электронный научно-технический журнал
ИНЖЕНЕРНЫЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51036. ISSN 2307-0595

Методика разработки программ обработки файлов произвольного доступа

 # 12, декабрь 2014

Мартынюк Н. Н., Степанова И. И.

УДК: 004.043.085

Россия, МГТУ им. Н.Э. Баумана
martynyuk1950@mail.ru

Вопросы разработки алгоритмов и написание программ для обработки файлов вызывают большие затруднения у студентов. Рассматриваются вопросы, связанные с обработкой файлов произвольного доступа в среде Visual Studio на языке Visual Basic.

Различают физические и логические файлы.

Физический файл – это именованная область носителя, предназначенная для хранения информации. Имя физического файла состоит из пути к файлу, имени и расширения, например: “D:\temp\imv\file1.dat”.

Логический файл – положительное целое число (дескриптор файла), которое при выполнении процедуры открытия файла связывается с физическим файлом.

Относительно программ, разработанных на языке BASIC, файлы можно подразделить на следующие группы: типизированные (файлы произвольного доступа), текстовые (файлы последовательного доступа) и нетипизированные файлы. Файлы произвольного доступа можно создать только с помощью программы, написанной на языке BASIC. Текстовые файлы создает программа Блокнот. Все остальные физические файлы относятся к группе нетипизированных.

Типизированные файлы (Файлы произвольного доступа)

Типизированный файл представляет собой последовательность компонентов одного типа. Тип компонентов может быть любой, кроме файлового типа.

Свойства типизированных файлов

Данные в типизированном файле хранятся во внутреннем представлении, при занесении данных в файл преобразование в символьное представление не происходит. Поэтому создать файл или просмотреть его в текстовом редакторе нельзя. Для этого необходимо составить программу.

Типизированный файл, как и все остальные, является файлом последовательного доступа. Но так как основное свойство типизированного файла заключается в том, что все компоненты в нем одного типа, и для их хранения выделяется одинаковое количество байт, можно организовать прямой доступ. Это значит, что к любой компоненте файла можно обратиться непосредственно, минуя предыдущие компоненты. Поэтому такие файлы называются файлами произвольного доступа.

После открытия типизированного файла можно и читать из файла, и писать в файл.

Для обмена информацией с файлом необходимо использовать переменную такого же типа, как и компоненты файла.

Структура типизированного файла имеет вид, представленный на рис.1.

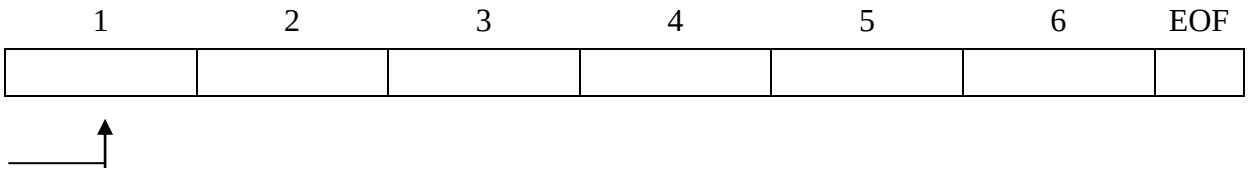

Рис 1.

Все компоненты в файле пронумерованы, имеют одинаковую длину. В конце файла ставится признак конца (**End Of File**). Каждый файл имеет указатель, который содержит адрес компоненты, с которой возможен обмен в данный момент. Указатель сдвигается на следующую компоненту при каждом чтении из файла или записи в файл.

Процедуры для работы с типизированными файлами

FileOpen(<дескриптор файла>, <имя физического файла>, <режим работы>, , ,<длина компонента файла>). Связывает имя физического файла с дескриптором. После выполнения этой процедуры у файла в программе появится «псевдоним» - дескриптор файла. Далее все подпрограммы обращаться к файлу должны, задавая дескриптор. В качестве дескриптора необходимо использовать целое положительное число.

<Режим работы> - для работы с файлами произвольного доступа необходимо использовать режим OpenMode.Random.

Пример 1. Файл целых чисел.

```
Dim a as integer, fn as string
Fileopen(1,fn,OpenMode.Random,,,len(a))
```

Пример 2. Файл структур.

```
Dim a as student, fn as string
(student  -  тип-структура. Включает поля Фамилия, группа, массив оценок и т.д.
Fileopen(2,fn,OpenMode.Random,,,80)
```

80 – длина каждого компонента файла. Предварительно необходимо сделать оценку длины, исходя из состава полей структуры. И задать такую длину компонента файла, чтобы хватило места для самой длинной записи о студенте.

Для ввода данных из файла в программу используется процедура Fileget в 2 вариантах.

Первый вариант:

Fileget(<дескриптор файла>, <имя переменной соответствующего типа>)

Процедура работает следующим образом:

- Читает из файла содержимое очередного компонента и записывает в переменную.
- Указатель сдвигается на следующий компонент файла. Реализуется последовательный доступ к элементам файла.
- Если указатель достигает признак конца файла (EOF) и делается попытка чтения из файла, то происходит прерывание работы программы.

Пример.

```
Dim a as integer
Fileopen(1,fn,OpenMode.Random,,,len(a))
 Fileget(1,a)
```

Второй вариант:

Fileget(<дескриптор файла>, <имя переменной соответствующего типа>, <номер компонента файла>)

Процедура работает следующим образом:

- Указатель файла перемещает на компонент, номер которого задан.
- Читает из файла содержимое соответствующего компонента и записывает в переменную. Реализуется прямой доступ к элементам файла. (Если делается попытка чтения после конца файла, то происходит прерывание работы программы.)
- Указатель сдвигается на следующую компоненту файла.

Пример.

```
Dim a as integer
Fileopen(1,fn,OpenMode.Random,,,len(a))
Fileget(1,a,5)
```

Для вывода данных из программы в файл используется процедура Fileput в 2 вариантах.

Первый вариант:

Fileput(<дескриптор файла>, <имя переменной соответствующего типа>)

Процедура работает следующим образом:

- Записывает в файл значение переменной в компонент, на котором стоял указатель файла.

- После записи переменной указатель сдвигается на следующий компонент файла. Реализуется последовательный доступ к элементам файла.

Пример.

```
Dim a as integer
a=15
Fileopen(1,fn,OpenMode.Random,,,len(a))
Fileput(1,a)
```

Второй вариант:

Fileput(<дескриптор файла>, <имя переменной соответствующего типа>, <номер компонента файла>)

Процедура работает следующим образом:

- Указатель файла перемещает на компонент, номер которого задан.
- Записывает в файл значение переменной в соответствующий компонент файла. Реализуется прямой доступ к элементам файла.
- После записи переменной указатель сдвигается на следующий компонент файла.

Пример.

```
Dim a as integer
a=15
Fileopen(1,fn,OpenMode.Random,,,len(a))
Fileput(1,a,6)
```

После окончания работы файл необходимо закрыть.

Fileclose((<дескриптор файла>) - Прерывает связь между дескриптором и физическим файлом. Файл окончательно обновляется и закрывается.

Для удаления файла из папки используется процедура Kill(<имя физического файла>). Перед удалением необходимо файл закрыть.

Пример:

```
FileClose(1)
Kill(fn)
```

Если необходимо переименовать файл, надо использовать процедуру Rename. Перед переименовыванием необходимо файл закрыть.

Rename(<старое физическое имя файла>, <новое физическое имя файла>)

Пример:

```
FileClose(1)
Rename(”C:\Temp\sp.dat”,”A:\b.dat”)
```

Функции для работы с типизированными файлами

Eof(<дескриптор файла >) as Boolean. Проверяет, стоит ли указатель файла на признаке конца файла? Если да, возвращает **True**, иначе **False.**

Пример:
If Eof(1) then ...

Lof(<дескриптор файла >) as integer. Возвращает длину файла в байтах.

