

Отчёт по Лабораторной работе №1

Выполнил: Ишгулов Ратмир Русланович, группа 4311

Цель работы:

- 1) Изучение основ Windows Scripting Host.
- 2) Изучение языка JScript.
- 3) Получение первичных навыков разработки простейших сценариев.

Содержание работы:

- 1) Изучение структуры и синтаксиса языка JScript.
- 2) Изучение основных конструкций JScript.
- 3) Изучение и выполнение всех рассмотренных в руководстве примеров.
- 4) Разработка процедуры решения поставленной задачи.
- 5) Отладка и выполнение процедуры.
- 6) Составление отчета.

Краткие теоретические сведения

Windows Scripting Host (WSH) – языконезависимая платформа для Win32. Microsoft предоставляет его как вместе со своими продуктами (Windows 4 98, IIS 4.0, Windows 2000 и выше), так и как отдельный инсталлятор (для Windows 95 и Windows NT 4). В базовую поставку входит поддержка скриптовых языков Java Script и Visual Basic Script, но возможна установка расширений для поддержки других скриптовых языков (разработаны расширения для Perl, TCL, REXX, Python и многих других). Раньше единственным поддерживаемым Windows скриптовым языком был командный язык DOS. Но его возможности довольно бедны по сравнению с VBScript и JScript. Хотя командный язык DOS по-прежнему поддерживается, современные ActiveX-скрипты позволяют решать более сложные задачи. Скрипт может принимать решения на основе использования полноценных операторов If/Else. Сценарий может выполнять один набор команд, если данное условие истинно, или другой набор, если условие ложно. Кроме того, JScript и VBScript хороши при выполнении математических операций, включая общие тригонометрические функции. Другое свойство WSH заключается в том, что скрипты могут исполняться вне браузера. Достаточно кликнуть по файлу с текстом скрипта или ввести его название в командной строке, чтобы запустить его на исполнение. WSH не требователен к памяти и прекрасно подходит для автоматизации Windows. Как ни хороши скриптовые языки, многого с ними не сотворишь. Они неспособны повлиять на что-либо за пределами родного скрипта. Эти языки не имеют прямых инструкций, позволяющих читать или записывать файлы на диске, выводить информацию в командную строку, изменять записи в Реестре Windows и так далее. Чтобы справиться с такими задачами, можно воспользоваться дополнительными COM-объектами. Ряд таких объектов входит в поставку WSH, а один из них, WScript, даже уже имеет созданный экземпляр, и им можно пользоваться непосредственно, без предварительного создания. Остальные создаются с помощью синтаксиса, соответствующего конкретному языку или функции WScript.CreateObject.

Примеры работы с синтаксисом:

```
// conddemo.js – Код 1
var weekday=new Date().getDay();// получение дня недели

var str;

switch(weekday)
{
case 0:
    str="седьмой";
    break;
case 1:
```

```

str="первый";

break;

case 2:

str="второй";

break;

case 3:

str="третий";

break;

case 4:

str="четвёртый";

break;

case 5:

str="пятый";

break;

case 6:

str="шестой";

break;

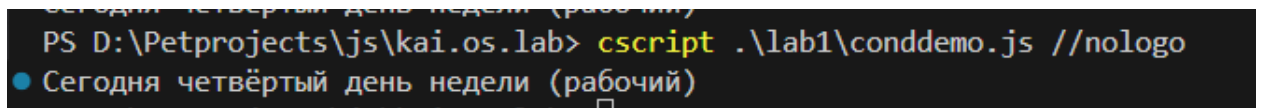
default:

str="неизвестный";

}

WScript.Echo("Сегодня "+str+" день недели ("+(weekday==0||weekday==6?"выходной":"рабочий")+");");

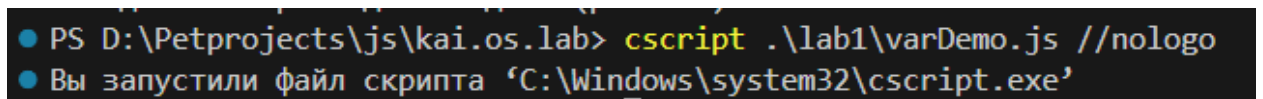
```



// varDemo.js – демонстрация использования переменных Код 2

```
var str = "Вы запустили файл скрипта " + WScript.FullName + " " // сложение строк
```

```
WScript.Echo(str)
```



Постановка задачи

Разработать процедуру расчета суммы n первых членов ряда, в соответствии с индивидуальным заданием. Вычисление факториалов произвести в циклах.

Вариант 6: $A_n = ((-1)^n * (2 * n^2 + 1) / (2 * n!)) * x^{2 * n}$

Листинг

```

function calculateFactorial(num) {
    if (num <= 1) return 1;

    var result = 1;

    for (var i = 2; i <= num; i++) {
        result *= i;
    }
}

```

```

        return result;
    }

function calculateSeriesSum(n, x) {
    var totalSum = 0;

    for (var currentN = 1; currentN <= n; currentN++) {
        var numerator = Math.pow(-1, currentN) * (2 * currentN * currentN + 1);
        var denominator = calculateFactorial(2 * currentN);
        var power = Math.pow(x, 2 * currentN);

        var currentTerm = (numerator / denominator) * power;
        totalSum += currentTerm;

        WScript.Echo(
            'A' + currentN + ' = (' + numerator + ' / ' + denominator + ') * ' + x +
            '^' +
            (2 * currentN) +
            ' = ' +
            currentTerm
        );
    }

    return totalSum;
}

var n = 3;
var x = 2;

WScript.Echo(
    'Расчет суммы первых ' + n + ' членов ряда');
var finalResult = calculateSeriesSum(n, x);
WScript.Echo('Сумма первых ' + n + ' членов ряда: ' + finalResult);

```

Результат работы программы:

```

PS D:\Petprojects\js\kai.os.lab> cscript .\lab1\lab1_task6.js //nologo
Расчет суммы первых 3 членов ряда
A1 = (-3 / 2) * 2^2 = -6
A2 = (9 / 24) * 2^4 = 6
A3 = (-19 / 720) * 2^6 = -1.6888888888888888
Сумма первых 3 членов ряда: -1.6888888888888888

```