

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное образовательное бюджетное
учреждение высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»

А.Н. Губин, Ф. В. Филиппов

**ОСНОВЫ
ПРОГРАММИРОВАНИЯ
СЦЕНАРИЕВ В СРЕДЕ WSH**

УЧЕБНОЕ ПОСОБИЕ

СПбГУТ)))

**САНКТ-ПЕТЕРБУРГ
2013**

УДК 004.43

Рецензент

кандидат технических наук, доцент кафедры робототехники и автоматизации
производственных систем Санкт-Петербургского государственного
электротехнического университета «ЛЭТИ» *А. В. Шевченко*

*Утверждено редакционно-издательским советом СПбГУТ
в качестве учебного пособия*

Губин, А.Н.

Основы программирования сценариев в среде WSH: учебное пособие /
А.Н. Губин, Ф. В. Филиппов - СПб.: Изд-во СПбГУТ, 2013. - 180 с.

Рассматриваются основные принципы программирования сценариев на языке
VBScript в среде Windows Script Host и их использования для управления
приложениями и администрирования компьютерных систем и сетей.

Пособие предназначено для бакалавров и магистров специальности 230400 –
Информационные системы и технологии, а также аспирантов и специалистов в
области информационных технологий.

УДК 004.43

© Губин А.Н., Филиппов Ф.В., 2013

© Федеральное государственное образовательное
бюджетное учреждение высшего профессионального
образования «Санкт-Петербургский государственный
университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича», 2013

ПРЕДИСЛОВИЕ

Основными элементами современных информационных систем (ИС) как правило, являются вычислительные структуры. Реализация алгоритмов функционирования этих структур в большинстве своем подразумевает их программную реализацию. Достаточно часто в качестве вычислительных элементов ИС используются компьютеры под управлением операционных систем (ОС) семейства Windows. Такие компьютеры используются для реализации функций автоматизированных рабочих мест в различных ИС. В процессе разработки и проектирования подобных систем возникает необходимость решения задач не только обработки информации на прикладном уровне, но и на канальном и сетевом уровнях, а так же управления режимами работы самих компьютеров.

До операционной системы Windows-98 в ОС Windows практически не было средств автоматизации выполнения рутинных операций обработки служебной информации и управления режимами работы компьютера. Определенные задачи, такие как копирование файлов и изменение некоторых параметров ОС можно было выполнить с помощью командных файлов MS-DOS (BAT-файлов). Однако в BAT-файлы можно помещать лишь простые последовательности команд MS-DOS с достаточно серьезными ограничениями по выполняемым функциям (нет поддержки диалоговых окон, ограничения поддержки циклов и др.).

Результатом развития средств автоматизации управления ОС и обработки информации явилось появление в списке технологий Microsoft достаточно мощного средства – технологии Microsoft Windows Script, которая реализована автономным сервером сценариев *Windows Script Host* (WSH).

WSH позволяет исполнять сценарии на уровне ОС. Можно вызвать исполнение сценария из командной строки или из Windows Explorer, дважды щелкнув по выделенному имени файла сценария.

С самого начала для обеспечения обмена данными между приложениями в ОС Windows использовалась технология связывания и внедрения объектов (OLE – *Object Linking and Embedding*). Вначале технология OLE использовалась только для создания составных документов, а затем и для решения задач предоставления приложениям функций других приложений. Технология, позволяющая одному приложению вызывать функции другого приложения, получила название OLE Automation. В основе OLE и OLE Automation лежит разработанная Microsoft базовая компонентная технология COM (Component Object Model).

Компонентное программное обеспечение – это способ разработки программ, при котором используются технологии подобные технологиям, применяемым для разработки аппаратных средств. Сложные элементные

схемы собираются из стандартизированных микросхем, которые имеют четко определенные документированные функции. Разработчик может эффективно пользоваться такими микросхемами, не задумываясь об их внутренней структуре. В программных компонентах, написанных на каком-то языке программирования, детали реализации используемых алгоритмов также скрыты внутри компонента, разработчику доступны только интерфейсы, которыми могут пользоваться и другие приложения.

В настоящее время, вместо термина OLE используется термин ActiveX, обозначающий компоненты (объекты) созданные на базе COM технологии.

