**ЛЕКЦИЯ. ЯЗЫК СКРПТОВЫХ СЦЕНАРИЕВ WSH**

***Назначение и основные свойства WSH***

WSH предъявляет минимальные требования к объему оперативной памяти, и является очень удобным инструментом для автоматизации повседневных задач пользователей и администраторов операционной системы Windows. Используя сценарии WSH, можно непосредственно работать с файловой системой компьютера, а также управлять работой других приложений (серверов автоматизации). При этом возможности сценариев ограничены только средствами, которые предоставляют доступные серверы автоматизации.

Перечислим только наиболее очевидные задачи, для автоматизации которых прекрасно подходят сценарии WSH.

 Организация резервного копирования на сетевой сервер файлов с локальной машины, которые отбираются по какому-либо критерию.

 Быстрое изменение конфигурации рабочего стола Windows в зависимости от задач, выполняемых пользователем.

 Автоматический запуск программ Microsoft Office, создание там сложных составных документов, распечатка этих документов и закрытие приложений.

 Управление работой приложений, не являющихся серверами автоматизации, с помощью посылки в эти приложения нажатий клавиш.

 Подключение и отключение сетевых ресурсов (дисков и принтеров).

 Создание сложных сценариев регистрации для пользователей.

 Выполнение задач администрирования локальной сети (например, добавление или удаление пользователей).

***Создание и запуск простейших сценариев WSH***

Простейший WSH-сценарий, написанный на языке JScript или VBScript — это обычный текстовый файл с расширением js или vbs

соответственно, создать его можно в любом текстовом редакторе, способном сохранять документы в формате "Только текст".

Размер сценария может изменяться от одной до тысяч строк, предельный размер ограничивается лишь максимальным размером файла в соответствующей файловой системе.

В качестве первого примера создадим JScript-сценарий, выводящий на экран диалоговое окно с надписью "Привет!". Для этого достаточно c помощью, например, стандартного Блокнота Windows (notepad.exe) создать файл First.js, содержащий всего одну строку:

WScript.Echo("Привет!");

Тот же самый сценарий на языке VBScript, естественно, отличается синтаксисом и выглядит следующим образом:

WScript.Echo "Привет!"

Несмотря на то, что для работы этих двух сценариев достаточно всего одной строки, желательно сразу приучить себя к добавлению в начало файла информации о находящемся в нем сценарии: имя файла, используемый язык, краткое описание выполняемых действий. На языке JScript такая информация, оформленная в виде комментариев, может выглядеть следующим образом:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Имя: First.js \*/

/\* Язык: JScript \*/

/\* Описание: Вывод на экран приветствия \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

На языке VBScript то же самое выглядит следующим образом:

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Имя: First.vbs

' Язык: VBScript

' Описание: Вывод на экран приветствия

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Для запуска сценариев WSH существует несколько способов.

**Запуск сценария из командной строки в консольном режиме**

Можно выполнить сценарий из командной строки с помощью консольной версии WSH cscript.exe. Например, чтобы запустить сценарий, записанный в файле

C:\Script\First.js,

нужно загрузить командное окно и выполнить в нем команду

cscript C:\Script\First.js

В результате выполнения этого сценария в командное окно выведется строка "Привет!"

**Запуск сценария из командной строки в графическом режиме**

Сценарий можно выполнить из командной строки с помощью (оконной) графической версии WSH wscript.exe. Для нашего примера в этом случае нужно выполнить команду

wscript C:\Script\First.js

Тогда в результате выполнения сценария на экране появится нужное нам диалоговое окно.

Таким образом, мы видим, что при запуске сценария в консольном режиме, вывод текстовой информации происходит в стандартный выходной поток (на экран), при запуске в графическом режиме — в диалоговое окно.

