***Лабораторна робота №6***

***«Використання об’єкту FileSystemObject для роботи з файловою системою. Методи роботи з файлами та каталогами у сценаріях WSH»***

***(Термін виконання – 2 тижні!)***

Найбільшу частку виконуваних в процесі адміністрування і обслуговування комп'ютера операцій складають дії над об'єктами файлової системи: створення, читання і зміна вмісту файлів і папок, їх копіювання та видалення, отримання інформації про диски, папки, файли і т.п. Скрипти **WSH** можуть виконувати подібні операції можуть за допомогою об'єкта ***FileSystemObject***. Цей об'єкт не належить до **WSH**, а є стороннім **ActiveX**-об'єктом. Але, використовуючи метод ***CreateObject***, скрипт **WSH** може створювати подібні об'єкти і використовувати багату функціональність ***FileSystemObject***:

***var FSO = WScript.CreateObject(“Scripting.FileSystemObject”);***

Об'єкт ***FileSystemObject*** надає в розпорядження програміста об'єкти і колекції, перераховані в табл. 9.

Табл. 9. Об'єкти і колекції для роботи з файловою системою

|  |  |
| --- | --- |
| **Об’єкт/Колекція** | **Опис** |
| ***FileSystemObject*** | **Основний об'єкт.** Містить методи і властивості, які забезпечують доступ до файлової системи комп'ютера: створюють, видаляють, отримують інформацію, і керують дисками, папками та файлами. Багато методів цього об'єкта дублюються в інших об'єктах (наприклад, видалити файл можна як викликом ***DeleteFile*** цього об'єкта, так і викликом методу ***Delete*** об'єкта ***File***) |
| ***Drive*** | **Об'єкт.** Містить методи і властивості, які дозволяють збирати інформацію про накопичувачі, наявні в системі: ім'я диска, мітка тому, загальний розмір і обсяг вільного місця на диску. |
| ***Drives*** | **Колекція об'єктів типу *Drive*.** Включає в себе всі диски на даному комп'ютері незалежно від їх типу. |
| ***File*** | **Об'єкт.** Містить методи і властивості, які дозволяють виконувати основні операції з файлом: створювати, видаляти, або переміщувати файл, отримувати і змінювати атрибути. Властивості цього об'єкта зберігають характеристики файлу: розмір, ім'я, шлях до нього, атрибути. |
| ***Files*** | **Колекція об'єктів типу *File*.** Містить список всіх файлів, що знаходяться в заданій папці. |
| ***Folder*** | **Об'єкт.** Аналогічний об'єкту ***File***, але дозволяє отримати доступ до заданої папки. |
| ***Folders*** | **Колекція об'єктів типу *Folder.*** Містить список всіх папок, що знаходяться в заданій папці. |
| ***TextStream*** | **Об'єкт.** Дозволяє читати і писати інформацію в текстовий файл. |

Для маніпулювання з файлом необхідно створити пов'язаний з ним об'єкт ***File***. Для цього призначений метод об'єкта ***FileSystemObject*** (далі ***FSO***) ***GetFile***. Якщо файл не існує, він попередньо створюється методом ***FSO.CreateTextFile***. Наступний приклад демонструє застосування перерахованих методів для роботи з файлом.

***var FSO=WScript.CreateObject("Scripting.FileSystemObject");***

***var fname="d:\\test.txt"; // можна fname="d:/test.txt"***

***if (!FSO.FileExists(fname))***

***{***

***file=FSO.CreateTextFile("d:\\test.txt");***

***file.WriteLine("New file");***

***file.Close();***

***}***

***file=FSO.GetFile(fname);***

***file.Copy(fname+".copy");***

***//або так FSO.CopyFile(fname,fname+".copy");***

***file.Delete();***

Передостанній рядок скрипта демонструє, що для копіювання, переміщення і видалення файлів скористатися як методами об'єкта ***FSO***, так і об'єкта ***File*** (див. табл. 10).

Табл. 10. Методи для роботи з файлами

|  |  |
| --- | --- |
| **Дія** | **Метод** |
| **Переміщення** | ***File.Move*** або ***FileSystemObject.MoveFile*** |
| **Копіювання** | ***File.Copy*** або ***FileSystemObject.CopyFile*** |
| **Видалення** | ***File.Delete*** або ***FileSystemObject.DeleteFile*** |
| **Створення** | ***FSO.CreateTextFile*** |

Набір властивостей об'єкта ***File*** надає програмісту доступ до основних характеристик файлу (табл. 11).