Технология ActiveX до последнего времени являлась ключевой в продуктах Microsoft (Microsoft Office, Internet Explorer и др.). В эти продукты для управления объектами были встроены интерпретаторы специальных языков сценариев: VBScript и JScript, однако вне этих продуктов, выполнять сценарии, написанные на VBScript и JScript, было нельзя.

Сервер сценариев WSH устраняет этот недостаток, предоставляя единый интерфейс для языков сценариев, которые позволяют использовать любые внешние объекты ActiveX. С помощью WSH сценарии могут быть исполнены непосредственно в ОС Windows, без встраивания их в HTML – страницы или другие объекты Microsoft Office.

В учебном пособии рассмотрены основные вопросы создания и использования сценариев на VBScript, а в приложении приведен краткий справочник по языку программирования сценариев VBScript.

Ограниченный объем пособия не позволяет достаточно подробно осветить все вопросы касающиеся всех важнейших проблем программирования сценариев на VBScript и JScript в среде WSH, поэтому для получения дополнительных сведений рекомендуется обратиться к соответствующим источникам, список которых включает 3 наименования. При работе над пособием авторы неоднократно осуществляли заимствование из этих источников определений, примеров и методов изложения, без отдельных ссылок в тексте.

В заключение авторы благодарят рецензентов и редакторов за внимательное прочтение рукописи и замечания, способствовавшие улучшению качества предлагаемого пособия

1. СОЗДАНИЕ И ЗАПУСК ПРОСТЕЙШИХ СЦЕНАРИЕВ

1.1. Серверы сценариев

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 VBScript-сценарий, выводящий на экран диалоговое окно с надписью "Привет Всем!". Для этого достаточно с помощью, например, стандартного Блокнота Windows (notepad.exe) создать файл Privet.vbs, содержащий всего одну строку:

WScript.Echo «Привет Всем!».

Для обеспечения работы этого сценария достаточно всего одной строки, однако полезно добавить в начало файла информацию о находящемся в нем сценарии: имя файла, используемый язык, краткое описание выполняемых действий. Для нашего сценария можно использовать следующий вариант оформления содержимого файла (рис. 1).

```
'=====
'Язык – VBScript
'Имя файла – Privet.vbs
'Автор – Иванов И.И.
'Дата – 12.02.2013
'Пример 1 – Вывод на экран
'=====
WScript.Echo "Привет всем!"
```

Рис. 1. Пример оформления файла-сценария

Запустить сценарий в работу можно несколькими способами.

Исполняемые файлы сервера сценариев расположены в директории %WINDIR%system32 и содержат два основных файла. Файл WScript.exe – оконно-диалоговый сервер сценариев и файл CScript.exe – консольный сервер сценариев. Разница в работе этих серверов состоит только в способе и форме вывода информации при выполнении скрипта – WScript генерирует вывод, основанный на диалоговых окнах, в то время как CScript возвращает вывод в командное окно, в котором был запущен скрипт.

Запустим сценарий, записанный в файле Privet.vbs из командной строки. Для этого необходимо запустить режим командной строки и выполнить в нем команду:

Cscript F:\PRIMER_WSH\Privet.vbs ,

где F:\PRIMER_WSH\Privet.vbs – полное (с указанием пути к файлу) имя файла-сценария.

Результат работы сценария в консольном режиме представлен на рис.2.

