**Отчёт по Лабораторной работе №3**

**Выполнил: Ишгулов Ратмир Русланович, группа 4311**

Цель работы:

1. Изучение объектов WSH.
2. Изучение приемов программирования с использованием объектов WSH.
3. Получение навыков разработки сценариев.

Примеры работы с синтаксисом:

Задание 1:

Листинг:

//wshShk.js - создание ярлыка Блокнота на Рабочем столе

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

//путь к Рабочему столу

var DesktopPath = WSHShell.SpecialFolders("Desktop");

//cоздаем ярлык на Рабочем столе

var MyShortcut=WSHShell.CreateShortcut(DesktopPath+"\\Ярлык на блокнот.lnk");

//Задаем свойства объекта-ярлыка и сохраняем их

MyShortcut.TargetPath=WSHShell.ExpandEnvironmentStrings("%windir%\\notepad.exe");

MyShortcut.WorkingDirectory=WSHShell.ExpandEnvironmentStrings("%windir%");

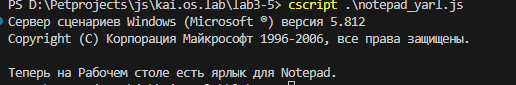
MyShortcut.WindowStyle=4;

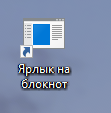
MyShortcut.IconLocation=WSHShell.ExpandEnvironmentStrings("%windir%\\notepad.exe,0");

MyShortcut.Save();

WScript.Echo("Теперь на Рабочем столе есть ярлык для Notepad.");

Результат:





Задание 2:

Листинг:

// Исполнение команды MS-DOS с помощью метода Run

var command, dos\_command, option

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

// Команда вызова командного процессора

command = "%COMSPEC% /K ";

// Команда получения справки о командах MS-DOS

dos\_command = "help ";

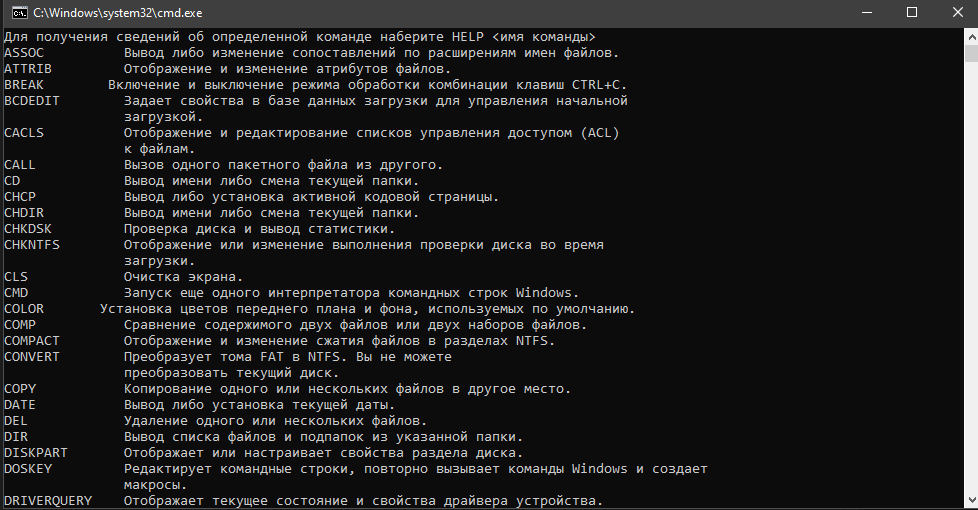
// Использование постраничного вывода

option = "| more"

// Исполнить команду.

Shell.Run(command + dos\_command + option);

Результат:



Постановка задачи

Вариант 6:

1. Разработать процедуру вывода справки о командах MS DOS с возможностью перенаправления справочной информации в файл. Организовать ввод имени интересуемой команды, пути и имени результирующего файла в процедуру с помощью параметров командной строки при вызове процедуры.
2. Разработать процедуру запуска Блокнота для создания новой или редактирования существующей процедуры с возможностью ее последующего запуска в оконном режиме или режиме командной строки. Организовать ввод пути и имени редактируемой процедуры и режим ее запуска с помощью параметров командной строки.
3. Разработать процедуру, выполняющую заданную в индивидуальном задании последовательность операций над файлами (каталогами) с помощью команд MS. Организовать ввод всех исходных данных в процедуру с помощью параметров командной строки при вызове процедуры. Предусмотреть возможность перенаправления результатов выполнения процедуры в файл.

