлы – это файлы прямого доступа. Доступ к записям файла осуществляется по их порядковым номерам. Записи нумеруются, начиная с 0.
- После открытия файла указатель стоит в его начале на нулевой записи. После каждого чтения или записи указатель сдвигается к следующему компоненту.

НЕТИПИЗИРОВАННЫЙ ФАЙЛ

- Нетипизированный файл - это файл, состоящий из последовательности одинаковых по длине блоков.
- В блоках могут находиться записи разных длин (и типов).
- Блок по умолчанию имеет размер 128 байт. Но его можно изменить, задав другую длину.
- В блоке может находиться только 1 запись. Она может занимать весь блок или ничтожно малую его часть.
- К любому блоку можно обратиться напрямую по номеру. Нетипизированный файл также является файлом прямого доступа.

ТЕКСТОВЫЙ ФАЙЛ

- Представляет собой последовательность символьных, текстовых записей.
- Каждый символ в записи занимает 1 байт.
- Позволяет размещать записи разной длины.
- Строки записей между собой разделяются маркерами конца строки. Весь файл заканчивается маркером конца файла.
- Текстовый файл – это файл последовательного доступа. Чтобы прочитать n запись, надо прочитать от 0 до $n-1$ записей.

ТЕХНОЛОГИЯ СОЗДАНИЯ ФАЙЛА

(ДЛЯ ВСЕХ ТИПОВ ФАЙЛОВ)

Файловая переменная (ФП) является посредником между программой и файлом.

ФП предназначена для выполнения следующих действий:

- ⦿ Задание типа файла
- ⦿ Осуществление связи с файлом
- ⦿ Как необходимый атрибут для работы с файлами

ФП описывается в разделе описания. Там она и указывает тип файла.

ФОРМАТ ОПИСАНИЯ ФП

- **Var <Имя ФП>: File Of <Тип>;** - задание типизированного файла
- **Var <Имя ФП>: File ;** - задание нетипизированного файла
- **Var <Имя ФП>: Text;** - задание текстового файла

ОСНОВНЫЕ ПРОЦЕДУРЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- **Assign** (<Имя ФП>, 'Путь и имя файла'); - устанавливает связь между файловой переменной и самим файлом.
- **ReSet** (<Имя ФП>); - открыть для чтения. Если файла не существует, то будет выведена ошибка.
- **ReWrite** (<Имя ФП>); - создать или перезаписать. Если файла не существует, то он будет создан.
- **Append** (<Имя ФП>); - создать или открыть и дописать в конец файла. Если файла не существует, то он будет создан.
- **Close** (<Имя ФП>); - сохранить и закрыть файл.




ТЕКСТОВЫЕ ФАЙЛЫ. ПРОЦЕДУРЫ ЧТЕНИЯ И ЗАПИСИ

Пример. Написать программу, которая считывает 2 числа из текстового файла, находит их сумму и результат сложения выводит в другой текстовый файл.

```
Var A, B, Sum: Integer;  
Begin  
  ReadLn(A,B);  
  Sum:=A+B;  
  Write('A+B=',Sum);  
  ReadLn;  
End.
```

```
Var A, B, Sum: Integer;  
      F1, F2: Text;  
Begin  
  Assign(F1,'input.txt');  
  Assign(F2,'res.txt');  
  Reset(F1);  
  Read(F1, A, B);  
  Sum:=A+B;  
  Append(F2);  
  WriteLn(F2, Sum);  
  Close(F1);  
  Close(F2);  
  ReadLn;  
End.
```

КАК ИСПОЛЬЗОВАТЬ ИЛИ ИЗМЕНИТЬ СОДЕРЖИМОЕ ФАЙЛА?

- ⦿ **Read(f,a,b);** — считывает из файла две переменные `a` и `b`. После выполнения этой процедуры указатель в файле передвинется за переменную `b`;
- ⦿ **Readln(f,a,b,c);** — считывает из файла три переменные `a`, `b` и `c`, а затем переводит указатель (курсор) на начало следующей строки; если кроме уже считанных переменных в строке содержалось еще что-то, то оно будет проигнорировано.
- ⦿ **Write(f,a,b,c);** — записывает в файл переменные `a`, `b` и `c`;
- ⦿ **Writeln(f,a,b);** — записывает в файл переменные `a` и `b`, а затем записывает туда же символ "конец строки". 

