**Лекция 5**

**Собственная объектная модель WSH**

**Операционные системы**

(Методы и технологии управления ОС Windows на основе использования языка командной оболочки и сценариев)

Перейдем к описанию собственной объектной модели *Windows* *Script* *Host*. С помощью внутренних объектов WSH из сценариев можно выполнять следующие основные задачи:

* 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.

**Объект WScript**

|  |  |
| --- | --- |
| Таблица 5.1. Свойства объекта WScript | |
| **Свойство** | **Описание** |
| Application | Предоставляет интерфейс Idispatch для объекта WScript |
| Arguments | Содержит указатель на коллекцию WshArguments, содержащую параметры командной строки для исполняемого сценария |
| FullName | Содержит полный путь к исполняемому файлу сервера сценариев (в Windows XP обычно это C:\WINDOWS\SYSTEM32\CSCRIPT.EXE или C:\WINDOWS\SYSTEM32\WSCRIPT.EXE) |
| Name | Содержит название объекта Wscript (Windows Script Host) |
| Path | Содержит путь к каталогу, в котором находится cscript.exe или wscript.exe (в Windows XP обычно это C:\WINDOWS\SYSTEM32) |
| ScriptFullName | Содержит полный путь к запущенному сценарию |
| ScriptName | Содержит имя запущенного сценария |
| StdErr | Позволяет запущенному сценарию записывать сообщения в стандартный поток для ошибок |
| StdIn | Позволяет запущенному сценарию читать информацию из стандартного входного потока |
| StdOut | Позволяет запущенному сценарию записывать информацию в стандартный выходной поток |
| Version | Содержит версию WSH |

Опишем более подробно некоторые свойства объекта WScript.

**Свойство Arguments**

В следующем примере (листинг 5.1) с помощью цикла For Each … Next на экран выводятся все параметры командной строки, с которыми был запущен сценарий.

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

' Имя: Args.vbs

' Язык: VBScript

' Описание: Работа с аргументами запущенного сценария

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

Option Explicit

Dim s,objArgs,Arg

Set objArgs = WScript.Arguments ' Создаем объект WshArguments

For Each Arg In objArgs

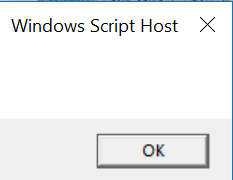
s=s & Arg ' Формируем строки со значениями аргументов

Next

WScript.Echo s ' Выводим сформированные строки

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

Листинг 5.1. Вывод всех параметров командной строки сценария (VBScript)



**Методы объекта WScript**

Объект WScript имеет несколько методов, которые описаны в табл. 5.3.

|  |  |
| --- | --- |
| Таблица 5.3. Методы объекта WScript | |
| **Метод** | **Описание** |
| CreateObject(strProgID [, strPrefix]) | Создает объект, заданный параметром strProgID |
| ConnectObject(strObject, strPrefix) | Устанавливает соединение с объектом strObject, позволяющее писать функции-обработчики его событий (имена этих функций должны начинаться с префикса strPrefix ) |
| DisconnectObject(obj) | Отсоединяет объект obj, связь с которым была предварительно установлена в сценарии |
| Echo([Arg1] [, Arg2] [,…]) | Выводит текстовую информацию на консоль или в диалоговое окно |
| GetObject(strPathname [, strProgID], [strPrefix]) | Активизирует объект автоматизации, определяемый заданным файлом (параметр strPathName ) или объект, заданный параметром strProgID |
| Quit([intErrorCode]) | Прерывает выполнение сценария с заданным параметром intErrorCode кодом выхода. Если параметр intErrorCode не задан, то объект WScript установит код выхода равным нулю |
| Sleep(intTime) | Приостанавливает выполнения сценария (переводит его в неактивное состояние) на заданное параметром intTime число миллисекунд |

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

'\* Имя: EchoExample.vbs

'\* Язык: VBScript

'\* Описание: Использование метода WScript.Echo

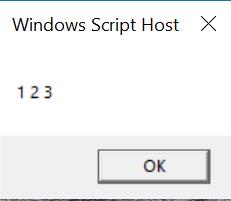
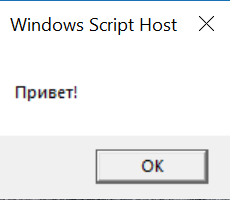
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WScript.Echo 'Выводим пустую строку