6. Перемещение каталогов: 6.1. Создать исходный каталог; 6.2. С помощью Блокнота создать исходный файл; 6.3. Переместить исходный каталог в результирующий каталог; 6.4. Переименовать файл в результирующем каталоге; 6.5. Запустить Блокнот для редактирования файла в результирующем каталоге; 6.6. Вывести оглавление результирующего каталога;

Задание 1 Листинг:

var WSH = WScript;

var shell = WSH.CreateObject("WScript.Shell");

var fso = new ActiveXObject("Scripting.FileSystemObject");

function showUsage() {

        WSH.Echo("Usage:");

        WSH.Echo("  cscript //nologo " + WSH.ScriptName + " <commandName> [outputFilePath]");

        WSH.Echo("");

        WSH.Echo("Examples:");

        WSH.Echo("  cscript //nologo " + WSH.ScriptName + " dir");

        WSH.Echo("  cscript //nologo " + WSH.ScriptName + " copy \"C:\\temp\\help\_copy.txt\"");

}

function ensureParentFolderExists(path) {

        var p = path.replace(/\//g, "\\");

        var idx = p.lastIndexOf("\\");

        if (idx <= 0) return;

        var parent = p.substring(0, idx);

        if (parent.length <= 2) return;

        if (fso.FolderExists(parent)) return;

        ensureParentFolderExists(parent);

        fso.CreateFolder(parent);

}

var args = WSH.Arguments;

if (args.Length < 1) {

        showUsage();

        WSH.Quit(1);

}

var cmdName = args.Item(0);

var outPath = null;

if (args.Length >= 2) {

        outPath = args.Item(1);

        outPath = shell.ExpandEnvironmentStrings(outPath);

        try {

                ensureParentFolderExists(outPath);

        } catch (e) {

                WSH.Echo("Ошибка при создании папки для файла: " + e.message);

                WSH.Quit(2);

        }

}

var dosCommand = "help " + cmdName;

var comspec = shell.ExpandEnvironmentStrings("%COMSPEC%");

var fullCmd;

if (outPath) {

        fullCmd = comspec + " /C " + dosCommand + " > \"" + outPath + "\"";

        WSH.Echo("Выполняю: " + dosCommand + "  ->  " + outPath);

        try {

                shell.Run(fullCmd, 1, true);

                WSH.Echo("Готово. Файл: " + outPath);

        } catch (e) {

                WSH.Echo("Ошибка при выполнении команды: " + e.message);

                WSH.Quit(3);

        }

} else {

        fullCmd = comspec + " /K " + dosCommand + " | more";

        WSH.Echo("Выполняю: " + dosCommand + " (интерактивный вывод)");

        try {

                shell.Run(fullCmd, 1, true);

        } catch (e) {

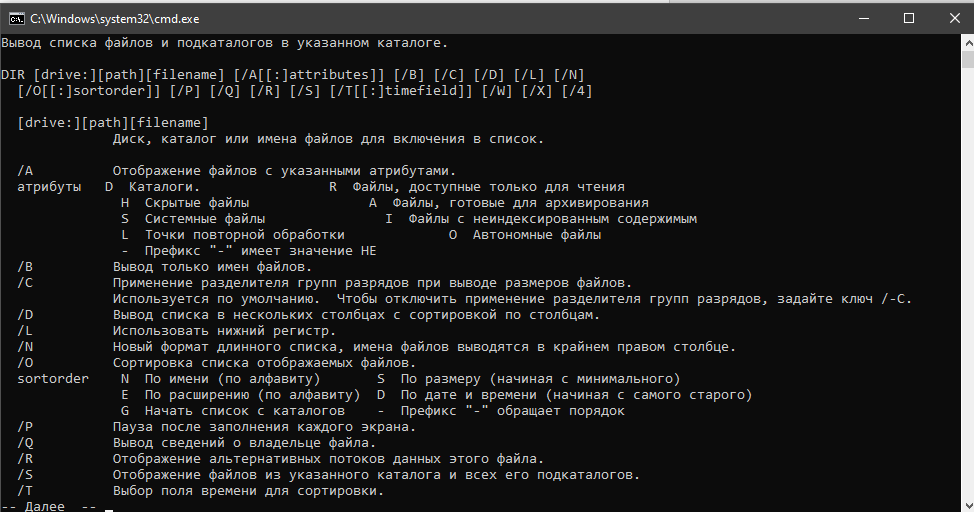
                WSH.Echo("Ошибка при выполнении команды: " + e.message);

                WSH.Quit(4);