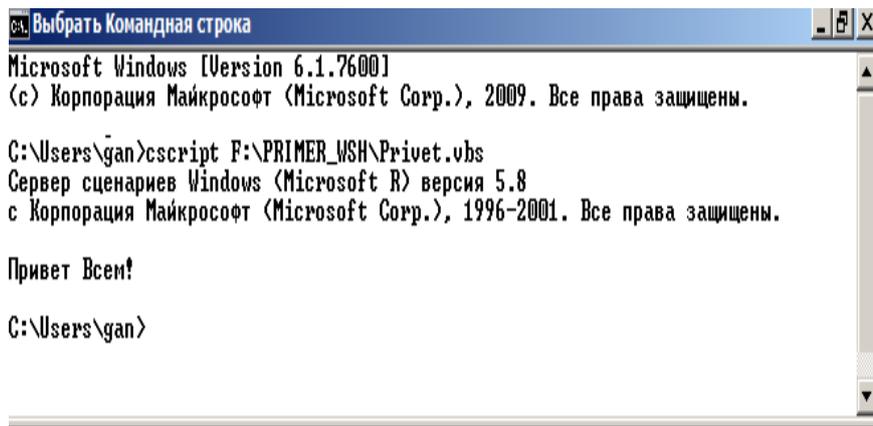


Рис. 2. Выполнение сценария Privet.vbs в консольном режиме

Этот сценарий можно выполнить из командной строки с помощью графической или оконно-диалоговой версии сервера сценариев. Для этого необходимо в режиме командной строки выполнить команду:

WCscript F:\PRIMER_WSH\Privet.vbs.

При выполнении этой команды на экране появится диалоговое окно (рис. 3).

Запустить сценарий на исполнение можно с помощью пункта Выполнить меню Пуск. Для этого необходимо набрать полное имя файла-сценария в поле Открыть (для Windows XP) или в поле Найти программы и файлы (для Windows 7). В этом случае по умолчанию сценарий будет выполнен с помощью сервера Wscript.exe. Результат работы файла будет представлен в виде графического диалогового окна.

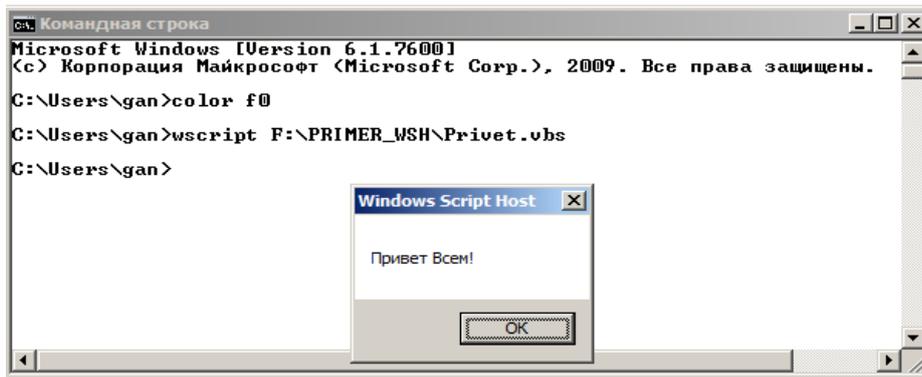


Рис. 3. Выполнение сценария Privet.vbs из командной строки с использованием графической версии сервера сценариев

Проще всего запустить сценарий, используя окно Проводника Windows или Рабочего стола. В этом случае достаточно выполнить двойной щелчок мышью на имени файла или на его значке.

Вопросы для самопроверки по разделу 1.1.

1. Как запустить в работу консольный сервер сценариев WSH?
2. С помощью запуска какого файла реализуется оконно-диалоговый режим работы сервера сценариев WSH?
3. Назовите простейшее средство для создания сценариев WSH.
4. Назовите основные способы запуска сценариев WSH.

1.2. Передача параметров серверу сценариев

В общем виде запуск сценариев через командную строку реализуется в следующей форме:

WScript имя сценария.расширение [параметры сервера] [аргументы сценария]
или
CScript имя сценария.расширение [параметры сервера] [аргументы сценария]

Перед параметрами сервера, которые включают или выключают различные опции Windows Script Host, ставятся два символа прямого слэша (//).

Аргументы скрипту можно передавать, указывая их после параметров (если они используются), отделяя от параметров пробелом.

Полный список параметров сервера позволяет вывести команда CScript //? (рис.4).

```

Командная строка
Microsoft Windows [Version 6.1.7600]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Users\gan>color f0

C:\Users\gan>Cscript //?
Сервер сценариев Windows (Microsoft R) версия 5.8
с Корпорация Майкрософт (Microsoft Corp.), 1996-2001. Все права защищены.
Использование: CScript имя_сценария.расширение [параметры...] [аргументы...]

Параметры:
//B          Пакетный режим: подавляются отображение ошибок и запросов сценария
//D          Включение режима Active Debugging
//E:ядро     Использование указанного ядра для выполнения сценария
//H:CScript  Стандартный сервер сценариев заменяется на CScript.exe
//H:WScript  Стандартный сервер сценариев заменяется на WScript.exe (по умолчанию)
//I          Диалоговый режим (по умолчанию, в противоположность //B)
//Job:xxxx   Выполнение указанного задания WSH
//Logo       Отображать сведения о программе (по умолчанию)
//NoLogo     Не отображать сведения о программе во время выполнения
//S          Сохранить для данного пользователя текущие параметры командной строки
//T:пп       Интервал ожидания (в секундах): максимальное время выполнения сценария
//X          Выполнение сценария в отладчике
//U          Использование кодировки Юникод при перенаправлении ввода-вывода с консоли

C:\Users\gan>

```

Рис. 4. Результат выполнения команды CScript //?

Список параметров для WScript.exe можно получить, если в меню «Пуск» выбрать команду «Выполнить», в окне которой набрать WScript //?.

Параметры исполнения сценария, запускаемого из командной строки, задаются параметрами сервера, указываемые в командной строке.

Так исполнение сценария Prvet.vbs с параметром //nologo приводит к результату (рис. 5), который отличается от результата исполнения этого сценария без параметров (рис. 2) – отсутствует информация о версии WSH.

```

Командная строка
Microsoft Windows [Version 6.1.7600]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Users\gan>color f0

C:\Users\gan>Cscript F:\PRIMER_WSH\Privet.vbs //nologo
Привет Всем!

C:\Users\gan>_

```

Рис. 5. Выполнение сценария Privet.vbs с параметром nologo.

Если сценарий запускается с использованием графического интерфейса (из Windows), то задать параметры сервера можно из диалогового окна «Выполнить».

В этом случае окно с результатом работы скрипта погаснет через 5 секунд после появления.

При запуске сценария двойным щелчком по значку сценария (или по его ярлыку) параметры сервера можно определить через дополнительный файл с именем сценария и расширением wsh.

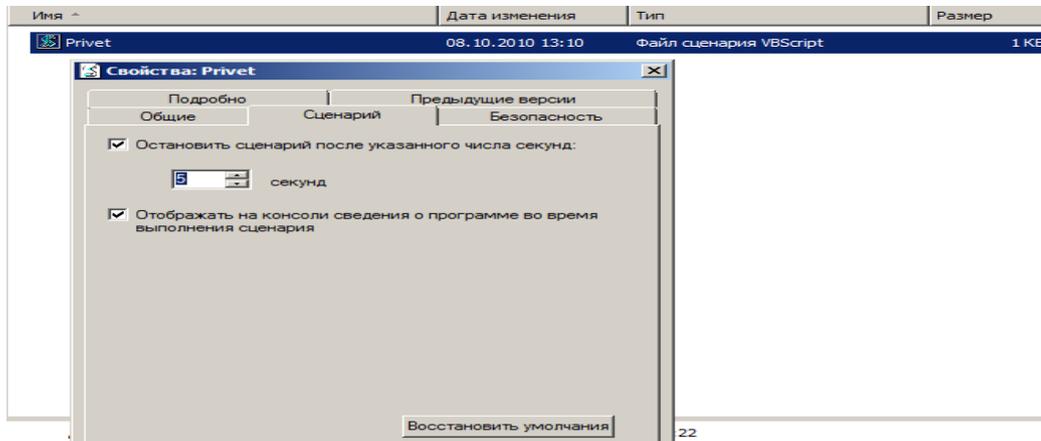


Рис.6. Параметры сервера заданы из меню пункт Properties (Свойства)

Для этого необходимо щелкнуть по файлу сценария правой кнопкой, выбрать из контекстного меню пункт Properties (Свойства) и в открывшемся окне свойств сценария установить нужные значения (рис.6). Окно закроется, если щелкнуть по кнопкам «Применить» и затем «ОК» и операционная система создаст новый файл с именем сценария и расширением wsh. Двойной щелчок по этому файла запустит исполнение сценария с параметрами, установленными на вкладке свойств сценария. Файлы с расширением wsh представляют собой текстовые файлы, похожие на ini – файлы ОС. Содержание такого файла можно получить, открыв файл с помощью текстового редактора Блокнот (рис.7).

```
[ScriptFile]
Path = F:\PRIMER_WSH\Privet.vbs
[Options]
TimeOut = 5
DisplayLogo = 1
```

Рис.7. Содержание файла с параметрами сервера для сценария Privet.vbs

Оператор Path в разделе [ScriptFile] идентифицирует файл, в котором находится подлежащий исполнению сценарий. Свойства

времени исполнения сценария определяются ключевыми словами в разделе [Options]. Величина Timeout равна значению, заданному пользователем на вкладке свойства сценария. Параметр DisplayLogo=1 активизирует вывод заставки в окне командной строки.

Вопросы для самопроверки по разделу 1.2.

1. Как можно вывести на экран все параметры консольного сервера WSH?
2. Какое расширение имеет файл ярлыка сценария WSH?
3. Приведите структуру файла-ярлыка содержащего параметры сервера WSH.
4. Какой параметр позволяет управлять временем вывода на экран диалогового окна сценария WSH?

1.3. Передача аргументов файлу сценариев

Для работы сценария из предыдущего примера никаких аргументов не требовалось. Однако чаще всего пользователь должен передавать сценарию один или несколько аргументов, таких как имя файла, значения переменных и др. На рис. 8 приведен текст программы Primer1.vbs, которая выводит все аргументы сценария на экран:

```
*****
' Язык - VBScript
' Имя файла – Primer 1.vbs
' Автор Иванов И.И.
' Дата – 12.02.2013
' Пример 2 - Сценарий предусматривающий передачу параметров
*****

text = "Передача аргументов"&VbCrLf ' VbCrLf – перевод строки
set objArgs = WScript.Arguments ' создаем объект
for i = 0 to objArgs.Count – 1 ' цикл обработки всех аргументов
text = text & objArgs(i) & VbCrLf ' запись i –го аргумент в text
next ' переход к (i + 1)-му аргументу
WScript.Echo text ' вывод аргументов на экран
```

Рис. 8. Листинг сценария Primer 1.vbs вывода аргументов на экран

При запуске этого сценария из меню «ПУСК» в окне «ВЫПОЛНИТЬ» кроме файла сценария необходимо указать аргументы к этому сценарию. Разделителем аргументов служит символ пробела. Пример запуск сценария и результаты его работы приведены на рис.9. Следует отметить, что слова “Привет всем” заключены в кавычки, чтобы передать строку с пробелами как один аргумент, иначе они воспринимаются WSH как отдельные аргументы.

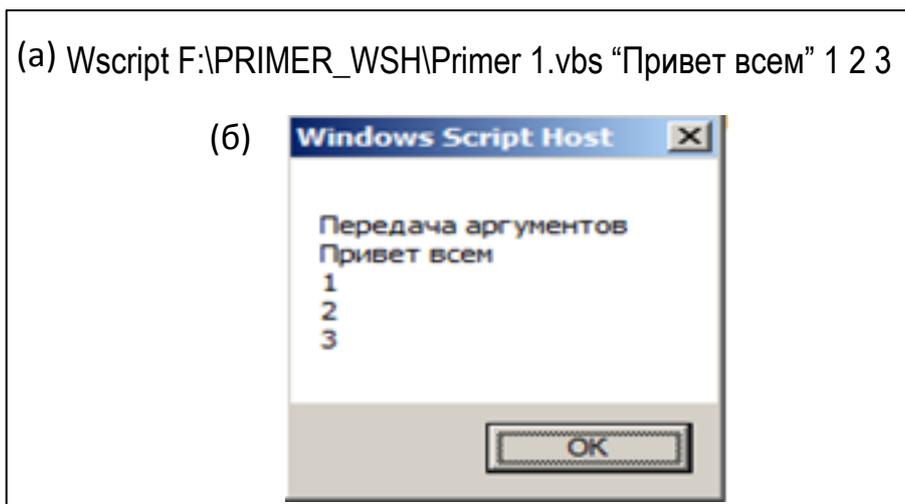


Рис. 9. Запуск сценария с аргументами (а) и результаты работы сценария (б)

Передача аргументов при запуске сценариев из командной строки осуществляется аналогично, учитывая замечания об использовании двойных кавычек.