Пример:
Dim dl as integer
dl=Lof(F)

Во многих алгоритмах необходимо знать количество компонентов в файле. Для нахождения чила компонентов в файле, надо длину всего файла поделить на длину одного компонента:

Пример.
Dim n,a as integer
Dim fn as string
Fileopen(1,fn,OpenMode.Random,,,len(a))
N=Lof(1)\len(a)

Ниже приведены 2 примера программ обработки файлов произвольного доступа. Первый пример показывает работу с файлом целых чисел. Второй работает с файлом, в котором хранятся сведения о студентах, данные о каждом студенте представлены в виде структуры.

Пример 1.

Разработать приложение для работы с типизированным файлом целых чисел. Предусмотреть создание файла двумя способами: с помощью ввода чисел с клавиатуры и чтения чисел из заранее созданного текстового файла. Вывести содержимое файла в окно списка ListBox. Для задания имени файла использовать компоненты SaveFileDialog и OpenFileDialog. Создать меню для работы с файлом, используя компоненту MenuStrip.

Разработать модуль, в который поместить следующие подпрограммы работы с файлом:

1. Функция, которая проверяет, составляют ли числа файла арифметическую прогрессию. Если составляют, функция возвращает TRUE, иначе функция возвращает FALSE.
2. Процедура редактирования файл: каждое четное число возводится в квадрат.
3. Процедура добавления несколько чисел в конец файла.
4. Процедура удаления из файла всех чисел, равных нулю.

```
Public Class Form1
    Public fname As String
    Private Sub ВыходToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ВыходToolStripMenuItem.Click
```

```
        End
    End Sub

    Private Sub СоздатьToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
СоздатьToolStripMenuItem.Click
        SaveFileDialog1.ShowDialog()
        fname = SaveFileDialog1.FileName
        sozd(fname)
    End Sub

    Private Sub ОткрытьToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ОткрытьToolStripMenuItem.Click
        OpenFileDialog1.ShowDialog()
        fname = OpenFileDialog1.FileName
        prosmotr(fname, ListBox1)
    End Sub

    Private Sub РедактироватьToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles РедактироватьToolStripMenuItem.Click
        If fname = "" Then
            OpenFileDialog1.ShowDialog()
            fname = OpenFileDialog1.FileName
        End If
        redakt(fname)
        prosmotr(fname, ListBox2)
    End Sub

    Private Sub УдалениеToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
УдалениеToolStripMenuItem.Click
        If fname = "" Then
            OpenFileDialog1.ShowDialog()
            fname = OpenFileDialog1.FileName
        End If
        udal(fname)
        prosmotr(fname, ListBox2)
    End Sub

    Private Sub ПроверкаToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ПроверкаToolStripMenuItem.Click
        If fname = "" Then
            OpenFileDialog1.ShowDialog()
            fname = OpenFileDialog1.FileName
        End If
        If proverka(fname) = True Then
            MsgBox("Числа в файле составляют арифметическую прогрессию")
        Else
            MsgBox("Числа в файле не составляют арифметическую прогрессию")
        End If
    End Sub
```

```
    Private Sub СоздатьТекстToolStripMenuItem1_Click(sender As Object, e As EventArgs)
Handles ToolStripMenuItem1.Click
        Dim fname1 As String
        OpenFileDialog1.ShowDialog()
        fname1 = OpenFileDialog1.FileName
        SaveFileDialog1.ShowDialog()
        fname = SaveFileDialog1.FileName
        sozd_iz_text(fname, fname1)
    End Sub
End Class
```

Модуль с подпрограммами обработки файла

```
Module Module1
    Public Sub sozd(ByVal fn As String)
        'Процедура создания файла целых чисел.Числа вводятся с клавиатуры.
        'fn - входной параметр, имя физического файла, который будет создан.
        Dim i, n, a As Integer
        FileOpen(1, fn, OpenMode.Random, , , Len(a))
        n = Val(InputBox("введите количество чисел"))
        For i = 1 To n
            a = Val(InputBox("Введите число"))
            FilePut(1, a)
        Next
        FileClose(1)
    End Sub

    Public Sub sozd_iz_text(ByVal fn As String, ByVal fn1 As String)
        'Процедура создания файла целых чисел. Числа вводятся из текстового файла.
        'fn - входной параметр, имя физического файла, который будет создан.
        'fn1 - входной параметр, имя физического текстового файла, из которого читаются
числа.
        Dim i, n, a As Integer
        FileOpen(1, fn, OpenMode.Random, , , Len(a))
        FileOpen(2, fn1, OpenMode.Input)
        Do Until EOF(2)
            a = Val(LineInput(2))
            FilePut(1, a)
        Loop
        FileClose(1, 2)
    End Sub

    Public Sub prosmotr(ByVal fn As String, ByRef list As ListBox)
        'Процедура вывода содержимого файла целых чисел в окно списка
        'fn - входной параметр, имя физического файла, содержимое которого будет выведе-
но.
        'list - имя окна списка, в который будет осуществлен вывод.
        Dim i, n, a As Integer
        FileOpen(1, fn, OpenMode.Random, , , Len(a))
        n = LOF(1) \ Len(a)
        For i = 1 To n
```

```
            FileGet(1, a)
            list.Items.Add(Str(a))
        Next
        FileClose(1)
    End Sub

    Public Sub redakt(ByVal fn As String)
        'Процедура редактирует файл: каждое четное число возводит в квадрат.
        'fn - входной параметр, имя физического файла, который будет изменен.
        Dim i, n, a As Integer
        FileOpen(1, fn, OpenMode.Random, , , Len(a))
        n = LOF(1) \ Len(a)
        For i = 1 To n
            FileGet(1, a, i)
            If a Mod 2 = 0 Then
                a = a * a
                FilePut(1, a, i)
            End If
        Next
        FileClose(1)
    End Sub

    Public Sub dobav(ByVal fn As String)
        'Процедура добавляет несколько чисел в конец файла.
        'fn - входной параметр, имя физического файла, который будет изменен.
        Dim i, n, a, k As Integer
        FileOpen(1, fn, OpenMode.Random, , , Len(a))
        n = LOF(1) \ Len(a)
        k = Val(InputBox("введите количество добавляемых чисел"))
        For i = 1 To k
            a = Val(InputBox("Введите число"))
            FilePut(1, a, n + i)
        Next
        FileClose(1)
    End Sub

    Public Sub udal(ByVal fn As String)
        'Процедура удаляет из файла все числа равные нулю.
        'fn - входной параметр, имя физического файла, который будет изменен.
        Dim i, n, a As Integer
        Dim fn1 As String
        FileOpen(1, fn, OpenMode.Random, , , Len(a))
        n = LOF(1) \ Len(a)
        fn1 = fn + "1"
        FileOpen(2, fn1, OpenMode.Random, , , Len(a))
        For i = 1 To n
            FileGet(1, a)
            If a <> 0 Then FilePut(2, a)
        Next
        FileClose(1, 2)
```

```
        Kill(fn)
        Rename(fn1, fn)
    End Sub

    Public Function proverka(ByVal fn As String) As Boolean
        'Функция проверяет, составляют ли числа файла арифметическую прогрессию.
        ' Если составляют, функция возвращает TRUE. В противном случае функция возвращает
FALSE.
        'fn - входной параметр, имя физического файла.
        Dim i, n, a, b, d As Integer
        FileOpen(1, fn, OpenMode.Random, , , Len(a))
        n = LOF(1) \ Len(a)
        FileGet(1, a)
        FileGet(1, b)
        d = b - a
        proverka = True
        For i = 3 To n
            a = b
            FileGet(1, b)
            If b - a <> d Then proverka = False
        Next
        FileClose(1)
    End Function

End Module
```

Пример 2.

Разработать приложение для работы с типизированным файлом структур, содержащих информацию о студентах. О каждом студенте известно: фамилия, группа, 3 оценки. Каждая структура имеет: процедуру ввода; функцию вычисления среднего балла; функцию формирования строки для вывода. Вывести содержимое файла в окно списка ListBox. Для задания имени файла использовать компоненты SaveFileDialog и OpenFileDialog. Создать меню для работы с файлом, используя компоненту MenuStrip.

Разработать модуль, в который поместить следующие подпрограммы работы с файлом:

1. Процедура сортировки студентов по фамилиям по алфавиту.
2. Процедура создания нового файла из студентов заданной группы, имеющих средний балл выше среднего по всем студентам. Группа передается в процедуру как параметр.
3. Процедура редактирования файла.

```
Public Class Form1
    Dim fn, fn2 As String
    Private Sub ВыходToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ВыходToolStripMenuItem.Click
        End
    End Sub
```

```
    Private Sub ЗаданиеСКлавиатурыToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ЗаданиеСКлавиатурыToolStripMenuItem.Click
        'Исходные данные вводятся в файл  пользователем с клавиатуры
        SaveFileDialog1.ShowDialog()
        fn = SaveFileDialog1.FileName
        Создание1(fn)
    End Sub

    Private Sub ПечатьФайлаToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ПечатьФайлаToolStripMenuItem.Click
        If fn = "" Then
            OpenFileDialog1.ShowDialog()
            fn = OpenFileDialog1.FileName
        End If
        Печать(fn, ListBox1)
    End Sub

    Private Sub СортировкаToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles СортировкаToolStripMenuItem.Click
        'Сортировка файла по фамилиям
        If fn = "" Then
            OpenFileDialog1.ShowDialog()
            fn = OpenFileDialog1.FileName
        End If
        Отсортировать(fn)
        Печать(fn, ListBox2)
    End Sub

    Private Sub РедактированиеToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles РедактированиеToolStripMenuItem.Click
        'Редактирование данных, записанных в файл
        If fn = "" Then
            OpenFileDialog1.ShowDialog()
            fn = OpenFileDialog1.FileName
        End If
        Корректировка(fn)
    End Sub

    Private Sub НовыйФайлToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles НовыйФайлToolStripMenuItem.Click
        'Запись в новый файл студентов заданной группы, средний балл которых выше общего
среднего балла
        Dim gruppa, fn1 As String
        If fn = "" Then
            OpenFileDialog1.ShowDialog()
            fn = OpenFileDialog1.FileName
        End If
        MsgBox("Задайте имя файла студентов заданной группы")
        SaveFileDialog1.ShowDialog()
        fn1 = SaveFileDialog1.FileName
        gruppa = InputBox("Введите номер группы")
        СрБалл(fn, fn1, gruppa)
```

```
        Печать(fn1, ListBox2)
    End Sub
End Class
```