**Запуск сценария с помощью меню *Пуск***

Для запуска сценария с помощью пункта **Выполнить** (Run) меню **Пуск** (Start), достаточно написать полное имя этого сценария в поле **Открыть** (Open) и нажать кнопку **Ok**. В этом случае по умолчанию сценарий будет выполнен с помощью wscript.exe, т. е. вывод информации будет вестись в графическое диалоговое окно.

**Запуск сценария с помощью Проводника Windows (Windows Explorer)**

Самым простым является запуск сценария в окнах Проводника Windows или на рабочем столе — достаточно просто выполнить двойной щелчок мышью на имени файла со сценарием или на его значке (аналогично любому другому исполняемому файлу). При этом, как и в случае запуска с помощью меню **Пуск** (Start), сценарий по умолчанию выполняется с помощью wscript.exe.

В WSH входят перечисленные ниже объекты:

 WScript. Это главный объект WSH, который служит для создания других объектов или связи с ними, содержит сведения о сервере сценариев, а также позволяет вводить данные с клавиатуры и выводить информацию на экран или в окно Windows.

 WshArguments. Обеспечивает доступ ко всем параметрам командной строки запущенного сценария или ярлыка Windows.

 WshNamed. Обеспечивает доступ к именным параметрам командной строки запущенного сценария.

 WshUnnamed. Обеспечивает доступ к безымянным параметрам командной строки запущенного сценария.

 WshShell. Позволяет запускать независимые процессы, создавать ярлыки, работать с переменными среды, системным реестром и специальными папками Windows.

 WshSpecialFolders. Обеспечивает доступ к специальным папкам Windows.

 WshShortcut. Позволяет работать с ярлыками Windows.

 WshUrlShortcut. Предназначен для работы с ярлыками сетевых ресурсов.

 WshEnvironment. Предназначен для просмотра, изменения и удаления переменных среды.

 WshNetwork. Используется при работе с локальной сетью: содержит сетевую информацию для локального компьютера, позволяет подключать сетевые диски и принтеры.

 WshScriptExec. Позволяет запускать консольные приложения в качестве дочерних процессов, обеспечивает контроль состояния этих приложений и доступ к их стандартным входным и выходным потокам.

 WshController. Позволяет запускать сценарии на удаленных машинах.

 WshRemote. Позволяет управлять сценарием, запущенным на удаленной машине.

 WshRemoteError. Используется для получения информации об ошибке, возникшей в результате выполнения сценария, запущенного на удаленной машине.

Кроме этого, имеется объект FileSystemObject, обеспечивающий доступ к файловой системе компьютера

**Метод Sleep**

В следующем примере сценарий переводится в неактивное состояние на 5 секунд:

**Использование метода WScript.Sleep**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\* Имя: SleepExample.vbs

'\* Язык: VBScript

'\* Описание: Использование метода WScript.Sleep

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WScript.Echo "Сценарий запущен, отдыхаем…"

WScript.Sleep 5000

WScript.Echo "Выполнение сценария завершено"

***Объект* WshShell**

С помощью объекта WshShell можно запускать новый процесс, создавать ярлыки, работать с системным реестром, получать доступ к переменным среды и специальным папкам Windows. Создается этот объект следующим образом:

var WshShell=WScript.CreateObject("WScript.Shell");

**Объект WshEnvironment**

Объект WshEnvironment позволяет получить доступ к коллекции, содержащей переменные среды заданного типа (переменные среды операционной системы, переменные среды пользователя или переменные среды текущего командного окна). Этот объект можно создать с помощью свойства Environment объекта WshShell или одноименного его метода:

Set WshShell=WScript.CreateObject("WScript.Shell")

Set WshSysEnv=WshShell.Environment

Set WshUserEnv=WshShell.Environment("User")

Объект WshEnvironment имеет свойство Length, в котором хранится число элементов в коллекции (количество переменных среды), и методы Count и Item. Для того чтобы получить значение определенной переменной среды, в качестве аргумента метода Item указывается имя этой переменной в двойных кавычках. В следующем примере мы выводим на экран значение переменной среды PATH

' Описание: Работа с переменными среды

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim WshShell, WshSysEnv

Set WshShell=WScript.CreateObject("WScript.Shell")

Set WshSysEnv=WshShell.Environment

WScript.Echo "Системный путь:",WshSysEnv.Item("PATH")

**Получение сведений о диске**

Доступ к свойствам заданного локального или сетевого диска можно получить с помощью объекта Drive, который возвращается методом GetDrive объекта FileSystemObject, а также может быть получен как элемент коллекции Drives.

