**РАБОТА С ТЕКСТОВЫМИ ФАЙЛАМИ**

**Организация текстовых файлов**

Текстовые файлы состоят из одной или нескольких строк с цифрами, словами или символами. Типовые текстовые файлы в компьютере идентифицируются MS Windows Explorer (Проводником) как текстовые документы и могут иметь расширения .txt, .ini, .log, .inf, .dat или .bat.

В VBA предусматривается использование файлов, организованных двумя способами. Первый – организация файлов последовательного доступа, второй – организация файлов прямого доступа. В обоих случаях файл представляет собой совокупность связанных записей, рассматриваемых как единое целое и имеющих общее имя. Записи, как правило, имеют одинаковую структуру и состоят из одного или нескольких полей. В каждом поле хранится какой-либо показатель (параметр, характеристика и т. п.).

В *файлах последовательного доступа* записи располагаются в том порядке, в каком они поступили в файл – последовательно (одна за другой). Для поиска требуемой записи необходимо последовательно просматривать весь файл – от начала до искомой записи. Следовательно, время доступа к записи находится в прямой зависимости от ее местоположения в файле. Особенностью файлов с последовательным доступом является то, что они могут иметь записи различной длины, которая определяется длинами входящих в запись полей. Такие файлы целесообразно использовать в том случае, если при обращении к файлу обрабатываются почти все записи, а их содержимое меняется достаточно редко. Основной недостаток файлов с последовательным доступом заключается в трудности обновления существующих записей, замены записи на новую, вставки записей. Файлы с последовательным доступом могут быть организованы на магнитных лентах и магнитных дисках.

*Файлы с прямым* или, как часто их называют, *с произвольным доступом* организованы таким образом, что можно осуществить непосредственный прямой доступ к любой записи файла независимо от ее местоположения. Для поиска требуемой информации необходимо знать только номер записи, в которой расположены соответствующие данные. С каждой записью в файле связан уникальный номер, который однозначно определяет ее положение в файле. Нумерация записей ведется последовательно, начиная с единицы, а номер записи фактически является ключом поиска.

Способ задания номера записи зависит от данных, составляющих запись, и решаемой задачи. Желательно, чтобы номер записи был связан с данными, входящими в нее, и располагался в одном из полей записи. Например, если записи содержат информацию о сотрудниках учреждения, то в качестве ключа поиска, т.е. номера записи, целесообразно использовать табельный номер сотрудника. Однако это необязательно.

Файлы прямого доступа могут быть организованы только на магнитном диске. В таких файлах записи должны иметь строго определенный формат, а, следовательно, и длину. Файлы с прямым доступом широко используются в информационно-поисковых системах, где требуется быстрый поиск информации и возникает необходимость в частом изменении содержимого различных записей (полей) или их замене.

**Операции с файлами**

При создании приложений с текстовыми файлами может возникнуть необходимость удаления, переименования, копирования файла, определения его длины, числа записей и др. VBA предоставляет ряд операторов для выполнения различных действий над файлами.

Оператор **Kill** – удаляет файл

**Kill** File\_name

где *File\_name -* полное имя удаляемого файла. Можно использовать шаблоны «\*» и «?». Например, \*.tmp, чтобы удалить все файлы с расширением tmp.

Оператор **Name** – переименовывает файл

**Name** Old\_name **As** New\_name

где *Old\_name* – строка, содержащая исходный путь и имя переименуемого файла; *New\_name* – новое имя файла. За одну операцию может быть переименован только один файл. Если новое имя содержит другой путь, файл будет перемещен в этот каталог.

Оператор **FileCopy** – копирует файл

**FileCopy** Source Destination

где: *Source* – строка, содержащая имя копируемого файла; *Destination* – имя нового файла. Оба имени могут при необходимости включать каталог и дисковод. За одну операцию может быть скопирован только один файл.

Операторы **MkDir**, **RmDir** и **ChDir** – создают, удаляют и изменяют каталоги соответственно. За каждым из этих операторов должна следовать строка, содержащая имя каталога.

Функция **EOF** – используется для обнаружения конца файла. Функция возвращает значение *True*, когда достигнут маркер конца файла.

Функция **LOF** – определяет длину файла в байтах. Ее можно использовать для определения количества записей в файле, разделив полученную длину файла на длину записи.

Функция **LOC** – возвращает число символов, прочитанных на данный момент. Ее можно использовать для определения количества прочитанных записей.

Все три функции имеют в качестве аргумента номер файла. Номер может принимать значения от 1 до 255 и связывается с файлом при его открытии. Когда файл закрыт, номер может быть использован повторно для другого файла. Если файл открывается заново, номер также присваивается заново.