WScript.Echo 1,2,3 'Выводим числа

WScript.Echo "Привет!" 'Выводим строку

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

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

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

'\* Имя: SleepExample.vbs

'\* Язык: VBScript

'\* Описание: Использование метода WScript.Sleep

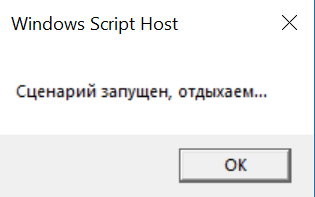
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WScript.Echo "Сценарий запущен, отдыхаем…"

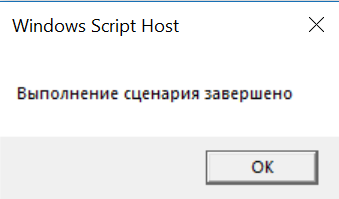
WScript.Sleep 5000

WScript.Echo "Выполнение сценария завершено"

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



Прошло 5 секунд…



|  |  |
| --- | --- |
| Таблица 5.4. Свойства объекта WshShell | |
| **Свойство** | **Описание** |
| CurrentDirectory | Здесь хранится полный путь к текущему каталогу (к каталогу, из которого был запущен сценарий) |
| Environment | Содержит объект WshEnvironment, который обеспечивает доступ к переменным среды операционной системы для Windows NT/2000/XP или к переменным среды текущего командного окна для Windows 9x |
| SpecialFolders | Содержит объект WshSpecialFolders для доступа к специальным папкам Windows (рабочий стол, меню Пуск (Start) и т. д.) |

Опишем теперь методы, имеющиеся у объекта WshShell

|  |  |
| --- | --- |
| Таблица 5.5. Методы объекта WshShell | |
| **Метод** | **Описание** |
| AppActivate(title) | Активизирует заданное параметром title окно приложения. Строка title задает название окна (например, "calc" или "notepad" ) или идентификатор процесса ( Process ID, PID ) |
| CreateShortcut(strPathname) | Создает объект WshShortcut для связи с ярлыком Windows (расширение lnk) или объект WshUrlShortcut для связи с сетевым ярлыком (расширение url). Параметр strPathnameзадает полный путь к создаваемому или изменяемому ярлыку |
| Environment(strType) | Возвращает объект WshEnvironment, содержащий переменные среды заданного вида |
| Exec(strCommand) | Создает новый дочерний процесс, который запускает консольное приложение, заданное параметром strCommand. В результате возвращается объект WshScriptExec, позволяющий контролировать ход выполнения запущенного приложения и обеспечивающий доступ к потокам StdIn, StdOut и StdErr этого приложения |
| ExpandEnvironmentStrings(strString) | Возвращает значение переменной среды текущего командного окна, заданной строкой strString (имя переменной должно быть окружено знаками "%" ) |
| LogEvent(intType, strMessage [,strTarget]) | Протоколирует события в журнале Windows NT/2000/XP или в файле WSH.log. Целочисленный параметр intType определяет тип сообщения, строка strMessage — текст сообщения. Параметр strTarget может задаваться только в Windows NT/2000/XP, он определяет название системы, в которой протоколируются события (по умолчанию это локальная система). Метод LogEvent возвращает true, если событие записано успешно и false в противном случае |
| Popup(strText,[nSecToWait], [strTitle], [nType]) | Выводит на экран информационное окно с сообщением, заданным параметром strText. Параметр nSecToWait задает количество секунд, по истечении которых окно будет автоматически закрыто, параметр strTitle определяет заголовок окна, параметр nTypeуказывает тип кнопок и значка для окна |
| RegDelete(strName) | Удаляет из системного реестра заданный параметр или раздел целиком |
| RegRead(strName) | Возвращает значение параметра реестра или значение по умолчанию для раздела реестра |
| RegWrite(strName, anyValue [,strType]) | Записывает в реестр значение заданного параметра или значение по умолчанию для раздела |
| Run(strCommand, [intWindowStyle], [bWaitOnReturn]) | Создает новый независимый процесс, который запускает приложение, заданное параметром strCommand |
| SendKeys(string) | Посылает одно или несколько нажатий клавиш в активное окно (эффект тот же, как если бы вы нажимали эти клавиши на клавиатуре) |
| SpecialFolders(strSpecFolder) | Возвращает строку, содержащую путь к специальной папке Windows, заданной параметром strSpecFolder |

Рассмотрим более подробно некоторые методы, приведенные в табл. 5.5.