Передавать аргументы через диалоговое окно «Пуск» не совсем удобно. Иногда, особенно в случаях, когда значения аргументов изменяются редко, удобнее использовать для передачи аргументов ярлык файла сценариев. Чтобы задать значения аргументов как свойства ярлыка необходимо:

- щелкнуть правой кнопкой значок файла сценариев и выбрать из контекстного меню команду Create Shortcut (Создать ярлык);
- далее щелкнуть правой кнопкой мыши по значку появившегося файла ярлыка и выбрать из контекстного меню Properties (Свойства);
- в текстовом поле Target (Объект) страницы свойств ярлыка необходимо добавить к команде запуска скрипта нужные аргументы.

ОС сохранит аргументы после того как окно свойств будет закрыто. После этого, если щелкнуть два раза по ярлыку сценария, то он запустится, а указанные в свойствах аргументы будут переданы сценарию.

Вопросы для самопроверки по разделу 1.3.

1. Какой элемент служит разделителем при задании аргументов для сценариев WSH?
2. Каким образом передать строку с пробелами как один аргумент сценария WSH?
3. Как задать аргументы сценария при его запуске из меню ПУСК?
4. Как задать аргументы сценария при его запуске из командной строки?

1.4. Рекомендуемая структура сценария

Требования обеспечения «читаемости» сценариев, необходимость организации процедур их отладки с использованием средств отладки обуславливает необходимость выполнения при написании сценариев определенных требований к структуре сценариев. В общем случае каждый сценарий должен содержать:

- информационный блок;
- блок инициализации;
- главное тело сценария;
- функции и процедуры.

В информационном блоке должна содержаться следующая информация: используемый для написания сценария язык (VBS), имя файла сценария, имя автора, дата создания, наименование сценария с описанием выполняемых функций.

Блок инициализации содержит объявления компонентов уровня сценария: переменные, константы, имена объектов (желательно с комментариями).

В главном теле сценария находится сам текст сценария, из главного тела сценария осуществляются вызовы функций и процедур.

В блоке функции и процедуры описываются используемые в главном теле сценария функции и процедуры.

```
' ===== Script Information Header =====
' Script Name:      ' название сценария;
' Date:            ' дата изменения;
' Author:          ' автор;
' Description:     ' описание;
' ===== Initialization Block =====
Option Explicit    ' директива, запрещающая автоматическое создание переменных
Dim <Variable>    ' объявление переменных;
Dim <Array>()      ' объявление массивов;
Set <Object>       ' подключение объектов;
' ===== Script Main Logic =====
                  ' тело сценария;
' ===== Functions and Procedures =====
                  ' процедуры и функции.
```

Рис.10. Рекомендуемая структура сценария

Для того, чтобы было удобно следовать приведенным рекомендациям при создании сценариев желательно использовать шаблоны. В качестве

шаблона возможно использование простых текстовых файлов, которые могут быть по структуре рис.10.

В угловых скобках указаны обязательные команды и параметры, значения которых изменяются в зависимости от контекста сценария.

Вопросы для самопроверки по разделу 1.4.

1. Какова общая структура файлов сценариев WSH?
2. Каково содержимое информационного блока файла сценария WSH?
3. Каково содержимое блока инициации файла сценария WSH?
4. В каком блоке находится текст сценария WSH?

Оглавление

ПРЕДИСЛОВИЕ.....	3
1. СОЗДАНИЕ И ЗАПУСК ПРОСТЕЙШИХ СЦЕНАРИЕВ	4
1.1. Серверы сценариев	4
1.2. Передача параметров серверу сценариев	7
1.3. Передача аргументов файлу сценариев	10
1.4. Рекомендуемая структура сценария	12

**Губин Александр Николаевич
Филиппов Феликс Васильевич**

**ОСНОВЫ ПРОГРАММИРОВАНИЯ СЦЕНАРИЕВ
В СРЕДЕ WSH**

Учебное пособие

Редактор

План 2013 г., п.

Подписано к печати
Объем усл.-печ. л. Тираж экз. Заказ

Издательство СПбГУТ. 191186 СПб., наб. р. Мойки, 61

Отпечатано в