Табл. 11. Властивості об’єкта ***File***

|  |  |
| --- | --- |
| **Властивість** | **Опис** |
| ***Attributes*** | Дозволяє переглянути або встановити атрибути файлу |
| ***DateCreated*** | Містить дату та час створення файлу. Доступно тільки для читання |
| ***DateLastAccessed*** | Містить дату та час останнього звернення. Доступно тільки для читання |
| ***DateLastModified*** | Містить дату та час останньої зміни. Доступно тільки для читання |
| ***Drive*** | Містить необхідну букву пристрою, на якому знаходиться файл. Доступно тільки для читання |
| ***Name*** | Дозволяє переглянути і змінити ім'я файлу. |
| ***ParentFolder*** | Містить об'єкт ***Folder*** для батьківського каталогу файлу. Доступно тільки для читання |
| ***Path*** | Містить шлях до файлу |
| ***ShortName*** | Містить коротку назву файлу |
| ***ShortPath*** | Містить шлях до файлу, що складається з коротких імен каталогів |
| ***Size*** | Повертає розмір файлу в байтах |
| ***Type*** | Повертає інформацію про тип файлу |

Наступний приклад дозволяє отримати розмір файлу запущеного скрипта і встановити його атрибут **Тільки для читання**.

***FSO = WScript.CreateObject("Scripting.FileSystemObject");***

***file1 = FSO.GetFile(WScript.ScriptFullName);***

***WScript.Echo("Розмір файлу скрипта" + file1.Name + ":" + file1.Size);***

***file1.Attributes = 1;***

Властивість ***Attributes*** є цілим числом, яке описує атрибути файлу згідно з табл. 12. Установка або читання окремого атрибута виконуються з використанням двійкової маски і порозрядних логічних операцій. У наведеному прикладі молодший біт властивості встановлюється в ***1***, що відповідає установці атрибуту **Тільки для читання**.

Табл. 12. Атрибути файлу у властивості ***Attributes*** об'єкта ***File***

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Значення** | **Опис** |
| ***Normal*** | ***0*** | Звичайний файл без встановлених атрибутів |
| ***Readonly*** | ***1*** | Файл з атрибутом «лише для читання» (1-й біт) |
| ***Hidden*** | ***2*** | Прихований файл (2-й біт) |
| ***System*** | ***4*** | Системний файл (3-й біт) |
| ***Directory*** | ***16*** | Папка або каталог (можливо, з атрибутом «лише для читання») (5-й біт) |
| ***Archive*** | ***32*** | Файл з атрибутом «архівний» (6-й біт) |
| ***Alias*** | ***1024*** | Ярлик (.Lnk-файл) (11-й біт) |
| ***Compressed*** | ***2048*** | Стиснутий файл (12-й біт) |

Для роботи з вмістом файла використовується об'єкт ***TextStream*** і його методи.

Типовими операціями для цього об'єкта є читання і запис інформації з / у файл.

Відкриття файлу (і створення об'єкта типу ***TextStream***) може здійснюватися методами ***CreateTextFile***, ***OpenTextFile*** і ***OpenAsTextStream***. Метод ***OpenAsTextStream*** належить об'єкту ***File***, він відкриває вже існуючий файл, для якого раніше був отриманий об'єкт ***File*** (методом ***GetFile***). Методи ***CreateTextFile*** і ***OpenTextFile*** визначені в об'єктах ***FSO*** і ***Folder***, вони можуть відкрити файл на ім'я, причому, якщо файл, що відкривається не існує, можуть його створити. Обидва об'єкти повертають посилання на об'єкт типу ***TextStream***.

