


Федеральное государственное бюджетное образовательное
учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Н. П. ОГАРЁВА»

Факультет математики и информационных технологий
Кафедра фундаментальной информатики

УТВЕРЖДАЮ

Зав. кафедрой

к. ф.-м. н., доцент

 А. Г. Смольянов
(подпись)

«20» июня 2016 г.

БАКАЛАВРСКАЯ РАБОТА

**ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ СЦЕНАРИЕВ WINDOWS ДЛЯ
РЕШЕНИЯ ОФИСНЫХ ЗАДАЧ**

Автор бакалаврской работы


(подпись)

27.05.2016
(дата)

Я. Е. Вдовина

Обозначение бакалаврской работы БР-02069964-02.03.02-05-16

Направление 02.03.02 Фундаментальная информатика и информационные
технологии

Руководитель работы

к. ф.-м. н., доцент


(подпись)

15.06.2016
(дата)

А. В. Попов

Нормоконтролер

к. пед. н., доцент


(подпись)

15.06.2016
(дата)

И. А. Карпюк

Рецензент

к.ф.-м.н., доцент


(подпись)

14.06.16г.
(дата)

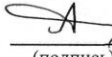
Л. А. Сухарев

Саранск

2016

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н. П. ОГАРЁВА»

Факультет математики и информационных технологий
Кафедра фундаментальной информатики

Зав. кафедрой
к. ф.-м. н.

(подпись) А. Г. Смольянов
«20» июня 2016 г.

ЗАДАНИЕ НА ВЫПУСНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студент Вдовина Яна Евгеньевна

1 Тема Практическое применение сценариев Windows для решения офисных задач

Утверждена приказом № 934-С от 12.02.2016 г.

2 Срок представления работы к защите 27.05.2016

3 Исходные данные для научного исследования литература по теме исследования

4 Содержание выпускной квалификационной работы

4.1 Офисные задачи

4.2 Сценарии Windows Script Host как инструмент автоматизации офисных задач

4.3 Сценарий для формирования и учета почтовых отправлений

5 Приложения Пользовательский интерфейс; код программы

Руководитель работы  12.02.2016 А. В. Попов

подпись, дата

Задание принял к исполнению  12.02.2016 Я. Е. Вдовина

подпись, дата

Реферат

Выпускная бакалавриатская работа содержит 80 страниц, 5 таблиц, 25 рисунков, 8 листингов, 12 использованных источников.

Объектом исследования являются СЦЕНАРИИ WINDOWS SCRIPT HOST, HTA, ОБЪЕКТНЫЕ МОДЕЛИ WORD И EXCEL, VBA.

Цель работы – систематизация и обобщение теоретических и практических знаний по теме исследования, разработка утилиты для автоматизации оформления и учета почтовых отправок организации (с использованием сценария на WSH в формате HTA и добавлением макросов VBA в xls-файл).

В процессе работы были освоены приемы создания сценариев на WSH, изучены методы доступа к объектам Microsoft Office, разработано приложение в формате HTA.

Область применения – в практической деятельности организации.

Степень внедрения – частичная.

Эффективность – снижение трудозатрат при выполнении операций по оформлению и учету почтовых отправок в организации.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Офисные задачи	7
1.1 Пакеты программ для офисной деятельности	8
LibreOffice	9
Apache OpenOffice	9
Corel Office	9
Ashampoo Office	10
SoftMaker Office	10
Microsoft Office	10
1.2 Сравнение офисных пакетов	14
1.3 Возможности автоматизации в Microsoft Office	16
1.4 Объектная модель Microsoft Word	18
1.5 Объектная модель Microsoft Excel	20
2 Сценарии Windows Script Host как инструмент автоматизации офисных задач	21
2.1 Работа с объектными моделями Microsoft Office	24
2.2 Технология HTA для создания графического интерфейса	30
3 Сценарий для формирования и учета почтовых отправлений	34
3.1 Формулировка задачи	34
3.2 Структура приложения	35
ЗАКЛЮЧЕНИЕ	36
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	37
ПРИЛОЖЕНИЕ А	39
ПРИЛОЖЕНИЕ Б	47

ВВЕДЕНИЕ

В настоящее время практически во всех направлениях профессиональной деятельности занято большое количество офисных сотрудников, выполняющих повторяющиеся однотипные задачи по поддержке делопроизводства, оформлению отчетности, обработке корреспонденции и т.п. Для облегчения и автоматизации выполнения офисных задач используют специальные программные пакеты, например Microsoft Office, LibreOffice, OpenOffice, Apache OpenOffice, Corel Office, Ashampoo Office, SoftMaker Office и др. В первой главе более подробно рассмотрены возможности наиболее популярных офисных пакетов.