Свободный номер файла, особенно при разработке больших программ, удобно получить, используя функцию **FreeFile**.

**Файлы последовательного доступа**

Последовательные файлы состоят из рядов строк текста, и их часто называют ASCII-файлами. Каждая строка текста завершается символом возврата каретки (ASCII 13), а файл заканчивается сим­волом Ctrl-Z (ASCII 26). Каждый символ текста занимает один байт. Текст можно просматривать и редактировать с помощью редактора *Notepad* (Блокнот).

С последовательным файлом можно работать в двух режимах: запись (или добавление данных) и чтение. Для создания последовательного файла необходимо открыть файл, записать данные и закрыть файл.

Файл открывается оператором вида:

**Open** "filename" **For** mode **As** #number

filename – полное имя файла,

mode – режим доступа к файлу:

*Output* – это режим, при котором файл открыт для записи; обычно при этом режиме создается новый файл; если открывается для записи уже существующий файл, то ранее хранимая информация будет стерта.

*Input* – это режим, при котором файл открыт только для чтения информации, записывать в файл при таком режиме нельзя;

*Append* – это режим, при котором файл открыт для записи, но в отличие от режима OUTPUT, при котором добавление новой информации приводит к уничтожению имеющейся, в режиме APPEND новая информация размещается в конце файла, за последней записью без уничтожения ранее имеющейся

number – идентифицирует файл в программе и может принимать значения от 1 до 255. Номер связан с файлом только тогда, когда файл открыт.

В последовательный файл *данные записываются* с помощью оператора **Print** или **Write**. Каждый оператор записывает одну строку данных в файл, заканчивающуюся возвратом каретки. Формат операторов следующий:

**Print** #number, valuel, value2,...

**Write** #number, valuel, value2,...

Оператор **Print** записывает данные с метками между отдельными элементами. Он подходит для записи текстовых файлов с одним элементом на строку. Оператор **Write** хранит данные в формате, разделенном запятыми.

Файл закрывается оператором **Close** в формате

**Close** #number

*Чтение последовательного файла*, как и его создание, состоит из трех этапов: открытие файла, чтение данных и закрытие файла. Файл открывается с помощью оператора следующего вида:

**Open** "filename" **For** Input **As** #number

Данные из файла читаются с помощью одного из приведенных операторов:

**Line Input** #number, string

**Input** #number, variablel, variable2,...

**Line Input** считывает строку полностью в переменную string и может использоваться для чтения текстовых файлов, когда они записаны с помощью оператора **Print**. Оператор считывает из файла по одному символу, пока не встретит символ «возврат каретки» или последовательность «возврат каретки» – перевод строки».

**Input** считывает данные и размещает их в переменных заданного списка. Данные считываемые оператором записываются в файл с помощью оператора **Write** #.

Последовательные файлы всегда должны читаться с начала файла. Для обнаружения конца файла можно использовать функцию **EOF**. После завершения чтения файл нужно закрыть с помощью команды **Close**.

**Пример 1.**

**Sub** filee()

‘открытие файла и запись в нее информации

Number = FreeFile

**Open** "d:\3.txt" **For** Output **As** #Number

**Print** #Number, "Это хранится в файле"

**Close** #Number

‘открытие файла и чтение из него информации

Number = FreeFile

**Open** "d:\3.txt" **For** Input **As** #Number

**Input** #Number, a$

**MsgBox** ("Чтение из файла: " & Chr(13) & a$)

**Close** #Number

**End Sub**

**Пример 2.**

**Sub** filee()

‘открытие файла и запись в нее информации

Number = FreeFile

n$ = **InputBox**("ВВедите имя файла")

**Open** n$ **For** Output **As** #Number

**Print** #Number, "Это хранится в файле"

**Close** #Number

‘открытие файла и чтение из него информации

Number = FreeFile

**Open** n$ **For** Input **As** #Number

**Input** #Number, a$

**MsgBox** ("Чтение из файла: " & Chr(13) & a$)

**Close** #Number

**End** Sub

**Пример 3.**

Sub filee()

Number = FreeFile

n$ = **InputBox**("ВВедите имя файла")

**Open** n$ **For** Output **As** #Number

**Do**

marka$ = **InputBox**("Введите марку автомобиля (для выхода QWE)")

**If** marka$ = "QWE" **Then Exit Do**

Inomer$ = **InputBox**("Введите номер автомобиля")

tsvet$ = **InputBox**("Введите цвет автомобиля")

god$ = **InputBox**("Введите год производства автомобиля")

tsena$ = **InputBox**("Введите продажную цену автомобиля")