Синтаксис виклику методу ***OpenTextFile***:

***FSO.OpenTextFile(filename[,iomode[,create[,format]]])***, де

* ***filename*** - ім'я файлу
* ***iomode*** - режим відкриття файлу, який може приймати значення:
* ***1*** - файл відкривається для читання, запис заборонений;
* ***2*** - файл відкривається для запису; якщо файл існує, його вміст видаляється;
* ***8*** - файл відкривається для додавання даних:
* ***create*** - логічна величина, що вимагає (при значенні **true**) створювати новий файл, якщо файлу з вказаним ім'ям не існує.
* ***format*** - кодування для відкриття файлу, може приймати значення:
* ***-2*** - відкривається з використанням системної кодування;
* ***-1*** - відкривається в кодуванні Unicode;
* ***0*** - відкривається в кодуванні ASCII.

***var fso = WScript.CreateObject("Scripting.FileSystemObject")***

***var ts = fso.OpenTextFile("d:\\testfile.txt", 1, True)***

Властивості і методи об'єкта TextStream наведені в табл. 13.

Табл. 13. Властивості і методи об'єкта ***TextStream***

|  |  |
| --- | --- |
| **Властивість** | **Опис** |
| ***Attributes*** | Дозволяє переглянути або встановити атрибути файлу |
| ***DateCreated*** | Містить дату та час створення файлу. Доступно тільки для читання |
| ***DateLastAccessed*** | Містить дату та час останнього звернення. Доступно тільки для читання |
| ***DateLastModified*** | Містить дату та час останньої зміни. Доступно тільки для читання |
| ***Drive*** | Містить необхідну букву пристрою, на якому знаходиться файл. Доступно тільки для читання |
| ***Name*** | Дозволяє переглянути і змінити ім'я файлу |
| ***ParentFolder*** | Містить об'єкт Folder для батьківського каталогу файлу. Доступно тільки для читання |
| ***Path*** | Містить шлях до файлу |
| ***ShortName*** | Містить коротку назву файлу |
| ***ShortPath*** | Містить шлях до файлу, що складається з коротких імен каталогів |
| ***Size*** | Повертає розмір файлу в байтах |
| ***Type*** | Повертає інформацію про тип файлу |
| **Метод** | **Опис** |
| ***Write(String)*** | Записує дані в файл в один рядок |
| ***WriteLine(String)*** | Записує дані, з переходом на новий рядок |
| ***WriteBlankLines()*** | Записує порожній рядок |
| ***Read(nChars)*** | Читає з файлу ***nChars*** символів |
| ***ReadLine()*** | Читає рядок з файлу |
| ***ReadAll()*** | Зчитує весь файл |
| ***Skip(nChars)*** | Пропускає ***nChars*** символів |
| ***SkipLine()*** | Пропускає рядок |
| ***Close()*** | Закриває відкритий файл |

Розглянемо приклад, в якому для вирішення конкретного завдання використовуються перераховані методи і властивості об'єкта ***TextStream***. У наступному прикладі скрипт перебирає всі рядки файлу, ім'я якого передано йому в якості першого параметра, і виводить у вікно (разом з порядковим номером) тільки ті з них, в яких немає рядка "***rem***".

***var FSO = WScript.CreateObject("Scripting.FileSystemObject");***

***if (WScript.Arguments.length>0 && FSO.FileExists(WScript.Arguments(0)))***

***{***

***var TextStream = FSO.OpenTextFile(WScript.Arguments(0));***

***var n=1;***

***while (!TextStream.AtEndOfStream)***

***{***

***Str = TextStream.ReadLine();***

***if (Str.indexOf("rem ")==-1)***

***WScript.Echo(n+" "+Str);***

***n++;***

***}***

***TextStream.Close();***

***}***

***else***

***WScript.Echo("Використання скрипта:”+WScript.ScriptFullName + ”ім’я\_файлу");***

Для роботи з папками і їх властивостями ***FSO*** надає об'єкт ***Folder*** і колекцію ***Folders***. Об'єкт ***Folder*** дозволяє працювати із заданою папкою. Багато його методів і властивості схожі за призначенням з однойменними в об'єкті ***File***: методи ***Copy***, ***Move***, ***Delete***, властивості ***DateCreated***, ***DateLastAccessed***, ***DateLastModified***, ***Size***, ***Name***, ***Path***, ***Parent*** і т.д. Відповідно, принципи роботи з папками аналогічні роботі з файлами (наприклад, поряд з методом ***FileExist*** ***FSO*** має метод ***FolderExist***, а поряд з методом ***GetFile*** можна використовувати метод ***GetFolder***). З властивостей, характерних для об'єкта ***Folder*** необхідно виділити колекції ***Files*** і ***SubFolders***, що містять списки файлів і папок заданої папки. З їх допомогою можна перебирати вміст папки.

