Лабораторная работа №18

Тема. Работа с файлом произвольного доступа

Цель работы. Закрепление на практике основных принципов обработки текстовых файлов, процедур и функций их обработки.

Теоретическая часть

Более характерным для Pascal являются типизированные файлы, или файлы произвольного доступа. Основным свойством этих файлов является то, что их структура данных представляет собой последовательность компонентов одного типа. Описывают подобный файл словосочетанием file of с последующим указанием типа компонентов файла, число которых (длина файла) не фиксируется:

var имя_файла: file of тип_компонентов

Поскольку известен тип элементов файла, а следовательно, и объем памяти, отводимой под каждый из них, можно рассчитать позицию каждого из элементов внутри файла. Это позволяет организовать непосредственный доступ к любому элементу типизированного файла. Так, например, рассмотрим описание:

Var FileInt: file of Integer

В этом описании указано, что элементами файла являются данные типа Integer. При этом отпадает необходимость в специальном разделении элементов файла, как это делалось в текстовых файлах. Также возможен произвольный доступ к элементам данных (этим типизированный файл несколько напоминает одномерный массив).

Чтобы можно было работать с типизированным файлом, необходимо, как и для текстовых файлов, сначала связать имя файловой переменной с внешним именем файла (оператор Assign). Затем нужно открыть его (используются операторы Reset и Rewrite, но не Append). Операторы Reset и Rewrite открывают файл и для чтения, и для записи (а не только для чтения или только для записи, как при использовании текстовых файлов). Отличие их в том, что оператор Reset открывает только существующий файл (если такого файла нет, будет сгенерирована ошибка времени выполнения). С другой стороны, оператор Rewrite создает новый файл (если файл с таким именем уже имеется, то он будет уничтожен и создан заново). При открытии файла с ним связывается текущий указатель файла, который позиционируется на его первый элемент. Оперировать можно только тем элементом файла, на который ссылается указатель файла. При чтении или записи элемента файла происходит автоматическое перемещение указателя на следующий элемент. Чтение из типизированного файла производится оператором Read (но не ReadLn), а запись в него — оператором Write (но не WriteLn). Однако следует помнить, что в списке вывода оператора Write могут быть только переменные. Типы элементов файла и типы переменных в списках ввода-вывода должны быть согласуемы по присваиванию. Элементами типизированных файлов могут быть числовые, символьные, булевы, строковые значения, массивы, записи, но не файлы или структуры с файловыми элементами.

Узнать количество элементов типизированного файла (размер файла) можно с помощью функции FileSize, для которой используется следующий синтаксис: FileSize(имя_файла)

Элементы типизированного файла нумеруются с нуля (порядковый номер последнего элемента файла на единицу меньше размера файла). Чтобы узнать, на каком элементе располагается указатель файла, используют функцию FilePos: FilePos(имя_файла)

Текущим положением указателя можно управлять, для чего служит процедура Seek, которая использует следующий синтаксис: Seek(имя_файла, номер_элемента)

Второй параметр (тип LongInt) задает номер элемента (отсчет от 0), на который должен переместиться указатель файла. Рассмотрим несколько примеров.

Перейти к пятому (фактически шестому) элементу файла f: Seek(f, 5);

Перейти к предыдущему элементу: Seek(f, FilePos(f)-1);

Перейти в конец файла: Seek(f, FileSize(f)-1);

Как и для текстовых файлов, можно использовать функцию Eof(имя_файла), которая возвращает значение True, если текущий указатель расположен на признаке конца файла (т. е. при выполнения равенства FilePos(имя_файла) = FileSize(имя_файла)).

Процедура Seek и функция FilePos и FileSize позволяют легко осуществлять коррекцию элементов типизированного файла, имя которого указано в качестве е параметра, начиная с элемента, на котором расположен указатель. Однако уничтожить элемент внутри файла нельзя, для этого файл должен быть перезаписан.

Процедура Truncate(<имя файловой переменной>) удаляет все компоненты в файле, начиная с того над которым находиться указатель.

Текстовые файлы могут быть созданы текстовым редактором. Однако типизированные файлы создаются в результате работы какой-либо программы.

Индивидуальные задания

Задание. Составить программу в соответствии с вариантом. Проверить работу программы на компьютере; оформить отчет, содержащий блок-схему алгоритма работы программы, листинг программы, результаты работы программы.