Ниже приведен сценарий DriveInfo.vbs, который выводит на экран некоторые свойства диска С.

**Вывод информации о диске (VBScript)**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Имя: DriveInfo.vbs

' Язык: VBScript

' Описание: Вывод на экран свойств диска C

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Объявляем переменные

Dim FSO,D,TotalSize,FreeSpace,s

'Создаем объект FileSystemObject

Set FSO = WScript.CreateObject("Scripting.FileSystemObject")

'Создаем объект Drive для диска C

Set D = FSO.GetDrive("C:")

s = "Информация о диске C:" & VbCrLf

'Получаем серийный номер диска

s = s &"Серийный номер: " & D.SerialNumber & VbCrLf

'Получаем метку тома диска

s = s & "Метка тома: " & D.VolumeName & VbCrLf

'Вычисляем общий объем диска в килобайтах

TotalSize = D.TotalSize/1024

s = s & "Объем: " & TotalSize & " Kb" & VbCrLf

'Вычисляем объем свободного пространства диска в килобайтах

FreeSpace = D.FreeSpace/1024

s = s & "Свободно: " & FreeSpace & " Kb" & VbCrLf

'Выводим свойства диска на экран

WScript.Echo s

'\*\*\*\*\*\*\*\*\*\*\*\*\* Конец \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Примеры других сценариев.

**Вывод информации о файле (VBScript)**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Имя: FileInfo.vbs

' Язык: VBScript

' Описание: Вывод на экран некоторых свойств файла

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim FSO,F,s 'Объявляем переменные

'Создаем объект FileSystemObject

Set FSO = WScript.CreateObject("Scripting.FileSystemObject")

'Создаем объект File

Set F = FSO.GetFile("C:\boot.ini")

'Получаем имя файла

s = "Файл: " & F.Name & VbCrLf

'Получаем дату создания файла

s = s & "Дата создания: " & F.DateCreated & VbCrLf

'Получаем тип файла

s = s & "Тип: " & F.Type & VbCrLf

'Выводим информацию на экран

WScript.Echo s

'\*\*\*\*\*\*\*\*\*\*\*\*\* Конец \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Создание нового каталога (VBScript)**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Имя: MakeFolder.vbs

' Язык: VBScript

' Описание: Создание нового каталога

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Объявляем переменные

Dim FSO, F, SubFolders

'Создаем объект FileSystemObject

Set FSO = WScript.CreateObject("Scripting.FileSystemObject")

'Создаем каталог C:\Program Files\Новая папка

FSO.CreateFolder("C:\Program Files\Новая папка")

'Создаем объект Folder для каталога C:\Program Files

Set F = FSO.GetFolder("C:\Program Files")

'Создаем коллекцию подкаталогов каталога C:\Program Files

Set SubFolders = F.SubFolders

'Создаем каталог C:\Program Files\Еще одна новая папка

SubFolders.Add "Еще одна новая папка"

'\*\*\*\*\*\*\*\*\*\*\*\*\* Конец \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Создание текстового файла**

Для создания текстового файла используется метод CreateTextFile объекта FileSystemObject, который имеет один обязательный текстовый параметр (путь к создаваемому файлу) и два необязательных логических параметра (Overwrite и Unicode).

Параметр Overwrite имеет значение в том случае, когда создаваемый файл уже существует. Если Overwrite равно True, то такой файл перепишется (старое содержимое будет утеряно), если же в качестве

Overwrite указано False, то файл переписываться не будет. Если этот параметр вообще не указан, то существующий файл также не будет переписан.