Пример. Написать программу, которая создает текстовый файл, содержащий 2 числа, вводимых с клавиатуры.

```
Var A, B: Integer;
```

```
    F2: Text;
```

```
Begin
```

```
Assign(F2,'res.txt');
```

```
Rewrite(F2);
```

```
WriteLn('Введите A ');
```

```
Read(A);
```

```
WriteLn('Введите B');
```

```
Read(B);
```

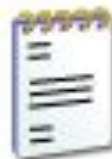
```
Write(F2, A, ' ', B);
```

```
Close(F2);
```

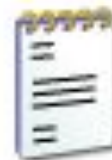
```
End.
```



sum.pas
Pascal Program
1 КБ



input.txt
Текстовый документ
1 КБ



res.txt
Текстовый документ
0 КБ



- Что такое файл?
- Зачем нужно использовать файлы?
- Верно ли, что у файла количество элементов фиксируется в момент распределения памяти?
- Какие виды файлов вы знаете?
- В чем особенность типизированных видов файлов?
- В чем особенность нетипизированных видов файлов?
- В чем особенность текстовых видов файлов?
- Какие виды файлов являются файлами прямого доступа (последовательного доступа)? Что это означает?
- Какие основные процедуры для работы с файлами вы знаете ?

- Объясните разницу между следующими процедурами ввода:

Read(<элемент>) и Read(<имя ФП, элемент>);

Read(<имя ФП, элемент>) и ReadLn(<имя ФП, элемент>)

- Объясните разницу между следующими процедурами вывода:

Write(<элемент>) и Write(<имя ФП, элемент>);

Write(<имя ФП, элемент>) и WriteLn(<имя ФП, элемент>)

ПРИЗНАКИ КОНЦА СТРОКИ И КОНЦА ФАЙЛА

- **Функция *Eof*** возвращает значение *истина*, если указатель находится за последним символом файла, или если файл пуст, и значение *ложь* в противном случае.
- **Функция *Eoln*** возвращает значение *истина*, если указатель находится перед символом конца строки, или если достигнут конец файла, т.е. значение функции *Eof* истинно, и значение *ложь* в противном случае.

Эти функции лучше всего применять в цикле.

```
Var f: Text;  
    N, K, I: Integer;  
Begin Assign(f, 'dan.txt');  
    Reset(f);  
    I:=0;  
    While not EOF(f) do  
        Begin I:=I+1;  
            K:=0;  
            While not EOLN(f) do  
                Begin  
                    Read(f, N);  
                    If N mod 2=0 Then K:=K+1;  
                End;  
                Writeln ('Количество четных в ', I, ' строке=',K);  
                Readln(f);  
            End;  
        End;  
    Close(f);  
    Readln;  
End.
```

Пример. Дан текстовый файл, в строках числа. Подсчитать количество четных чисел в каждой строке.

ЗАДАЧИ

Дан текстовый файл, в строках целые числа.
Найдите в каждой строке:

1. Сумму положительных (отрицательных, четных, нечетных, кратных трем).
2. Среднеарифметическое.
3. Среднеарифметическое четных.
4. Максимальный (минимальный) элемент.
5. Максимальный элемент и его номер.
6. Получить 2 новых файла. В один из них поместить четные элементы файла, в другой – нечетные.

ЗАДАЧИ

- Создать текстовый файл `chisla.txt` содержащий все четные двузначные числа.
- Создать текстовый файл `proba.txt`, содержащий все трехзначные числа, кратные 125.

ЗАДАЧИ

Даны три целых числа. Определить, имеется ли среди них хотя бы одна пара равных между собой чисел.

- **Формат входных данных (1-ый текстовый файл):**

Входной файл содержит три целых числа через пробел.

- **Формат выходных данных (2-ой текстовый файл):**

Выведите 'YES' если это так, и 'NO' в противном случае.