'запись информации в файл

**Write** #Number, marka$, nomer$, tsvet$, god$, tsena$

**Loop**

'закрытие файла

**Close** #Number

**MsgBox** ("Внесение информации завершено")

Number = FreeFile

**Open** n$ **For** Input **As** #Number

i% = 1

**Do Until EOF**(Number)

**Input** #Number, marka$, nomer$, tsvet$, god$, tsena$

**MsgBox** "Данные об " & i% & " автомобиле" & Chr(13) & marka$ & Chr(9) & nomer$ & Chr(9) & tsvet$ & Chr(9) & god$ & Chr(9) & tsena$

i = i + 1

**Loop**

**Close** #Number

**End** Sub

**Пример 4. Изменение содержания файла**

**Sub** filee()

**Open** "c:\avto.txt" For Input As #1

**Open** "c:\avto2.txt" For Output As #2

**Do**

**Input** #1, marka$, nomer$, tsvet$, god$, tsena$

**If** marka$ = "москвич" **Then** marka$ = "ока"

**Print** #2, marka$, nomer$, tsvet$, god$, tsena$

**Loop Until EOF**(1)

**Close** #1, #2

'удаление исходного файла

**Kill** "c:\avto.txt"

' переименование нового файла

**Name** "c:\avto2.txt" **As** "c:\avto.txt"

**End** Sub

**Файлы с прямым доступом**

Такие файлы хранят данные в формате, основанном на записи. Этот формат задается пользователем с помощью оператора **Type...End Туре** в зависимости от количества полей и их типа в записи:

Type recordtype

Variablel As type

Variable2 As type

…

VariableN As type

End Type

Записи читаются и/или записываются в три этапа: открытие файла, чтение или запись данных, закрытие файла.

Для открытия файла используется оператор **Open** в следующем формате:

**Open** "filename" **For** Random **As** #number **Len** = length

Если опустить параметр Len, то длина записи по умолчанию будет равна 128 байтам.

Для того чтобы сделать запись, нужно сначала объявить переменную, которая имеет тип, предварительно определенный пользователем с помощью оператора **Type...End Type**. Например^

**Dim** Record **As** recordtype

После этого нужно заполнить запись значениями отдельных переменных, представляющих поля записи:

Record.variablel = выражение1

Record.variable2 = выражение2

Сформированная запись *записывается в файл* с помощью оператора Put следующего формата:

**Put** #number, recordnumber, Record

где *recordnumber* – номер записи, начинающийся с 1 для первой записи. Если номер находится за пределами конца текущего файла, файл расширяется. При этом неиспользованные записи будут содержать «мусор», если их специально не очистить.

*Читаются записи* из файла с помощью оператора Get следующего формата:

**Get** #number, recordnumber, Record

Переменные в *Record* заполняются соответствующими данными из записи и могут использоваться для отображения, печати и др.

Рассмотрим пример работы с файлом прямого доступа. Пусть требуется разработать программу заполнения файла прямого доступа записями заданной структуры с последующим чтением записи заданного номера. Формат записи:

Поле1 – Фамилия

Поле2 – Имя

Поле3 – Номер группы

После разработки формы к проекту необходимо добавить модуль, в котором следует с помощью оператора **Туре.. .End Type** ввести структуру записи:

**Type** recordtype

Name As String\* 15

Namel As String\* 15

NumberGroup As String \* 6

**End Type**

**Privat Sub** Command1\_Click()

Dim Record **As** recordtype

**Dim** number **As** Integer, RetInt **As** Integer, number1 **As** Integer

Dim NumberFile **As** Integer

NumberFile = FreeFile

**Open** "C:\File.txt" For Random As #NumberFile Len = 36

number =1

**Do**

Record.Name = **InputBox**("Введите фамилию", "Окно ввода")

Record.Namel = **InputBox**("Введите имя", "Окно ввода")

Record. NumberGroup = **InputBox**("Введите группу", "Окно ввода")

**Put** #NumberFile, number, Record

RetInt = **MsgBox**("Продолжить ввод? Да или Нет", vbQuestion +vbYesNoCancel)

**If** RetInt = vbYes **Then**

number = number + 1

**Else**

**Exit Do**

**End If**

**Loop**

numberl = InputBox("Введите № записи,которую нужно выдать")

**If** numberl *<=* number **And** numberl <> 0 **Then**

**Get** #NumberFile, numberl, Record

Text1.Text = Record.Name

Text2.Text = Record.Name1

Text3.Text = Record.NumberGroup

**Else**

**MsgBox** "Запись не найдена", vbCritical, "Ошибка"

**End If**

**End Sub**