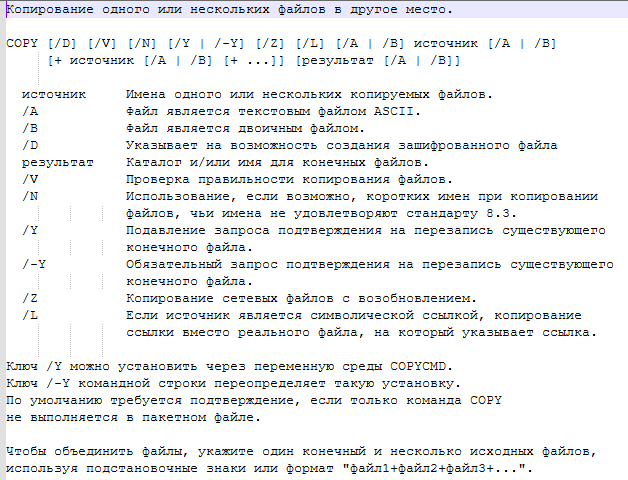
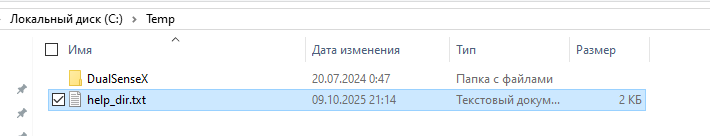
        }

}

Результат:



Задание 2 Листинг:

var WSH = WScript;

var shell = WSH.CreateObject("WScript.Shell");

var fso = WSH.CreateObject("Scripting.FileSystemObject");

if (WSH.Arguments.Length < 2) {

    WSH.Echo("Usage: cscript //nologo script.js \"<file\_path>\" <win|console>");

    WSH.Quit(1);

}

var filePath = WSH.Arguments.Item(0);

var mode = WSH.Arguments.Item(1).toLowerCase();

if (!fso.FileExists(filePath)) {

    var tf = fso.CreateTextFile(filePath, true);

    tf.WriteLine("// New script file");

    tf.Close();

}

shell.Run('"' + shell.ExpandEnvironmentStrings("%windir%\\notepad.exe") + '" "' + filePath + '"', 1, true);

if (mode === "win") {

    shell.Run('"' + shell.ExpandEnvironmentStrings("%windir%\\system32\\wscript.exe") + '" "' + filePath + '"', 1, true);

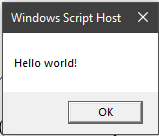
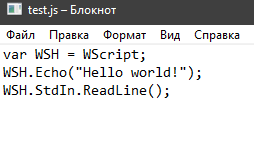
} else if (mode === "console") {

    shell.Run('cscript //nologo "' + WSH.Arguments.Item(0) + '"', 1, true);

}

Результат:





Задание 3 Листинг:

var WSH = WScript;

var fso = WSH.CreateObject("Scripting.FileSystemObject");

var shell = WSH.CreateObject("WScript.Shell");

if (WSH.Arguments.Length < 3) {

    WSH.Echo("Usage: cscript //nologo script.js <source\_dir> <dest\_dir> <file\_name>");

    WSH.Quit(1);

}

var sourceDir = WSH.Arguments(0);

var destDir = WSH.Arguments(1);

var fileName = WSH.Arguments(2);

// 6.1. Создать исходный каталог;

if (!fso.FolderExists(sourceDir)) {

    fso.CreateFolder(sourceDir);

}

var filePath = sourceDir + "\\" + fileName;

// Проверка и удаление файла, если он существует

if (fso.FileExists(filePath)) {

    fso.DeleteFile(filePath);

}

// 6.2. С помощью Блокнота создать исходный файл;

var file = fso.CreateTextFile(filePath, true);

file.WriteLine("// Created by script");

file.Close();

shell.Run("notepad.exe \"" + filePath + "\"", 1, true);

// 6.3. Переместить исходный каталог в результирующий каталог;

if (fso.FolderExists(destDir)) {

    fso.DeleteFolder(destDir);

}

fso.MoveFolder(sourceDir, destDir);

// 6.4. Переименовать файл в результирующем каталоге;

var oldFilePath = destDir + "\\" + fileName;

var newFilePath = destDir + "\\renamed\_" + fileName;

fso.MoveFile(oldFilePath, newFilePath);

// 6.5. Запустить Блокнот для редактирования файла в результирующем каталоге;

shell.Run("notepad.exe \"" + newFilePath + "\"", 1, true);

// 6.6. Вывести оглавление результирующего каталога;

var result = shell.Exec("cmd /c dir \"" + destDir + "\"");

WSH.Echo(result.StdOut.ReadAll());

Результат:

