

Microsoft® Small Basic

Ввод и вывод файлов

Предполагаемое время работы с этим уроком: 1 час

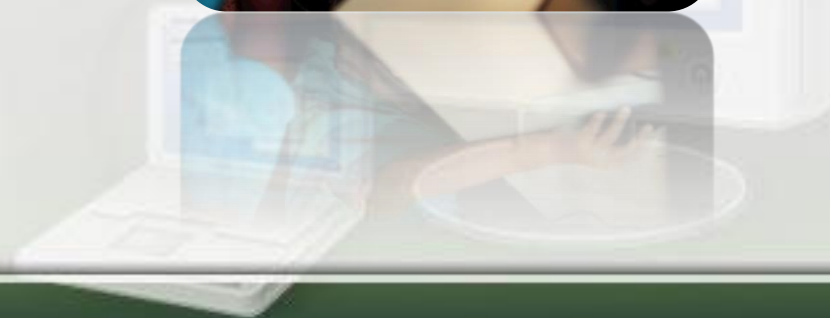


Ввод и вывод файлов

В этом уроке вы изучите следующее.

Использование различных свойств объекта **File**.

Использование различных операций объекта **File**.



Объект File

Компьютерный файл — это набор данных, сохраненных на компьютере. В Small Basic можно работать с внешними файлами из программы.

С помощью объекта **File** в Small Basic можно получать доступ к информации из файла, хранящегося на компьютере. Также можно выполнять чтение и запись в файл.

Объект **File** включает следующие операции и свойства.

CreateDirectory

GetDirectories

WriteLine

AppendContents

ReadContents

CopyFile

GetFiles

LastError

DeleteDirectory

Операции объекта File

Как вы видите, объект **File** позволяет работать с файлами множеством способов. Давайте познакомимся с некоторыми операциями объекта **File**...

❖ WriteLine

Можно записать строку текста в строку, номер которой указан в файле с помощью операции **WriteLine**.

```
File.WriteLine("C:\Small Basic.txt", 1, "Привет")
```

❖ AppendContents

Можно добавить указанный текст в конец файла, используя операцию **AppendContents**.

```
File.AppendContents("C:\Small Basic.txt", "Пока")
```

❖ ReadContents

Можно прочитать все содержимое файла, используя операцию **ReadContents**.

```
File.ReadContents("C:\Small Basic.txt")
```

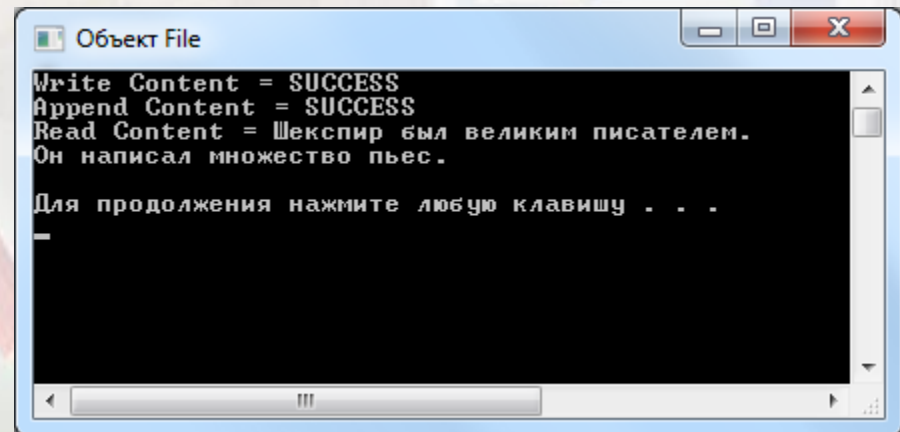
Операции объекта File

Напишем программу, чтобы лучше понимать эти операции.

```
FilePath = "C:\temp\my.txt"  
TextWindow.WriteLine("Write Content = " + File.WriteLine(FilePath, 1, "Шекспир был великим писателем."))  
TextWindow.WriteLine("Append Content = " + File.AppendContents(FilePath, "Он написал множество пьес."))  
TextWindow.WriteLine("Read Content = " + File.ReadContents(FilePath))
```

В этом примере указывается путь к файлу, а затем в файл записывается предложение с использованием операции **WriteLine**. Далее к существующему содержимому добавляется предложение с помощью операции **AppendContents**. Наконец, выполняется чтение всего содержимого файла с помощью операции **ReadContents**.

