***Лабораторна робота №8***

***«Запуск програмних застосувань зі сценаріїв WSH. Імітація натиснення клавіш у програмних застосуваннях»***

Метод ***Run*** об'єкта ***WshShell*** дозволяє запускати інші програми. У табл. 4 наведено синтаксис виклику цього методу. Для другого параметра (***intWindowStyle***), що визначає зовнішній вигляд вікна програми яку ви запускаєте, допустимі значення, наведені в табл. 1.

Табл. 1. Можливі значення параметра ***intWindowStyle*** методу ***Run***

|  |  |
| --- | --- |
| **Значення** | **Опис** |
| ***0*** | Запуск у прихованому вигляді |
| ***1*** | Звичайний розмір вікна, якщо вікно обернено або розгорнуто на весь екран, то йому повертається вихідний вигляд і положення на екрані |
| ***2*** | Запуск у згорнутому вигляді |
| ***3*** | Вікно розгортається на весь екран |
| ***4*** | Запуск в звичайному розмірі, в неактивному стані (без фокусу) |
| ***8*** | Звичайний розмір в неактивному стані, але у фокусі залишається запущене застосування |
| ***10*** | Вид вікна визначається додатком, який його викликав |

Запустимо за допомогою методу ***Run*** в браузері **Internet Explorer** заданий сайт:

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

***wh.Run ("\"C:\\Program Files\\Internet Explorer\\iexplore.exe\"www.ukr.net");***

Наведений приклад демонструє додатково властивість методу ***Run***: якщо в шляху до запускається з додатком присутні проміжки, шлях необхідно укладати в подвійні лапки. При використанні **JScript** необхідно пам’ятати про екранування лапками спецсимволів (\ "і \\ в останньому прикладі).

За допомогою методу ***Run*** можна виконати консольну команду:

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

***var status=wh.Run(”ping 192.168.0.1”, 0, true);***

***if( status!=0) WScript.Echo(“Не можу зв'язатися з віддаленим комп'ютером ”);***

В останньому прикладі метод ***Run*** виконує команду в прихованому вікні командного інтерпретатора, припиняючи роботу скрипта і повертаючи код завершення виконаної команди. Ще однією можливістю запустити консольний додаток з скрипта є використання методу ***WShell.Exec***. На відміну від метода ***Run***, він створює новий процес як дочірній, даючи батьківському скрипту можливості доступу до своїх об'єктів ***StdIn***, ***StdOut*** і ***StdErr*** і тим самим спрощуючи процес обміну інформацією. У наступному прикладі команда ***Exec*** не тільки запускає дочірній процес, але і створює об'єкт ***wse***, за допомогою властивості ***StdOut*** якого скрипт, який запускає, може отримати виведення скрипта, який запускається.

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

***var wse=wh.Exec("netstat");***

***str=wse.StdOut.ReadAll();***

***StrMas=str.split("\n");***

***for (i=4;i<StrMas.length;i++)***

***{***

***if (StrMas[i].indexOf(":http")>0)***

***WScript.Echo(StrMas[i].substring(StrMas[i].indexOf(":")+5,***

***StrMas[i].lastIndexOf(":http")));***

***}***

Наведений вище скрипт запускає команду ***netstat*** в дочірньому процесі і вибирає в його виведення інформацію тільки про сполуки за протоколом **http**.

Для взаємодії з вже запущеним додатком крім його власних інтерфейсних методів і властивостей можна використовувати стандартний метод класу ***WshShell*** ***SendKeys***. Цей метод дозволяє імітувати клавіатурне введення для програми, що має фокус введення (який, в свою чергу, можна привласнити додатком методом ***AppActivate***). Наступний приклад відкриває стандартний Блокнот і заносить в нього рядок ***Hello, world!*** і зберігає цей текст на диску під ім'ям ***WSH.txt***.

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

***var wse=wh.Exec("%windir%\\notepad.exe");***

***WScript.Sleep(1000);***

***wh.AppActivate(wse.ProcessID);***

***var str=new String("Hello world.");***

***wh.SendKeys(str);***

***WScript.Sleep(1500);***

***wh.SendKeys("{BACKSPACE}");***

***wh.SendKeys("!");***

***WScript.Sleep(1500);***

***var Answer=wh.Popup("Зберегти документ? \n(Інакше закрити блокнот)",10,"WSH",36);***

***wh.AppActivate(wse.ProcessID);***

***WScript.Sleep(1500);***

***if (Answer==6)***

***{***

***wh.SendKeys("^s");***

***WScript.Sleep(100);***

***wh.SendKeys("WSH.txt");***

***wh.SendKeys("{ENTER}");***

***}***

***else***

***wse.Terminate();***

***for (;;)***

***{***

***if (wse.Status==1) break;***

***WScript.Sleep(100);***

***}***

***WScript.Echo("Блокнот завершений с кодом "+wse.ExitCode);***

В останньому прикладі звертає на себе увагу спосіб відправки повідомлень про натискання клавіші ***S*** в комбінації c ***Ctrl*** (***^S***). Аналогічно комбінацію з ***Alt*** відправляться з символом ***%*** (наприклад, ***%A***), а комбінацію з ***Shift*** - c символом ***+*** (наприклад, ***+D***). Для спецклавіш, таких як функціональні, ***Enter***, ***Delete*** і іншихх використовуються рядкові позначення, які необхідно укладати в фігурні дужки. Наприклад, закрити активне вікно можна командою ***wh.SendKeys("%{F4}")***. Більш докладно про функції ***SendKeys*** (методах емуляції комбінацій клавіш, автоповтору натискань, строкових позначеннях спецклавіш і ін.) можна подивитися в документації.

Ще одна деталь останнього скрипта, що не зустрічалася раніше - використання стандартного діалогового вікна викликом методу ***WScript.Popup***. Цей метод дозволяє визначати текст повідомлення і заголовка вікна, набір кнопок у вікні. Як результат повертається код натиснутої кнопки.

***Завдання***

1. Написати скрипт, який запускає редактор реєстру, або будь-який інший додаток?
2. Написати скрипт, який запускає стандартний калькулятор і обчислює у ньому значення виразу (12345-678)/910.

***Контрольні питання***

1. Як з коду скрипта виконати сторонній скрипт або запустити застосування? Які засоби дозволяють управляти стороннім застосуванням?
2. Як у скрипті WSH імітувати натиснення певних клавіш у запущених програмних застосуваннях?
3. Яким чином при імітації натиснення клавіш задаються звичайні та спеціальні клавіші? Наведіть приклади.