Параметр Unicode указывает, в каком формате (ASCII или Unicode) следует создавать файл. Если Unicode равно True, то файл создается в формате Unicode, если же Unicode равно False или этот параметр вообще не указан, то файл создается в режиме ASCII.

**Запись информации в текстовый файл и чтение из него (VBScript)**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Имя: TextFile.vbs

' Язык: VBScript

' Описание: Запись строк в текстовый файл и чтение из него

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim FSO,F,TextStream,s 'Объявляем переменные

' Инициализируем константы

Const ForReading = 1, ForWriting = 2, TristateUseDefault = -2

' Создаем объект FileSystemObject

Set FSO=WScript.CreateObject("Scripting.FileSystemObject")

' Создаем в текущем каталоге файл test1.txt

FSO.CreateTextFile "test1.txt"

' Создаем объект File для файла test1.txt

set F=FSO.GetFile("test1.txt")

' Создаем объект TextStream (файл открывается для записи)

Set TextStream=F.OpenAsTextStream(ForWriting, TristateUseDefault)

' Записываем в файл строку

TextStream.WriteLine "Это первая строка"

' Закрываем файл

TextStream.Close

' Открываем файл для чтения

Set TextStream=F.OpenAsTextStream(ForReading, TristateUseDefault)

' Считываем строку из файла

s=TextStream.ReadLine

' Закрываем файл

TextStream.Close

' Отображаем строку на экране

WScript.Echo "Первая строка из файла test1.txt:" & vbCrLf & vbCrLf & s

'\*\*\*\*\*\*\*\*\*\*\*\*\* Конец \*\*\*\*\*\*\*\*\*\*

***Запуск приложений Windows***

Из сценария WSH запустить другое приложение можно с помощью методов Run или Exec объекта WshShell.

При использовании метода Run для запускаемого приложения можно задать тип окна (при условии, что приложение поддерживает этот тип). Например, в результате выполнения следующих двух строк VBScript-кода:

Set WshShell = WScript.CreateObject("WScript.Shell")

WshShell.Run "notepad",3

программа Блокнот (notepad.exe) будет запущена в максимизированном (развернутом на весь экран) окне.

***Замечание***

Метод Run всегда создает новый экземпляр запускаемого процесса, с его помощью нельзя ни повторно активизировать окно запущенного приложения (для этого используется метод AppActivate), ни свернуть или развернуть его.

Другим вариантом запуска из сценария приложения Windows, является применение метода Exec. Этод метод запускает приложение, путь к которому указан как параметр метода, и возвращает объект WshScriptExec.

Например:

Set WshShell = WScript.CreateObject("WScript.Shell")

Set theNotepad = WshShell.Exec("calc");

***Замечание***

При подобном запуске приложения, в отличие от метода Run, нельзя задать тип окна.

Объект WshScriptExec позволяет контролировать ход выполнения запущенного приложения с помощью свойства Status — если Status равен 0, то приложение выполняется, если Status равен 1, то приложение завершено. Кроме этого, используя метод Terminate, можно принудительно завершить работу того приложения, которому соответствует объект WshScriptExec.

Ниже приведен сценарий на языке VBScipt, в котором с помощью метода Exec запускается Блокнот (notepad.exe); ссылка на соответствующий объект WshScriptExec сохраняется в переменной theNotepad:

Set theNotepad = WshShell.Exec("notepad")

После этого выполнение сценария приостанавливается на 1 секунду (пауза необходима, для того чтобы окно Блокнота успело появиться на экране), после чего выводится диалоговое окно с информацией о статусе

***Переключение между приложениями, имитация нажатий клавиш***

Производить переключение между окнами нескольких запущенных приложений позволяет метод AppActivate объекта WshScript. В качестве аргумента этого метода нужно указывать либо заголовок активизируемого окна, либо идентификатор процесса (Process ID, PID), который запущен в данном окне. Предпочтительным является использование PID, получаемого с помощью свойства ProcessID объекта WshScriptExec, соответствующего активизируемому приложению. Недостатки применения заголовка окна в методе AppActivate:

 при написании сценария необходимо знать точное название заголовка;

 само приложение может изменить текст в заголовке окна;

 в случае нескольких окон с одинаковыми заголовками AppActivate всегда будет активизировать один и тот же экземпляр, доступ к другим окнам получить не удастся.