ВЫВОД



```
Объект File  
Write Content = SUCCESS  
Append Content = SUCCESS  
Read Content = Шекспир был великим писателем.  
Он написал множество пьес.  
Для продолжения нажмите любую клавишу . . .  
-
```

Операции объекта File

❖ CopyFile

Можно скопировать указанный файл в место назначения, используя операцию **CopyFile**.

```
File.CopyFile("C:\Small Basic.txt", "C:\temp")
```

❖ GetFiles

Можно получить список всех файлов в указанном каталоге, используя операцию **GetFiles**.

```
File.GetFiles("C:\Documents and Settings")
```

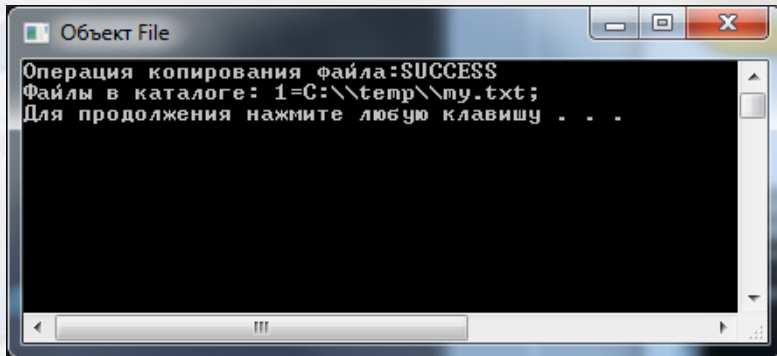


Операции объекта File

Напишем программу, чтобы лучше понимать эти операции.

```
sourcefilepath = "C:\temp\my.txt"  
destinationfilepath = "C:\temp\Move"  
directorypath = "C:\temp\  
TextWindow.WriteLine("Операция копирования файла:" + File.CopyFile(sourcefilepath, destinationfilepath))  
TextWindow.WriteLine("Файлы в каталоге: " + File.GetFiles(directorypath))
```

ВЫВОД



```
Объект File  
Операция копирования файла:SUCCESS  
Файлы в каталоге: 1=C:\\temp\\my.txt;  
Для продолжения нажмите любую клавишу . . .
```

В этом примере указанный исходный файл копируется в указанное назначение с помощью операции **CopyFile**. Также указывается путь к каталогу, после чего пути ко всем файлам отображаются в окне вывода с помощью операции **GetFiles**.

Операции объекта File

❖ CreateDirectory

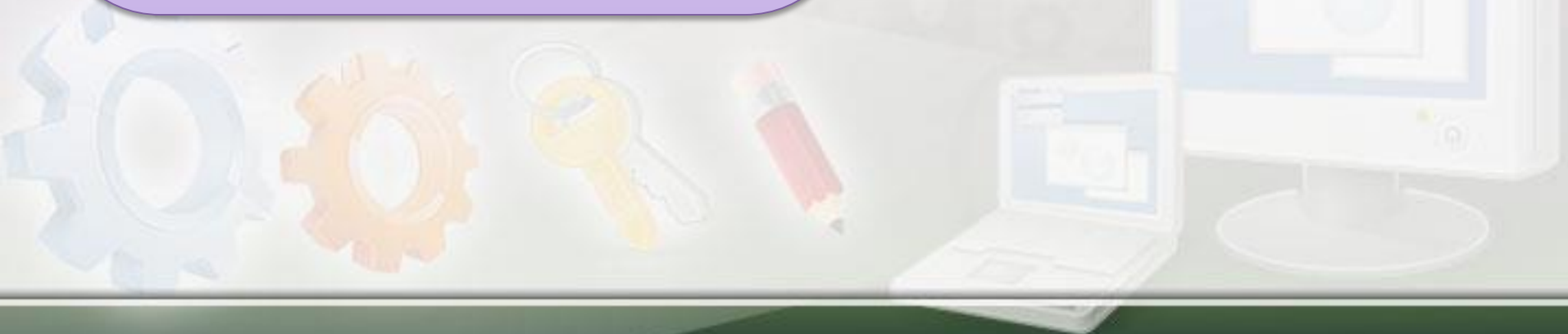
С помощью этой операции можно создать каталог с указанным именем в указанном назначении.

```
File.CreateDirectory("C:\File Object")
```

❖ GetDirectories

Эта операция позволяет получить пути ко всем каталогам в указанном пути каталога.

```
File.GetDirectories("C:\Documents and Settings")
```

