**43-46 ОАП Пр10 Типизированные файлы**

**Цель:** Научиться составлять и решать задачи с использованием файловых типов.

**Теоретическая справка:**

Файл – это упорядоченная последовательность однотипных компонентов, расположенных на внешнем носителе. Файлы предназначены только для хранения информации, а обработка этой информации осуществляется программами.

Использование файлов целесообразно в случае:

- долговременного хранения данных;

- доступа различных программ к одним и тем же данным;

- обработки больших массивов данных, которые невозможно целиком разместить в оперативной памяти компьютера.

Наиболее часто используемыми файловыми типами явлиются:

- текстовые файлы(*text*);

- типизированные файлы (*file of тип*).

**Стандартные процедуры и функции**

*Assign(f, FileName)*

– связывает файловую переменную *f* с физическим файлом, полное имя которого задано в строке *FileName*. Установленная связь будет действовать до конца работы программы, или до тех пор, пока не будет сделано переназначение.

*Reset(f)*

– открывает для чтения файл, с которым связана файловая переменная *f*. После успешного выполнения процедуры файл готов к чтению из него первого элемента. Процедура завершается с сообщением об ошибке, если указанный файл не найден.

*Rewrite(f)*

– открывает для записи файл, с которым связана файловая переменная *f*. После успешного выполнения этой процедуры файл готов к записи в него первого элемента. Если указанный файл уже существовал, то все данные из него уничтожаются.

*Close(f)*

– закрывает открытый до этого файл с файловой переменной *f*. Если по какой-то причине процедура не будет выполнена, файл будет создан на внешнем устройстве, но содержимое последнего буфера в него не будет перенесено.

*EOF(f): boolean*

– возвращает значение *TRUE*, когда при чтении достигнут конец файла. Это означает, что уже прочитан последний элемент в файле или файл после открытия оказался пуст.

*Rename(f, NewName)*

– позволяет переименовать физический файл на диске, связанный с файловой переменной *f*. Переименование возможно после закрытия файла.

*Erase(f)*

– уничтожает физический файл на диске, который был связан с файловой переменной *f*. Файл к моменту вызова процедуры должен быть закрыт.

РЕКЛАМА

**Работа с текстовыми файлами**

Текстовый файл – это совокупность строк, разделенных метками конца строки. Сам файл заканчивается меткой конца файла. Доступ к каждой строке возможен лишь последовательно, начиная с первой. Одновременная запись и чтение – запрещены.

*Read(f, список переменных); ReadLn(f, список переменных);*

– процедуры читают информацию из файла *f* в переменные. Отличие *ReadLn* от *Read* в том, что в нем после прочтения данных пропускаются все оставшиеся символы в данной строке, включая метку конца строки. Если список переменных отсутствует, то процедура *ReadLn(f)* пропускает строку при чтении текстового файла.

*Write(f, список переменных); WriteLn(f, список переменных);*

– процедуры записывают информацию в текстовый файл. *WriteLn* от *Write* отличается тем, что после записи всех значений из переменных записывает еще и метку конца строки.

*Append(f)*

*–*открывает текстовый файл для добавления информации к его концу. Используется вместо *Rewrite*.

**Работа с типизированными файлами**

Типизированный файл – это последовательность компонент любого заданного типа, кроме типа "файл". Доступ к компонентам файла осуществляется по их порядковым номерам. Компоненты нумеруются, начиная с 0. После открытия файла указатель стоит в его начале на нулевом компоненте. После каждого чтения или записи указатель сдвигается к следующему компоненту.

*Reset(f)*

– открывает файл, с которым связана файловая переменная *f,* для чтения и записи одновременно.

*Write(f, список переменных)*

– записывает в файл *f* всю информацию из списка переменных.

*Read(f, список переменных)*

– читает из файла *f* компоненты в указанные переменные. Тип файловых компонент и переменных должны совпадать. Если будет сделана попытка чтения несуществующих компонент, то произойдет ошибочное завершение программы.

*Seek(f, n)*

*–*смещает указатель файла *f* на *n*-ную позицию. Нумерация в файле начинается с 0.

*FileSize(f): longint*

– возвращает количество компонент в файле *f*.

*FilePos(f): longint*

*–* возвращает порядковый номер текущего компонента файла *f*.

*Truncate(f)*

– отсекает конец файла, начиная с текущей позиции включительно.

**Задания:**

**1.**Дано имя файла и целое число N (> 1). Создать файл целых чисел с данным именем и записать в него N первых положительных четных чисел (2, 4, …).

**2.**Дано имя файла и вещественные числа A и D. Создать файл вещественных чисел с данным именем и записать в него 10 первых членов арифметической прогрессии с начальным членом A и разностью D: A, A + D, A + 2·D, A + 3·D, … .

**3.**Дано имя файла целых чисел. Найти количество элементов, содержащихся в данном файле. Если файла с таким именем не существует, то вывести −1.

**4.**Дано целое число K и файл, содержащий неотрицательные целые числа. Вывести K-й элемент файла. Если такой элемент отсутствует, то вывести −1.

**5.**Дан файл целых чисел. Создать новый файл, содержащий те же элементы, что и исходный файл, но в обратном порядке.

**6.**Дан файл вещественных чисел. Создать два новых файла, первый из которых содержит элементы исходного файла с нечетными номерами (1, 3, …), а второй — с четными (2, 4, …).

**7.**Дан файл целых чисел. Создать два новых файла, первый из которых содержит положительные числа из исходного файла (в обратном порядке), а второй — отрицательные (также в обратном порядке). Если положительные или отрицательные числа в исходном файле отсутствуют, то соответствующий результирующий файл оставить пустым.

**8.**Дан файл вещественных чисел. Найти среднее арифметическое его элементов

**9.**Дан файл вещественных чисел. Найти сумму его элементов с четными номерами.

**10.**Дан файл вещественных чисел. Найти его первый локальный минимум (локальным минимумом называется элемент, который меньше своих соседей).

**11.**Дан файл вещественных чисел. Поменять в нем местами минимальный и максимальный элементы.

**12.**Дан файл целых чисел с элементами A1, A2, …, AN (N — количество элементов в файле). Заменить исходное расположение его элементов на следующее: A1, AN, A2, AN−1, A3, … .

**13.**Дан файл вещественных чисел. Заменить в файле каждый элемент, кроме начального и конечного, на его среднее арифметическое с предыдущим и последующим элементом.

**14.**Дан файл целых чисел, содержащий более 5 элементов. Уменьшить его размер до 5 элементов, удалив из файла необходимое количество конечных элементов.

**15.**Дан файл целых чисел. Удалить из него все элементы с четными номерами.

**16.**Дан файл целых чисел. Удвоить его размер, записав в конец файла все его исходные элементы (в том же порядке).

**17.**Дан файл целых чисел. Продублировать в нем все элементы с нечетными номерами.

**18.**Дан файл целых чисел. Продублировать в нем все числа, принадлежащие диапазону 5 – 10.

**19.**Дан файл целых чисел. Заменить в нем каждый элемент с четным номером на два нуля.

**20.**Дан файл целых чисел. Заменить в нем каждое положительное число на три нуля.