Во многих офисных пакетах поддерживаются средства макропрограммирования, с помощью которых можно автоматизировать работу с документами определенного типа (текстовыми документами, электронными таблицами или базами данных). Кроме того, пакет Microsoft Office имеет дополнительное преимущество – он не только позволяет автоматизировать работу с помощью макросов внутри документов, но и предоставляет доступ к своей объектной модели внешним программам и сценариям Windows Script Host (WSH). Таким образом, сценарии WSH, которые пишутся на языках VBScript или JScript и выполняются непосредственно в операционной системе, являются инструментом, позволяющим решать задачи автоматизации офисных задач с использованием специфических возможностей операционной системы Windows (системный реестр, ярлыки и специальные папки и т.д.). Для придания интерактивности подобным сценариям с помощью графического интерфейса можно оформить их в формате HTML Application (HTA) – примеры приведены во второй главе.

В третьей главе рассмотрено решение одной практической задачи – автоматизация оформления и учета почтовых посылок, отправляемых из организации по различным адресатам. Создан НТА-сценарий, который, взаимодействуя с объектами Microsoft Word и Microsoft Excel, позволяет вести реестр отправленных посылок и формировать ежедневные отчетные формы.

1 Офисные задачи

К типичным офисным подразделениям организации относятся обеспечивающие (вспомогательные) службы, не занятые основным производством: кадровая служба, канцелярия, бухгалтерия, секретариат, планово-экономический отдел и т.п. Основные задачи, которые часто приходится решать в офисе:

- Организация делопроизводства и документооборота.
- Создание различных отчетов.
- Составление планов и расписаний.
- Обмен информацией между отделами офиса, разными филиалами и с другими предприятиями.

При решении этих задач возникают следующие типовые процессы и процедуры:

- Обработка полученной и исходящей информации, работа с электронными письмами, документацией в различных форматах.
- Сбор и последующий анализ данных, составление отчетов за определенный период с учетом различных параметров.
- Создание информационных архивов, обеспечивающих сохранность информации и возможность получения в дальнейшем запрашиваемых данных.

Для упрощения вышеперечисленных действий и повышения производительности работы офисных сотрудников используют различные программные и аппаратные решения, позволяющие автоматизировать офисную деятельность в соответствии с рисунком 1. Они позволяют организовать и поддерживать коммуникационные процессы как внутри организации, так и с внешней средой на базе компьютерных сетей и других современных средств передачи и работы с информацией [5].

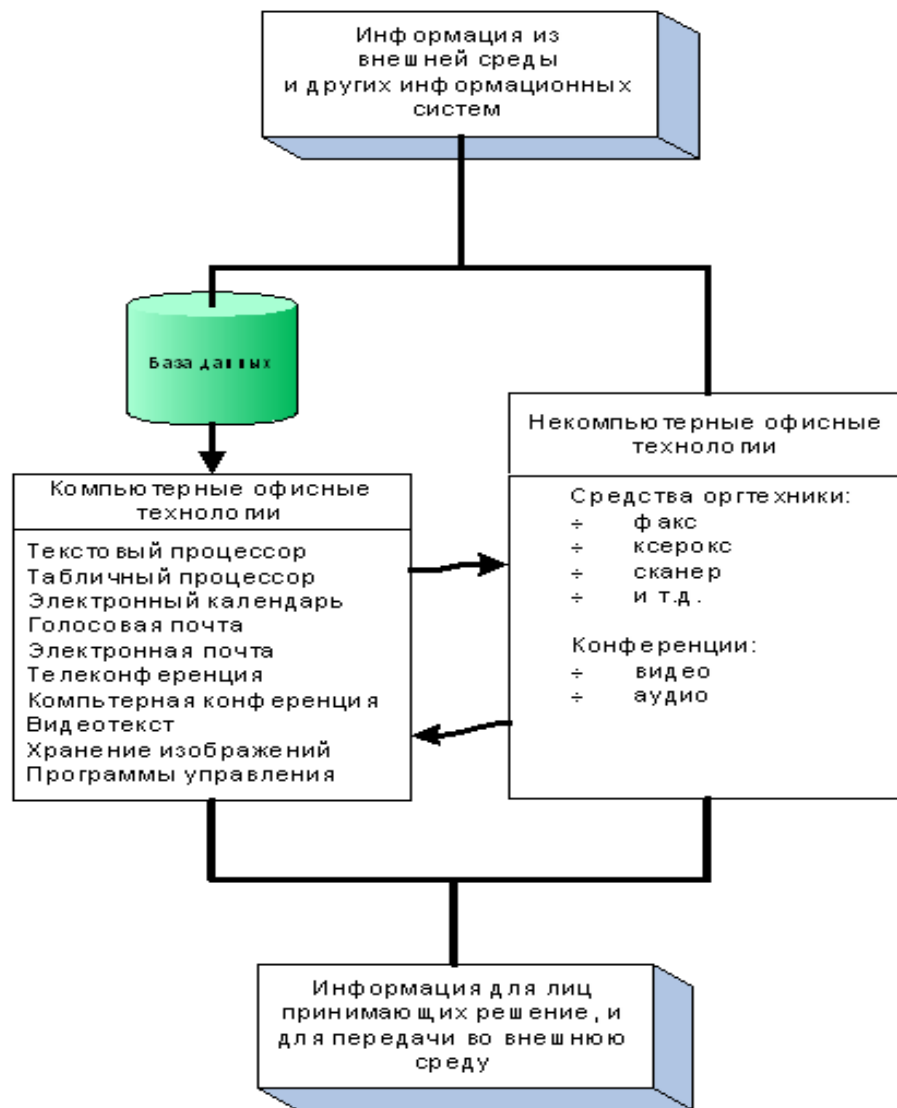


Рисунок 1 – Основные компоненты автоматизации офиса

1.1 Пакеты программ для офисной деятельности

В настоящее время существует множество программных продуктов, позволяющих автоматизировать различные виды офисных работ: программы для организации телеконференций, обработки электронной почты, электронного документооборота, специализированные программы управленческой деятельности и планирования. Для решения базовых задач по созданию и обработке текстовых документов, электронной почты и таблиц выбирается тот или иной офисный пакет, например Microsoft Office,

LibreOffice, OpenOffice, Apache OpenOffice, Corel Office, Ashampoo Office, SoftMaker Office. Кратко рассмотрим возможности наиболее распространенных офисных пакетов.

LibreOffice

Пакет LibreOffice имеет в своем интерфейсе 30 языков, поддерживается популярными операционными системами (Linux, Microsoft Windows, Mac OS X). Пакет бесплатен для пользователей, имеет в своем составе текстовый редактор Writer, табличный процессор Calc, программу для подготовки презентаций Impress, векторный графический редактор Draw, СУБД Base и редактор формул Math. Основным используемым форматом является OpenDocument, также поддерживается формат файлов Microsoft Office (doc).

Apache OpenOffice

Пакет Apache OpenOffice похож на LibreOffice (построены эти пакеты на одном ядре), он также является бесплатным для пользователей и находится в свободном доступе. Поддерживает основной формат OpenDocument. В состав пакета входят текстовый редактор Writer, табличный процессор Calc, программа для подготовки презентаций Impress, векторный графический редактор Draw, СУБД Base и редактор формул Math. Существуют версии пакет OpenOffice для Linux, Microsoft Windows, Mac OS X, также его можно запускать без инсталляции с внешнего носителя.

Corel Office

Пакет Corel Office является платным, одна лицензия на пакет составляет 45 евро. В состав пакет входят текстовый редактор, электронная таблица и программа для подготовки презентаций.

Особенностью этого пакета является то, что он будет работать и на старых компьютерах с операционной системой Windows XP и разрешением экрана 800x600, поэтому он хорошо подходит для нетбуков. Также в Corel

Office реализована интегрированная поддержка облачных решений через сервис DropBox.

Дистрибутив с расширенной функциональностью Corel WordPerfect Office X6 доступен только на английском языке [6].

Ashampoo Office

Пакет Ashampoo Office содержит три приложения: текстовый редактор TextMaker, табличный редактор PlanMaker и программу для подготовки презентаций Presentations. Данные приложения можно установить на съемный носитель и пользоваться ими в режиме portable.

Пакет поддерживает форматы Microsoft Office, также можно сохранять документы в формате pdf. Ashampoo Office является платным (1500 рублей за лицензию).

SoftMaker Office

Пакет содержит текстовый редактор TextMaker, инструмент для работы с электронными таблицами PlanMaker, программу для подготовки презентаций Presentations, средство программирования BasicMaker. Предназначен для операционных систем Windows, Linux, FreeBSD, Windows CE и Pocket PC. Стоимость пакета составляет около 100 долларов, пробный период бесплатного использования – 30 дней.

SoftMaker Office поддерживает форматы Microsoft Office и OpenDocument Format, имеет многоязычный интерфейс (14 языков, включая русский).

Microsoft Office

Общепринятым стандартом и лидером среди офисных приложения является пакет Microsoft Office (MS Office). Этот продукт компании Microsoft имеет реализации для операционных систем Microsoft Windows, Windows Phone, Android, OS X, iOS. В состав пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными

таблицами, базами данных [4]. Первые версии программ, входящих в Microsoft Office, появились еще в начале 1980-х годов (таблица 1).

Таблица 1 – История пакета Microsoft Office

Год выпуска	Версия	Программы в составе пакета	Особенности	Год прекращения поддержки
1	2	3	4	5
1980	Xenix	Multi-Tool Word	Первый текстовый редактор	1983
1983	Xenix	Multiplan, Microsoft Word	Первая программа редактирования таблиц.	1990
1990	Microsoft Office 1.0	Word 1.1, Excel 2.0, PowerPoint 2.0	Стал лидером на рынке.	1993
1992	Microsoft Office 3.0	Word 2.0c, Excel 4.0a, PowerPoint 3.0, Microsoft Mail	Выпуск пакета на CD. Переименован в Office 92.	1995
1994	Microsoft Office 4.3	Word 6.0, Excel 5.0, PowerPoint 4.0, Mail 3.2, Access 2.0 Pro.	Обновленный интерфейс.	1999
1996	Microsoft Office 97	Word 97, PowerPoint 97 и Mail 97	Поддержка встроенного языка программирования Visual Basic for Applications.	2001
1999	Microsoft Office 2000	Word, PowerPoint, Excel 2000	Последняя версия пакета, не запрашивающая активации продукта.	2009

Продолжение таблицы 1

1	2	3	4	5
2001	Microsoft Office XP	Word, Excel, PowerPoint, Outlook, Small Business Tools, Publisher, Access, Frontpage, PhotoDraw 2000	Сохранение документов в Excel и PowerPoint, даже после сбоя в их работе.	2011
2003	Microsoft Office 2003	Word, Excel, PowerPoint, Outlook, Модуль Диспетчер контактов, Publisher, Access	К документу в общей рабочей области могут иметь доступ несколько пользователей.	2014
2007	Microsoft Office 2007	Word, Excel, Outlook, PowerPoint, Access, Publisher, InfoPath, OneNote, Project	В Word можно добавлять ссылки единственным щелчком. В Excel повысились скорость и мощность обработки данных, увеличились размеры листа.	2014
2010	Microsoft Office 2010	Word, Excel, PowerPoint, Outlook, Publisher, Access, OneNote, InfoPath, Communication, SharePoint Workspace	Содержит функции, которые позволяют администратору настраивать, проверять, развертывать и защищать установки Office в организации.	2015
2012	Microsoft Office 2013	Word, Excel, PowerPoint, Outlook, Publisher, Access, OneNote, InfoPath, Lync, Project, Visio.	Соединен с облачным хранилищем, Skype. Вставка онлайн –видео, работа с pdf-файлами и их редактирование.	

Окончание таблицы 1

1	2	3	4	5
2016	Microsoft Office 2015	Word, Excel, PowerPoint, Outlook, Publisher, Access, OneNote, InfoPath, SharePoint Workspace, Lync, Web Apps, OneDrive.	Между Word, PowerPoint и OneNote появилась возможность совместного редактирования документов в реальном времени.	

В повседневной офисной работе чаще всего используются текстовый редактор Word, электронная таблица Excel, система управления базами данных Access, программа для создания презентаций PowerPoint, почтовый клиент и органайзер Outlook. Опишем более подробно некоторые из этих программ.

Microsoft Word – текстовый редактор, который позволяет создавать и редактировать документы разной сложности. Поддерживает различные модули и шаблоны, основным форматом в настоящее время является docx (Open XML, который представляет собой ZIP-архив, содержащий текст в виде XML). В Microsoft Word интегрирован язык программирования VBA.

Microsoft Excel – табличный редактор, позволяющий создавать электронные таблицы разной сложности и при помощи встроенных средств выполнять всевозможные операции над данным в таблице. Основные группы выполняемых в Excel операций:

- Ввод данных с клавиатуры и из базы данных.
- Обработка данных (сортировка, автоматическое формирование итогов, копирование и перенос данных и т.д.).
- Вывод информации в печатном виде, экспорт в другие системы и в базы данных.
- Оформление информации в виде диаграмм и графиков.

- Проведение инженерных, финансовых, статистических расчетов;
- Проведение математического моделирования.

Microsoft Excel поддерживает язык макропрограммирования VBA [9].

Microsoft Access – приложение для работы с базами данных. В автоматизированном офисе база данных может концентрировать в себе информацию о производственной системе фирмы и технологии обработки данных на операционном уровне [1]. Поступление информации в базу данных может осуществляться также и с внешнего окружения предприятия.

Информация из базы данных поступает на вход компьютерных приложений (программ): текстовый процессор, табличный процессор, электронная почта, компьютерные конференции и пр. Любое компьютерное приложение автоматизированного офиса обеспечивает работникам связь друг с другом и с другими фирмами [10].

Все вышеперечисленные программы могут взаимодействовать между собой, они образуют интегрированный пакет, единый удобный инструмент. Отличительная особенность этих приложений – общий интерфейс пользователя. Взаимодействие программ осуществляется на уровне документов, то есть открытие и редактирование документа может происходить в любом из приложений, не обязательно в том, в котором был создан.

1.2 Сравнение офисных пакетов

Рассмотрев несколько офисных пакетов, можно прийти к выводу, что многие альтернативные офисные пакеты, особенно бесплатные, очень сильно похожи друг на друга, как функционально, так и внешне. Происходит это из-за того, что большинство разработчиков не создают офисные пакеты с нуля, а стараются модифицировать уже готовые программы.

Сравним два офисных пакета: платный Microsoft Office и бесплатный LibreOffice:

1. Microsoft Office – кроссплатформенный продукт, входящи в него программные продукты имеют версии для Android, iOS, Windows Phone. Версии пакета LibreOffice на перечисленных платформах не существует.
2. Microsoft Office обладает интеграцией с почтовыми программами (приложение Outlook), тогда как у LibreOffice нет такой возможности.
3. Пользовательский интерфейс в Microsoft Office представлен в виде современного дизайна, с поддержкой сенсорных экранов. В LibreOffice реализованы в виде классические меню с выпадающими списками.
4. Microsoft Office обладает «облачной» версией – Office web apps. У LibreOffice данный продукт находится только в стадии разработки.
5. В Microsoft Office реализовано синхронное редактирование документов несколькими пользователями, а также импорт форматов MEZ, WNZ, PCZ, CGM. Пакет LibreOffice перечисленные возможности не поддерживает.
6. Лицензия у MS Office закрытая, проприетарная. Пакет LibreOffice имеет открытый исходный код.
7. Пакет MS Office платный, Libre Office – полностью бесплатный.
8. Электронные таблицы Calc в LibreOffice содержит свой язык макросов, не совместимый с и макросами созданные в таблицах MS.

Таким образом, по функциональности пакет Microsoft Office превосходит Libre Office, поэтому в настоящее время MS Office распространен шире, несмотря на то, что этот пакет нужно покупать. Нужно отметить, что пакет MS Office в зависимости от входящих в него приложений представлен в нескольких вариантах, имеющих различную цену: для студентов, для дома расширенный, для малого бизнеса расширенный, для дома и учебы, для дома и бизнеса, профессиональный.

1.3 Возможности автоматизации в Microsoft Office

Одной из возможностей операционной системы Microsoft Windows является технология автоматизации, благодаря которой различные приложения и сценарии могут получать доступ к свойствам и методам объектов, создаваемых в других приложениях. Такие приложения, предоставляющие объекты для общего использования, называются COM-серверами, их наименование происходит с помощью специальных символьных обозначений – программных идентификаторов. Эти идентификаторы позволяют системе определить каталог расположения выполняемого файла сервера автоматизации, а также запустить этот файл и создать экземпляр нужного объекта.

Программы, входящие в пакет Microsoft Office, обладают встроенным средством автоматизации при помощи макросов. В большинстве случаев они представлены в виде небольших по объему скриптов, благодаря которым происходят какие-либо операции над документами.