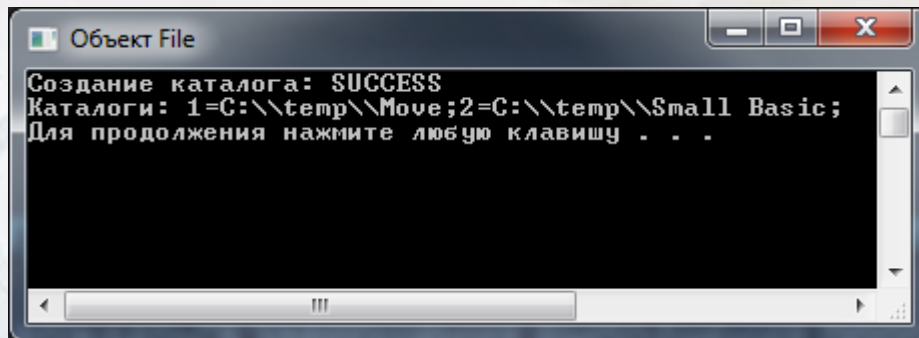


Операции объекта File

Посмотрим, как можно применить эти операции...

```
directorypath1 = "C:\\temp\\Small Basic"  
TextWindow.WriteLine("Создание каталога: " + File.CreateDirectory(directorypath1))  
directorypath2 = "C:\\temp"  
TextWindow.WriteLine("Каталоги: " + File.GetDirectories(directorypath2))
```

ВЫВОД



```
Объект File  
Создание каталога: SUCCESS  
Каталоги: 1=C:\\\\temp\\\\Move;2=C:\\\\temp\\\\Small Basic;  
Для продолжения нажмите любую клавишу . . .
```

Сначала создается каталог с помощью операции **CreateDirectory**.

Далее вы получаете путь ко всем каталогам в указанном расположении, используя операцию **GetDirectories**.

Свойство LastError

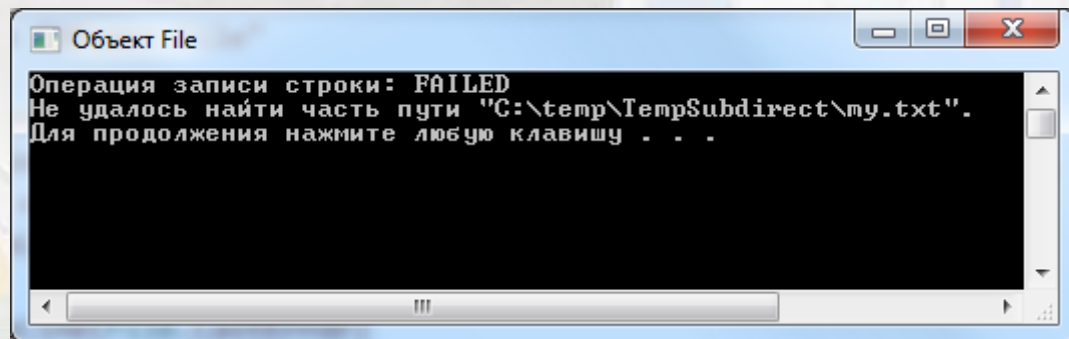
Используя свойство **LastError**, можно получить сведения о последней ошибке операции с файлом, произошедшей в программе. Это свойство довольно полезно, если ошибка препятствует выполнению программой операции с файлом.

В этом примере вы записываете текст в файл в строку с указанным номером, используя операцию **WriteLine** объекта **File**.

Далее вы получаете сведения о действительной ошибке в программе, если она существует, используя свойство **LastError** объекта **File**.

```
FilePath = "C:\temp\TempSubdirect\my.txt"|
TextWindow.WriteLine("Операция записи строки: " + File.WriteLine(FilePath, 1, "Как дела?"))
If File.LastError = "" Then
    TextWindow.WriteLine("Операция успешно выполнена")
Else
    TextWindow.WriteLine(File.LastError)
EndIf
```

ВЫВОД



```
Объект File
Операция записи строки: FAILED
Не удалось найти часть пути "C:\temp\TempSubdirect\my.txt".
Для продолжения нажмите любую клавишу . . .
```



Поздравляем! Вы изучили следующее.

- + Использование различных свойств объекта **File**.
- + Использование различных операций объекта **File**.



Продемонстрируйте свои знания

Написание программы, выполняющей следующие действия:

- ❖ Запрос у пользователя подходящего имени для каталога и создание каталога с этим именем.
- ❖ Загрузка файла из сети и его копирование в новый каталог.
- ❖ Отображение содержимого загруженного файла в текстовом окне.
- ❖ Получение от пользователя дополнительного содержимого и его добавление к файлу.
- ❖ Отображение итогового содержимого файла в текстовом окне.