**Метод CreateShortcut**

Приведем пример сценария, в котором создаются два ярлыка — на сам выполняемый сценарий (объект oShellLink ) и на сетевой ресурс ( oUrlLink ).

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

'\* Имя: MakeShortcuts.vbs

'\* Язык: VBScript

'\* Описание: Создание ярлыков из сценария

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

Dim WshShell,oShellLink,oUrlLink

' Создаем объект WshShell

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

' Создаем ярлык на файл

Set oShellLink=WshShell.CreateShortcut("Current Script.lnk")

' Устанавливаем путь к файлу

oShellLink.TargetPath=WScript.ScriptFullName

' Сохраняем ярлык

oShellLink.Save

' Создаем ярлык на сетевой ресурс

Set oUrlLink = WshShell.CreateShortcut("Microsoft Web Site.URL")

' Устанавливаем URL

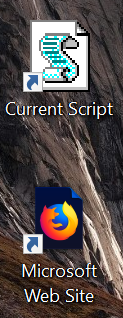
oUrlLink.TargetPath = "http://www.microsoft.com"

' Сохраняем ярлык

oUrlLink.Save

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

Создание ярлыков из сценария (VBScript)



**Метод Environment**

Параметр strType задает вид переменных среды, которые будут записаны в коллекции WshEnvironment ; возможными значениями этого параметра являются "System" (переменные среды операционной системы), "User" (переменные среды пользователя), "Volatile"(временные переменные) или "Process" (переменные среды текущего командного окна).

В следующем примере (листинг 5.6) мы распечатываем число процессоров, имеющихся в компьютере с операционной системой Windows XP (переменная NUMBER\_OF\_PROCESSORS ), и тип операционной системы.

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

'\* Имя: ShowEnvir.vbs

'\* Язык: VBScript

'\* Описание: Получение значений некоторых переменных среды

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

Dim WshShell,WshSysEnv

' Создаем объект WshShell

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

' Создание коллекции WshEnvironment

Set WshSysEnv = WshShell.Environment("SYSTEM")

WScript.Echo WshSysEnv("OS")

WScript.Echo WshShell.Environment.Item("NUMBER\_OF\_PROCESSORS")

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

Листинг 5.6. Получение значений переменных среды (VBScript)

**Метод Run**

Параметр intWindowStyle устанавливает вид окна для запускаемого приложения (табл. 5.6).

|  |  |  |
| --- | --- | --- |
| Таблица 5.6. Типы окна (intWindowStyle) | | |
| **Параметр** | **Константа Visual Basic** | **Описание** |
| 0 | vbHide | Прячет текущее окно и активизирует другое окно (показывает его и передает ему фокус) |
| 1 | vbNormalFocus | Активизирует и отображает окно. Если окно было минимизировано или максимизировано, система восстановит его первоначальное положение и размер. Этот флаг должен указываться сценарием во время первого отображения окна |
| 2 | vbMinimizedFocus | Активизирует окно и отображает его в минимизированном (свернутом) виде |
| 3 | vbMaximizedFocus | Активизирует окно и отображает его в максимизированном (развернутом) виде |
| 4 | vbNormalNoFocus | Отображает окно в том виде, в котором оно находилось последний раз. Активное окно при этом остается активным |
| 5 |  | Активизирует окно и отображает его в текущем состоянии |
| 6 | vbMinimizedNoFocus | Минимизирует заданное окно и активизирует следующее (в Z-порядке) окно |
| 7 |  | Отображает окно в свернутом виде. Активное окно при этом остается активным |
| 8 |  | Отображает окно в его текущем состоянии. Активное окно при этом остается активным |
| 9 |  | Активизирует и отображает окно. Если окно было минимизировано или максимизировано, система восстановит его первоначальное положение и размер. Этот флаг должен указываться, если производится восстановление свернутого окна (его нельзя использовать в методе Run ) |
| 10 |  | Устанавливает режим отображения, опирающийся на режим программы, которая запускает приложение |