Модуль с подпрограммами обработки файла

```
Module Module1
    Public Structure Студент
        Public Фамилия As String
        Public Группа As String
        Public Оценки() As Byte
Public Sub Ввод()
        Dim  j As Integer
Фамилия= InputBox("Фамилия  студента")
Группа= InputBox("Группа  студента")
ReDim st.Оценки(2)
For j = 0 To 2
  st.Оценки(j) = Val(InputBox("Оценка " + Str(j + 1) + "-я " +студента"))
         Next
End Sub

        Public Function СреднийБалл() As Single
            Dim sum, i As Byte
            sum = 0
            For i = 0 To 2
                sum = sum + Оценки(i)
            Next
            СреднийБалл = sum / 3
        End Function

        Public Function Строка() As String
            Dim i As Integer
            Строка = LSet(Фамилия, 20) + LSet(Группа, 10)
            For i = 0 To 2
                Строка = Строка + Str(Оценки(i)) + vbTab
            Next
        End Function
    End Structure

    Public st As Студент

    Public Sub Создание1(ByVal fn As String)
        Dim n, i, j As Integer
        FileOpen(1, fn, OpenMode.Random, , , 50)
        n = Val(InputBox("Количество студентов"))
        For i = 1 To n
            St.vvod()
            FilePut(1, st)
        Next
        FileClose()
    End Sub

    Public Function Ideal(ByRef s As String) As String
        'Функция удаляет точки в строке, заменяя их пробелами, а также удаляет между словами
```

```
        ' лишние пробелы, оставляя по одному из них
        s = Replace(s, ".", " ")
        s = Trim(s)
        Do While InStr(s, "  ") <> 0
            s = Replace(s, "  ", " ")
        Loop
        Return s
    End Function

    Public Sub Печать(ByVal fn As String, ByRef lst As ListBox)
        lst.Items.Clear()
        FileOpen(1, fn, OpenMode.Random, , , 50)
        Do Until EOF(1)
            FileGet(1, st)
            lst.Items.Add(st.Строка)
        Loop
        FileClose()
    End Sub

    Public Sub Отсортировать(ByVal fn As String)
        Dim i, j, n As Integer
        Dim st1, st2 As Студент
        FileOpen(1, fn, OpenMode.Random, , , 50)
        n = LOF(1) \ 50 + 1
        MsgBox(n)
        For i = 1 To n - 1
            For j = 1 To n - i
                FileGet(1, st1, j)
                FileGet(1, st2, j + 1)
                If st1.Фамилия > st2.Фамилия Then
                    FilePut(1, st2, j)
                    FilePut(1, st1, j + 1)
                End If
            Next
        Next
        FileClose()
    End Sub

    Public Sub Корректировка(ByVal fn As String)
        Dim n, i, j As Integer
        FileOpen(1, fn, OpenMode.Random, , , 50)
        n = LOF(1) \ 50 + 1
        For i = 1 To n
            FileGet(1, st, i)
            st.Фамилия = InputBox("Фамилия " + Str(i) + "-го студента", "", st.Фамилия)
            st.Группа = InputBox("Группа " + Str(i) + "-го студента", "", st.Группа)
            ' ReDim st.Оценки(2)
            For j = 0 To 2
                st.Оценки(j) = Val(InputBox("Оценка " + Str(j + 1) + "-я " + Str(i) + "-го
студента", " ", Str(st.Оценки(j))))
            Next
            FilePut(1, st, i)
        Next
```

```
        FileClose()
    End Sub

    Public Sub СрБалл(ByVal fn As String, ByVal fn1 As String, ByVal gruppa As String)
        Dim srball As Single, n As Integer
        srball = 0
        FileOpen(1, fn, OpenMode.Random, , , 50)
        n = LOF(1) \ 50 + 1
        Do Until EOF(1)
            FileGet(1, st)
            srball = srball + st.СреднийБалл
        Loop
        FileClose()

        srball = srball / n
        FileOpen(1, fn, OpenMode.Random, , , 50)
        FileOpen(2, fn1, OpenMode.Random, , , 50)
        Do Until EOF(1)
            FileGet(1, st)
            If st.Группа = gruppa And st.СреднийБалл > srball Then
                FilePut(2, st)

            End If
        Loop
        FileClose()
    End Sub
End Module
```

Список литературы

1. Волчёнков Н.Г. Программирование на Visual Basic 6: В 3-х ч. Часть 1. – М.: ИНФРА-М, 2000. – 286 с.
2. Шевякова Д.А., Степанов А.М., Карпов Р.Г. Самоучитель Visual Basic 2005 / под общ. ред. А.Ф. Тихонова. – СПб.: БХВ-Петербург, 2007. – 576 с.
3. Богданов М.Р. Visual Basic 2005 на примерах. – СПб.: БХВ-Петербург, 2007. – 592 с.
4. Дукин А.Н., Пожидаев А.А. Самоучитель Visual Basic 2010. – СПб.: БХВ-Петербург 2010 . – 560 с
5. Сафронов И.К. Visual Basic в задачах и примерах. – СПб.: БХВ-Петербург, 2008.–401 с.