Вар	Задание
1.	Создать числовой файл f . Получить файл g , в который записать сначала положительные, затем отрицательные и, наконец, нулевые компоненты файла f . Далее в файле g заменить точки стыковки последовательностей увеличением на 100 элементов стоящих слева и справа от этих точек (мест где положительные сменяются с отрицательными и отрицательные с нулевыми). Файлы f и g вывести до и после преобразования.
2.	Создать файл f целого типа. Получить два файла: f1, f2 . В файл f1 последовательно писать четные числа, в файл f2 – все нечетные числа файла f . В файле f2 элементы кратные трём или пяти уменьшить на единицу. Файлы f, f1,f2 распечатать.
3.	Создать числовой файл f . Найти сумму положительных компонент файла, расположенных до максимальной компоненты. Найденной суммой заменить отрицательные компоненты, строящие после максимума. Файл и сумму распечатать (как до, так и после преобразования).
4.	Создать числовой файл f . Найти в файле сумму и произведение максимальной и минимальной компонент. Выяснить что больше. Найденным значением заменить первую и предпоследнюю компоненты файла, если сумма окажется больше. Заменить все отрицательные компоненты, если большим будет произведение. Сумму, произведение, исходный файл и файл после замены распечатать.
5.	Создать файл f целого типа. Вычислить сумму четных компонент файла. Суммой заменить вторую отрицательную компоненту файла, или вывести сообщение о невозможности таковой замены. Исходный и файл после замены распечатать.
6.	Создать файл f вещественных чисел. Найти максимальную компоненту и дописать её в конец файла исходную максимальную компоненту уменьшить в два раза. Далее сформировать файл h , записав в него каждую вторую компоненту файла f . Файлы распечатать.

Вар	Задание
7.	Сформировать файл f целого типа. Найти и распечатать произведение максимальной и минимальной компонент файла. Произведением заменить все нулевые компоненты. Если нулевых компонент в файле нет, то поменять в нём местами максимум и минимум. Исходный и полученный файлы распечатать.
8.	Создать файл f вещественных чисел. Найти SrA – среднее арифметическое элементов расположенных левее максимума. Все компоненты файла, начиная с максимальной и по последнюю включительно, увеличить на число SrA и просуммировать. Суммой заменить максимальную компоненту файла.
9.	Ввести файл f , посчитать k – количество положительных компонент файла f , которые расположены на тех позициях, номера которых кратны трем. Далее все компоненты от k -й и до максимальной заменить единицами.
10.	Сформировать файл f , компонентами которого являются действительные числа. Найти и распечатать сумму компонент файла, произведение отрицательных компонент файла и среднее арифметическое компонент файла. Эти три значения вписать в файл справа от его максимума. Файл вывести до и после преобразования.
11.	Сформировать файл f . Определить сумму и произведение наибольшего и наименьшего из значений компонент. Заменить значением суммы первую и последнюю компоненты, а на значение произведения – увеличить каждую третью компоненту файла. Файл вывести до и после преобразования.
12.	Ввести файл f1 . Записать в файл f2 компоненты файла f1 в обратном порядке. В каждом из файлов элемент, стоящий справа от максимума продублировать значением этого максимума. Файлы вывести до и после преобразования. (При решении задачи вспомогательных массивов и файлов не использовать)
13.	Ввести файлы f1 и f2 . Переписать с сохранением порядка следования компоненты файла f1 в файл f2 , а компоненты файла f2 в файл f1 . При решении допускается использование вспомогательного файла h . Файлы вывести до и после преобразования с возведением в квадрат их максимальных компонент.
14.	Сформировать два файла целых чисел одинаковой длины f1 и f2 . Затем необходимо слить их в файл f3 таким образом, чтобы сначала шла компонента из f1 , потом из f2 и т. д. В файле f3 поменять местами минимальную и предпоследние компоненты. Файлы вывести до и после преобразования.
15.	Сформировать файл f целых чисел. Произвести «зеркалирование» файла, т.е. увеличение его длины в два раза методом добавления в конец собственных элементов в обратном порядке. В конец файла дописать произведение его нечётных и сумму чётных элементов. (При решении вспомогательных файлов и массивов не использовать)
16.	Задан числовой файл f . Найти максимум среди компонент, расположенных на четных позициях файла и минимум среди компонент на нечётных позициях. Все нулевые компоненты заменить значением максимума, а значение минимума дописать в конце файла пять раз. Файл до и после преобразований распечатать.
17.	Задан числовой файл f . Найти сумму компонент, стоящих между минимальной и максимальной. Заменить этой суммой все элементы из которых она сформирована (элементы между максимумом и минимумом). Файл вывести как до, так и после преобразования.
18.	Дан целочисленный файл f . Все его отрицательные компоненты увеличить в два раза, положительные уменьшить в три раза, на место нулевых компонент записать их количество. Файл f распечатать как до, так и после преобразования.