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

'\* Имя: RetCode.vbs

'\* Язык: VBScript

'\* Описание: Вывод кода выхода запущенного приложения

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

' Создаем объект WshShell

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

' Запускаем Блокнот и ожидаем завершения его работы

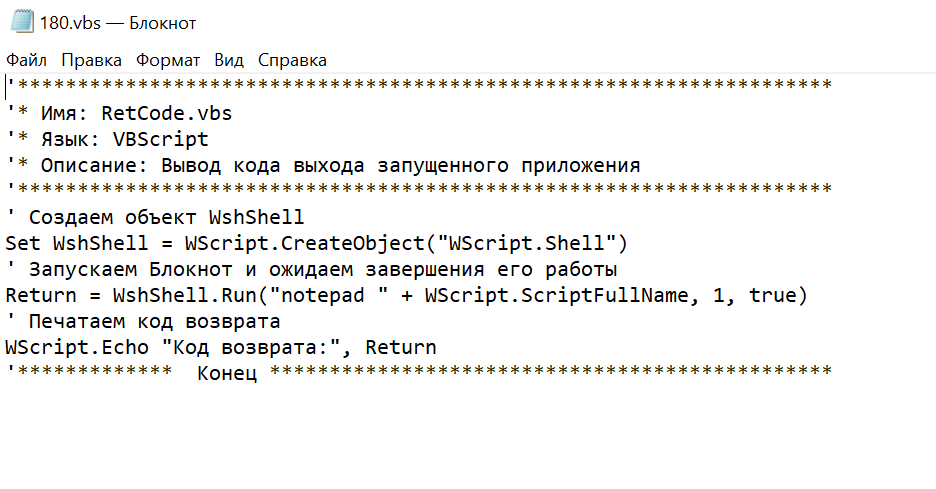
Return = WshShell.Run("notepad " + WScript.ScriptFullName, 1, true)

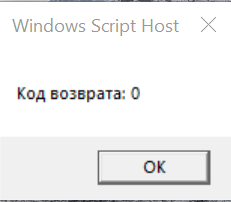
' Печатаем код возврата

WScript.Echo "Код возврата:", Return

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

Вывод кода выхода запущенного приложения





**Метод SendKeys**

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 5.7. Коды специальных клавишей для SendKeys | | | |
| **Названия клавиши** | **Код** | **Названия клавиши** | **Код** |
| <Backspace> | {BACKSPACE}, {BS} или {BKSP} | < \to > | {RIGHT} |
| <Break> | {BREAK} | <F1> | {F1} |
| <*Caps Lock*> | {CAPSLOCK} | <F2> | {F2} |
| <Del> или <Delete> | {DELETE} или {DEL} | <F3> | {F3} |
| <End> | {END} | <F4> | {F4} |
| <Enter> | {ENTER} или ~ | <F5> | {F5} |
| <Esc> | {ESC} | <F6> | {F6} |
| <Home> | {HOME} | <F7> | {F7} |
| <Ins> или <Insert> | {INSERT} или {INS} | <F8> | {F8} |
| <Num Lock> | {NUMLOCK} | <F9> | {F9} |
| <Page Down> | {PGDN} | <F10> | {F10} |
| <Page Up> | {PGUP} | <F11> | {F11} |
| <Print Screen> | {PRTSC} | <F12> | {F12} |
| <Scroll Lock> | {SCROLLLOCK} | <F13> | {F13} |
| <Tab> | {TAB} | <F14> | {F14} |
| < \uparrow > | {UP} | <F15> | {F15} |
| < \gets > | {LEFT} | <F16> | {F16} |
| < \downarrow > | {DOWN} |  |  |

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

'\* Имя: RunCalc.vbs

'\* Язык: VBScript

'\* Описание: Активизация приложения с помощью имени окна

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

Dim WshShell

' Создаем объект WshShell

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

' Запускаем Калькулятор

WshShell.Run "calc"

' Приостанавливаем сценарий на 1 секунду

WScript.Sleep 1000

' Активизируем Калькулятор

WshShell.AppActivate "Calculator"

' Приостановка сценария на 0,1 секунды

WScript.Sleep(100)