Наступний приклад дозволяє вивести у вікно список програм, що автоматично запускаються, з меню **Автозавантаження**:

***var WSHShell = WScript.CreateObject("WScript.Shell");***

***var fso = WScript.CreateObject("Scripting.FileSystemObject");***

***var StartFolder=fso.GetFolder(WSHShell.SpecialFolders("Startup"));***

***var enFiles=new Enumerator(StartFolder.Files);***

***WScript.Echo ("Список автозавантаження");***

***for(; !enFiles.atEnd(); enFiles.moveNext())***

***{***

***var cFile=enFiles.item();***

***WScript.Echo(cFile.Name);***

***}***

Наступний приклад демонструє, як можна рекурсивно обійти всі папки заданої папки (3-й параметр) і знайти в них файли з заданим розширенням (1-й параметр скрипта), які змінювалися в останній раз в заданий місяць (2-й параметр).

***FSO = WScript.CreateObject("Scripting.FileSystemObject");***

***// рекурсивна функція перебору вмісту папки***

***function findFiles(cFolder, cExt, cDateMonth)***

***{***

***var newFolder=true;***

***// перебираємо всі файли чергової папки***

***var cFiles=new Enumerator(cFolder.Files);***

***for(;!cFiles.atEnd(); cFiles.moveNext())***

***{***

***var cFile=cFiles.item();***

***// Створюємо об'єкт типу date c датою останньої модифікації файлу***

***var dateFile=new Date(cFile.DateLastModified);***

***if(FSO.GetExtensionName(cFile.Name)==cExt &&***

***dateFile.getMonth()==cDateMonth)***

***{***

***if(newFolder)***

***{***

***newFolder=false;***

***WScript.Echo("---- Папка: "+cFolder.Path+" -------");***

***}***

***WScript.Echo(cFile.Name);***

***}***

***}***

***// перебираємо всі підпапки і для кожної рекурсивно викликаємо findFiles***

***var cSubFlds=new Enumerator(cFolder.SubFolders);***

***for(;!cSubFlds.atEnd(); cSubFlds.moveNext())***

***{***

***var cSFld=cSubFlds.item();***

***findFiles(cSFld, cExt, cDateMonth);***

***}***

***}***

***//точка входу у скрипт***

***if(WScript.Argiments.length==3)***

***{***

***var FileExt=WScript.Arguments(0);***

***var FileMonth=WScript.Arguments(1);***

***var workFolder=FSO.GetFolder(WScript.Arguments(2));***

***findFiles(workFolder,FileExt,FileMonth);***

***}***

***else***

***{***

***WScript.Echo(“Использование скрипта:” + WScript.ScriptFullName + ” расширение месяц папка”);***

***}***

***Завдання***

1. Для всіх файлів заданої папки, що мають розширення txt, встановити атрибути Тільки для читання і Прихований.
2. Напишіть скрипт, який при кожному запуску буде дописувати в текстовий файл інформацію про поточну дату і час, а також розмірі папки, в якій розташований скрипт і кількості файлів в ній.
3. Написати скрипт, який би розбирав вміст папки, що задається як параметр запуску, і створював би папки відповідно до роками створення файлів (2012, 2011 і т.п.), всередині кожної з цих папок створити папки з назвами місяців (січень , лютий, ..., грудень). Кожен файл зазначеної папки перенести в відповідну йому на поточний рік і місяця створення папку.

***Контрольні питання***

1. Які дії можна виконувати за допомогою об’єкта FileSystemObject у WSH?
2. Які об’єкти-колекції використовуються для роботи з файловою системою у WSH?
3. Як у сценарії WSH отримати інформацію про заданий каталог або файл?
4. Як у сценарії WSH перевірити існування певного диску, файлу або каталогу?
5. Як у сценарії WSH скопіювати файли або каталоги?
6. Як у сценарії WSH перемістити файли або каталоги?
7. Як у сценарії WSH відкрити текстовий файл для читання, запису або додавання?