Активизировав то или иное окно, в котором выполняется приложение Windows, можно из сценария сымитировать нажатия клавиш в этом окне. Для этого используется метод SendKeys объекта WshShell.

Рассмотрим пример сценария Run&ExecWinApp.vbs, в котором запускается Калькулятор (calc.exe) и в его окно с помощью SendKeys последовательно посылаются нажатия клавиш <1>, <+>, <2> и <Enter>:

Set theCalculator = WshShell.Exec("calc")

WScript.Sleep 500

WshShell.AppActivate theCalculator.ProcessID

WshShell.SendKeys "1{+}"

WshShell.SendKeys "2")

WshShell.SendKeys "~") ' Клавиша <Enter>

Затем выполнение сценария приостанавливается на 1 секунду, чтобы результат вычислений был виден на экране и результат вычислений (символ "3") копируется в буфер Windows с помощью "нажатия" клавиш <Ctrl>+<C>:

WshShell.SendKeys "^c"

После этого на экран выводится сообщение о том, что Калькулятор будет закрыт, в результате чего окно Калькулятора теряет фокус. Для того чтобы вновь активизировать это окно, используется метод AppActivate, параметром которого служит PID Калькулятора:

WshShell.AppActivate theCalculator.ProcessID

Для того чтобы закрыть окно Калькулятора, в него посылаются нажатия клавиш <Alt>+<F4>:

WshShell.SendKeys "%{F4}"

После закрытия Калькулятора запускается Блокнот (notepad.exe) и в него записываются результаты работы Калькулятора (вставка из буфера вычисленной суммы производится с помощью "нажатий" <Ctrl>+<V>):

WshShell.Run "notepad"

WScript.Sleep 1000

WshShell.AppActivate "notepad"

WshShell.SendKeys "1{+}2="

WshShell.SendKeys "^v"

WshShell.SendKeys " {(}c{)} Calculator"

запущенного приложения и вопросом о необходимости закрытия Блокнота. В случае утвердительного ответа происходит закрытие Блокнота с помощью метода Terminate.

**Листинг 5.1. Запуск и принудительное закрытие приложения (VBScript)**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Имя: ExecWinApp.vbs

' Язык: VBScript

' Описание: Запуск и закрытие приложение (объект WshScriptExec)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option Explicit

Dim WshShell,theNotepad,Res,Text,Title ' Объявляем переменные

' Создаем объект WshShell

Set WshShell = WScript.CreateObject("WScript.Shell")

WScript.Echo "Запускаем Блокнот"

' Запускаем приложение (создаем объект WshScriptExec)

Set theNotepad = WshShell.Exec("notepad")

WScript.Sleep 500 ' Приостанавливаем выполнение сценария

Text="Блокнот запущен (Status=" & theNotepad.Status & ")" & vbCrLf \_

& "Закрыть Блокнот?"

Title=""

' Выводим диалоговое окно на экран

Res=WshShell.Popup(Text,0,Title,vbQuestion+vbYesNo)

' Определяем, какая кнопка нажата в диалоговом окне

If Res=vbYes Then

theNotepad.Terminate ' Прерываем работу Блокнота

' Приостанавливаем выполнение сценария, для того чтобы Блокнот

' успел закрыться

WScript.Sleep 100

WScript.Echo "Блокнот закрыт (Status=" & theNotepad.Status & ")"

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\* Конец \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/