- **Примеры:**

Вход

544 935 911

Выход

NO

Посимвольная обработка ТЕКСТОВЫХ файлов

```
Var F: Text;  
    C: Char;  
Begin  
Assign(F, ' C:\PABCWork\dan.txt' );  
Reset(F);  
    While not EOF(F) do  
        Begin  
            While not EOLN(F) do  
                Begin  
                    Read(F, C);  
                    Write(C);  
                End;  
            ReadLn(F);  
            WriteLn;  
        End;  
    Close(F);  
End.
```

Построчная обработка ТЕКСТОВЫХ файлов

```
Var F: Text;  
    S: String;  
Begin  
Assign(F, 'C:\PABCWork\dan.txt');  
Reset(F);  
While not EOF(F) do  
    Begin  
        ReadLn(F, S);  
        WriteLn(S);  
    End;  
Close(F);  
End.
```

ТЕКСТОВЫЕ ФАЙЛЫ. ПРОЦЕДУРЫ ЧТЕНИЯ И ЗАПИСИ

1. Какие виды файлов вы знаете?
2. Какие файлы называются текстовыми?
3. В чем особенность текстовых видов файлов?
4. Какими текстовые файлы являются по методу доступа?
5. Для чего предназначена ФП?
6. Сколько ФП следует объявлять в программе?
7. Какая процедура устанавливает связь ФП с файлом?
8. Какие существуют процедуры для открытия текстового файла? Охарактеризуйте каждую из них.
9. С помощью каких процедур осуществляется чтение данных из файла?
10. Объясните разницу между следующими процедурами:
 - ⦿ **Read**(<элемент>) и **Read**(<имя ФП, элемент>);
 - ⦿ **Read**(<имя ФП, элемент>) и **ReadLn**(<имя ФП, элемент>)

11. С помощью каких процедур осуществляется запись данных в файл?

12. Объясните разницу между следующими процедурами:

- ⊙ **Write**(<элемент>) и **Write**(<имя ФП, элемент>);
- ⊙ **Write**(<имя ФП, элемент>) и **WriteLn**(<имя ФП, элемент>)

13. Какая функция служит для определения конца файла, конца строки?

14. Что может произойти, если после работы с файлом, его не закрыть?

ЗАДАНИЕ: РАССКАЗАТЬ, ЧТО ДЕЛАЕТ И КАК РАБОТАЕТ ПРОГРАММА

```
Var A, I: Integer;  
    F: Text;  
Begin  
    Assign (F, 'data.txt' );  
    ReWrite(F);  
    For I:=1 to 10 do  
        Begin  
            Read(A);  
            Write(F, A)  
        End;  
    Close(F);  
End.
```

ЗАДАНИЕ: РАСКАЗАТЬ, ЧТО ДЕЛАЕТ И КАК РАБОТАЕТ ПРОГРАММА

```
Var A, S, I: Integer;  
      F: Text;  
Begin  
  Assign(F, 'primer.txt ');  
  Reset(F);  
  S:=0;  
  While not Eof(F) do  
    Begin  
      Read (F, A);  
      S:=S+A;  
    End;  
  Close(F);  
  WriteLn(' Сумма= ', S);  
End.
```

ЗАДАЧИ

- Дан файл f , компонентами которого являются целые числа (создать при помощи текстового редактора). Переписать все чётные числа в файл $f1$, нечётные в файл $f2$.
- Имеются два файла с числами (создать при помощи текстового редактора). Поменять местами их содержимое (использовать вспомогательный файл).

ЗАДАЧИ

Дан текстовый файл, содержащий три целых числа, записанных через пробел. Написать программу, которая:

- 1) считывает числа из файла и выводит их на экран
- 2) определяет, имеется ли среди них хотя бы одна пара равных между собой чисел. Выведите 'YES' если это так, и 'NO' в противном случае.

◉ **Пример:**

544 935 911

NO

ЗАДАЧИ

- ⦿ Создать текстовый файл `chisla.txt` содержащий все четные двузначные числа.
- ⦿ Создать текстовый файл `proba.txt`, содержащий все трехзначные числа, кратные 125.

СПАСИБО ЗА ВНИМАНИЕ!