Вар	Задание
19.	Ввести файл f действительных чисел. Заменить в нем максимальный и минимальный компоненты на значение среднего арифметического всех положительных компонент файла. Выдать на печать файл до преобразования и после. В случае невозможности преобразования дать об этом сообщение.
20.	Ввести файл целых положительных чисел f1 . Компоненты файла f1 , меньшие числа SrA (среднего арифметического его элементов), записать в файл f2 . В файле f1 этим компонентам присвоить значение нуль. Выдать на печать файлы f1 и f2 как до, так и после преобразования.
21.	Ввести с клавиатуры файл f действительных чисел. Компоненты файла, стоящие между наибольшим и наименьшим компонентом, переставить в обратном порядке. Выдать на печать исходный и преобразованный файл.
22.	Ввести с клавиатуры файл f с числовыми компонентами и выдать его на печать. Определить среднее арифметическое компонентов файла и все компоненты, меньшие среднего арифметического, увеличить в два раза, а большие уменьшить в три раза. Преобразованный файл выдать на печать.
23.	Ввести файл f целых чисел. Определить в нем первую компоненту, кратную пяти, и поменять её местами с минимальной по модулю компонентой в файле. Выдать на печать файл до преобразования и после. Если преобразование невозможно, то выдать об этом сообщение.
24.	Ввести числовой файл f . Переставить в нём все компоненты, предшествующие минимальной в обратном порядке. Определить в изменённом файле на какой позиции находится максимум. Файл вывести как до, так и после преобразования.
25.	Ввести числовой файл f , продублировать все компоненты файла, расположенные между максимумом и минимумом, дописав их в конец файла. Исходный файл, максимум, минимум и файл после преобразования вывести
26.	Ввести целочисленный файл f . Найти значение k – количество чётных элементов в файле, далее найти сумму последних k элементов. Полученным значением заменить каждую вторую компоненту файла. Вывести на экран файл как до, так и после преобразования.
27.	Ввести с клавиатуры файл f действительных чисел и выдать его на печать. Определить среднее геометрическое положительных компонент файла, стоящих после третьей отрицательной компоненты, и записать его в конец файла. Выдать на печать файл после преобразования, значение вычисленного среднего геометрического или сообщение о невозможности преобразования.
28.	Ввести числовой файл f . Если первая компонента файла положительна, то домножить все его компоненты на квадрат минимального элемента, если отрицательна, то домножить все компоненты на квадрат максимума. В случае нулевой первой компоненты максимум и минимум продублировать, записав их значения в конец файла. Файл до и после преобразования отпечатать.
29.	В файле f заменить все нулевые компоненты значением суммы первой и последней компонент файла, все отрицательные компоненты уменьшить на значение суммы, а также найти среднее арифметическое положительных компонент, которым заменить сами положительные элементы. Файл вывести как до, так и после преобразования.
30.	Ввести файл f действительных чисел и выдать его на печать. Поменять местами наибольший по модулю и первый компоненты файла. Преобразовать его, не создавая нового файла, разделив компоненты на абсолютное значение наибольшего по модулю компонента.

Пример выполнения лабораторной работы

Задание. Ввести файл f действительных чисел. В файле элементы стоящие после второго нуля переставить в обратном порядке. Вывести файл до и после преобразования.

1. Листинг программы

```
var    a,b,i,left,right,I2_0,k:integer;
       f:file of integer;
begin
       assign(f,'f.dat');
       writeLn('введите первую компоненту файла f (признак конца ввода: "555")');
       //ввод файла
       rewrite(f);
       readLn(b);
       i:=1; //номер компоненты файла
       while b<>555 do
              begin
                     write(f,b);
                     inc(i);
                     writeLn('введите ',i,'-ую компоненту файла f');
                     readLn(b);
              end;
       //вывод файла до изменения
       reset(f);
       writeLn('файл f до изменения:');
       while not EOF(f) do
              begin
                     read(f,b);
                     write(b:5);
              end;
       writeLn;
       // поиск позиции второго нуля
       k:=0;
       reset(f);
       while (not EOF(f)) and (k<>2) do
              begin
                     read(f,b);
                     if b=0 then
                            begin
                            k:=k+1;
                            if k=2 then
                            I2_0:=filePos(f)-1;
                            end;
              end;
       // проверка возможности перестановки в файле
       if k<2 then  writeLn('в файле нет двух нулей')
              else   if I2_0=fileSize(f) then  writeLn('второй ноль в файле на последнем
       месте')
                                   else
                                         begin
                                         // перестановка элементов файла
```

```
                left:=I2_0+1;
                right:=fileSize(f)-1;
                while left<right do
                        begin
                                seek(f,left);
                                read(f,b);
                                seek(f,right);
                                read(f,a);
                                seek(f,left);
                                write(f,a);
                                seek(f,right);
                                write(f,b);
                                left:=left+1;
                                right:=right-1;
                        end;
                //вывод файла после изменения
                reset(f);
                writeLn('файл f после изменения:');
                while not EOF(f) do
                        begin
                                read(f,b);
                                write(b:5);
                        end;
                writeLn;
                end;
        close(f);
end.
```

2. Результаты работы программы

1) Входные данные:
файл f до изменения:
5 74 2 8 0 2 5 0 6 5 4 1 56
Выходные данные:
файл f после изменения:
5 74 2 8 0 2 5 0 56 1 4 5 6

2) Входные данные:
файл f до изменения:
5 74 2 8 0 2 5 3 6 5 4 1 56
Выходные данные:
в файле нет двух нулей

3) Входные данные:
файл f до изменения:
5 74 2 8 0 2 5 3 6 5 4 1 56 0
Выходные данные:
второй ноль в файле на последнем месте

Контрольные вопросы

1. Определение файла. Классификация файлов.
2. Процедуры и функции для обработки типизированных файлов.