' Посылаем нажатия клавиш в Калькулятор

WshShell.SendKeys "1{+}"

WScript.Sleep 500

WshShell.SendKeys "2"

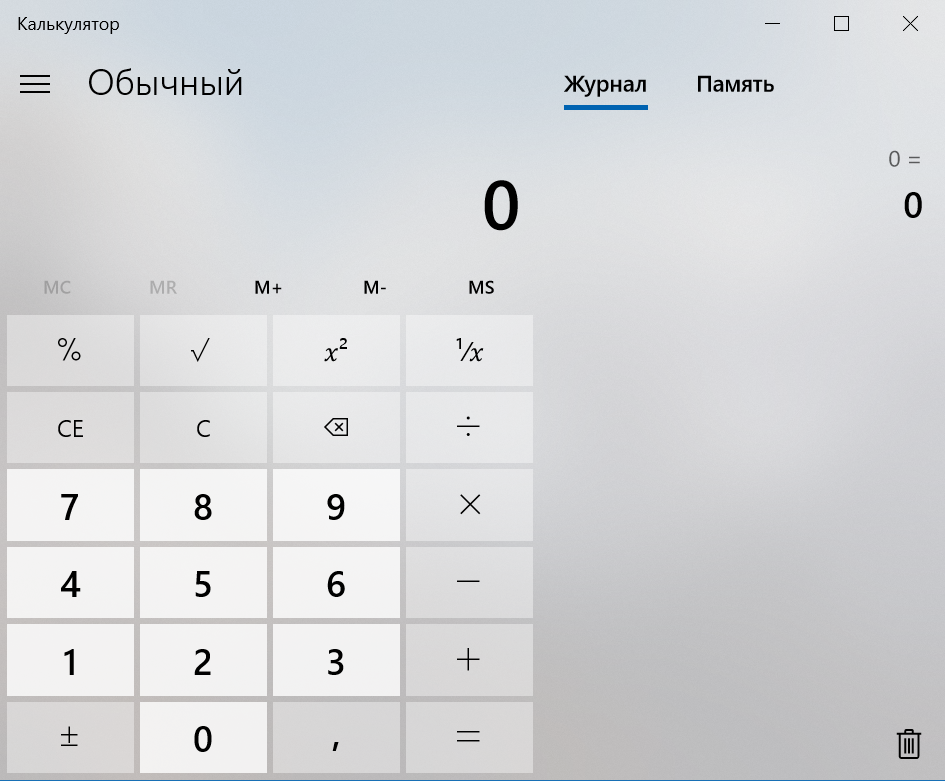
WScript.Sleep 500

WshShell.SendKeys "~"

WScript.Sleep 2500

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

Листинг 5.8. Стандартный калькулятор Windows



**Объекты-коллекции**

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

' Имя: Args.vbs

' Язык: VBScript

' Описание: Работа с аргументами запущенного сценария

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

Option Explicit

Dim i,Arg,objArgs,s,objNamedArgs,objUnnamedArgs ' Объявляем переменные

Set objArgs = WScript.Arguments ' Создаем объект WshArguments

' Определяем общее количество аргументов

s="Всего аргументов: " & objArgs.Count() & vbCrLf

For Each Arg In objArgs

s=s & Arg & vbCrLf ' Формируем строки со значениями аргументов

Next

Set objUnnamedArgs=objArgs.Unnamed ' Создаем объект WshUnnamed

' Определяем количество безымянных аргументов

s=s & vbCrLf & "Безымянных аргументов: " & objUnnamedArgs.length & vbCrLf

For Each Arg In objUnnamedArgs

' Формируем строки со значениями безымянных аргументов

s=s & Arg & vbCrLf

Next

Set objNamedArgs=objArgs.Named ' Создаем объект WshNamed

' Определяем количество именных аргументов

s=s & vbCrLf & "Именных аргументов: " & objNamedArgs.Length & vbCrLf

' Проверяем, существует ли аргумент /Имя:

If objNamedArgs.Exists("Имя") Then

s=s & objNamedArgs("Имя") & vbCrLf

End If

' Проверяем, существует ли аргумент /Comp:

If objNamedArgs.Exists("Comp") Then

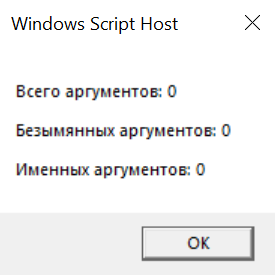
s=s & objNamedArgs("Comp") & vbCrLf

End If

WScript.Echo s ' Выводим сформированные строки

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

Листинг 5.9. Доступ к параметрам командной строки запущенного сценария (VBScript)



**Объект WshEnvironment**

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

' Имя: Environment.vbs

' Язык: VBScript

' Описание: Работа с переменными среды

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

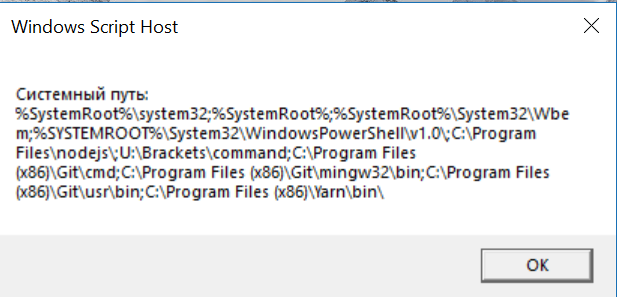
Dim WshShell, WshSysEnv

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

Set WshSysEnv=WshShell.Environment

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

Листинг 5.10. Работа с переменными среды (VBScript)



**Объект WshSpecialFolders**

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

' Имя: SpecFold1.vbs

' Язык: VBScript

' Описание: Вывод названий всех специальных папок Windows

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

Option Explicit

Dim WshShell, WshFldrs, SpecFldr, s ' Объявляем переменные

' Создаем объект WshShell

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

' Создаем объект WshSpecialFolders

Set WshFldrs = WshShell.SpecialFolders

s="Список всех специальных папок:" & vbCrLf & vbCrLf

' Перебираем все элементы коллекции WshFldrs

For Each SpecFldr In WshFldrs

' Формируем строки с путями к специальным папкам

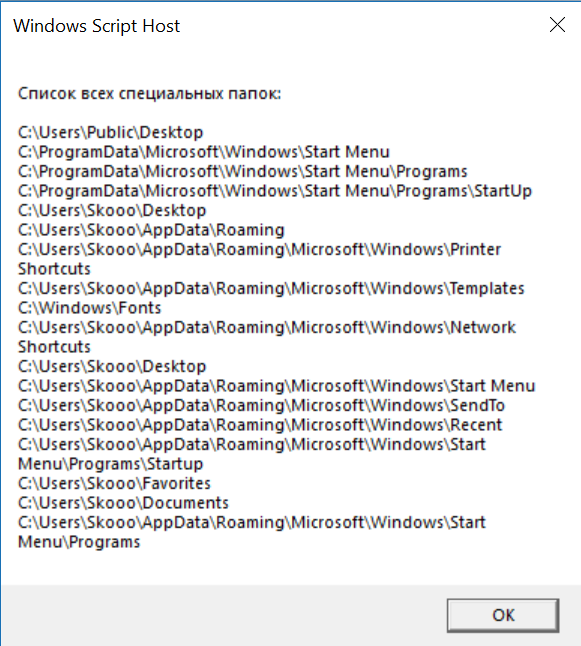
s=s & SpecFldr & vbCrLf

Next

WScript.Echo s

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

Листинг 5.11. Формирование списка всех специальных папок (VBScript)



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

' Имя: SpecFold2.vbs

' Язык: VBScript

' Описание: Вывод названий заданных специальных папок Windows

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

Option Explicit

Dim WshShell, WshFldrs, s ' Объявляем переменные

' Создаем объект WshShell

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

' Создаем объект WshSpecialFolders

Set WshFldrs = WshShell.SpecialFolders

' Формируем строки с путями к конкретным специальным папкам

s="Некоторые специальные папки:" & vbCrLf & vbCrLf

s=s+"Desktop:"+WshFldrs("Desktop") & vbCrLf

s=s+"Favorites:"+WshFldrs("Favorites") & vbCrLf

s=s+"Programs:"+WshFldrs("Programs")

WScript.Echo s ' Выводим сформированные строки на экран

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

Листинг 5.12. Доступ к определенным специальным папкам (VBScript)