Операции, производимые вручную над документом, можно сохранить в макрос при помощи макрорекордера, который встроен в каждое приложение пакета MS Office. В результате будет сгенерирован скрипт на языке VBA (Visual Basic for Applications) — диалекте языка Visual Basic, расширяющем возможности Visual Basic и предназначенном специально для работы с приложениями Microsoft Office и другими приложениями от Microsoft [3]. Написанный код на VBA сохраняется в самом файле с документом (документ Word, книге Excel, база данных Access и т.д.).

Рассмотрим несколько примеров автоматизации приложений Word и Excel при помощи VBA.

- Создание документа Word при помощи метода Add коллекции Documents объекта Application:

App.Documents.Add;

- Открытие созданного документа при помощи метода Open коллекции Documents:

App.Documents.Open('C:\Word1.doc');

- Активация документа с порядковым номером 1:

App.Documents.Item(1).Activate;

- Вставка текста в Word при помощи объекта `Range()`, где в скобках указывается начало и конец символов в тексте, либо номер абзаца:

Var Rng : Variant;

...

Rng := App.ActiveDocument.Range(2,4);

- Вставка текста при помощи методов `InsertBefore` или `InsertAfter` объекта `Range`:

Rng.InsertAfter('Это вставляемый текст');

- Создание новой рабочей книги Excel и открытие существующей книги:

App.WorkBooks.Add;

App.Documents.Open('C:\Excel1.xls');

- Активизация второй открытой рабочей книги:

App.WorkBooks[2].Activate;

- Закрытие рабочей книги:

App.WorkBooks[2].Close; или

App.ActiveWorkBook.Close;

- Запросы к листам и ячейкам при помощи коллекции `Worksheets` объекта `WorkBook`:

App.WorkBooks[1].Worksheets[1].Name := 'Страница_1';

Внутренний уровень пакета MS Office представляет собой совокупность связанных между собой объектов, которые обладают наборами свойств и методов. Совокупность объектов и связей между ними называется объектной моделью [11]. Каждому приложению MS Office соответствует своя объектная модель. Все существующие компоненты офисных программ, как и сами приложения, являются объектами. Зная архитектуру объектных моделей

Иерархия, в соответствии с рисунком 2, обладает перекрытием, так как для доступа к объекту одного типа используются множество способов. У объектов верхнего (таблица 2) уровня пространство имен содержит фундаментальные и основные классы, определяющие широко используемые значимые и ссылочные типы данных, события, интерфейсы, атрибуты и обработку исключений [7].

Таблица 2 – Описание объектов верхнего уровня

Название объекта	Описание объекта
1	2
Application	Его элементы применяются к Word как к единому целому. Свойства и методы данного объекта можно использовать для управления средой Word.
Document	Представляет документ и все его содержимое. При открытии существующего документа или создании нового документа вы создаете новый объект Microsoft.Office.Interop.Word.Document, который добавляется в коллекцию T:Microsoft.Office.Interop.Word.Documents объекта Application. Документ, который находится в фокусе, называется активным документом.
Selection	При выполнении операции в пользовательском интерфейсе Word, например при выделении текста полужирным шрифтом, выбирая или выделяя текст, а затем применяете форматирование. Если ничего не выбрано, данный объект представляет точку вставки. Кроме того, выделение может охватывать несколько блоков текста, которые не являются смежными.

Окончание таблицы 2

1	2
Range	<p>Объект Range не обязательно должен быть единственным. В одном документе можно определить несколько объектов Range. Обладает характеристиками:</p> <ul style="list-style-type: none"> • Может включать в себя одну точку вставки, диапазон текста или весь документ; <ul style="list-style-type: none"> • Содержит непечатаемые символы; • Может быть областью, представляемой текущим выделением, или может представлять область, отличную от текущего выделения; <ul style="list-style-type: none"> • Он не отображается в документе; • Он не сохраняется вместе с документом и существует только во время выполнения кода. <p>При вставке текста в конец диапазона Word автоматически расширяет диапазон на объем вставленного текста.</p>
Bookmark	<p>Закладки можно использовать, чтобы пометить расположение в документе или в качестве контейнера для текста в документе.</p> <p>Имеет следующие свойства:</p> <ul style="list-style-type: none"> • Можно задать имя закладки; • Объекты Microsoft.Office.Interop.Word.Bookmark сохраняются вместе с документом. Следовательно, при остановке выполнения кода или закрытии документа они не удаляются. • Закладки можно скрывать или делать видимыми, устанавливая значение false или true

1.5 Объектная модель Microsoft Excel

Объектная модель Excel (рисунок 3) похожа объектную модель Word, здесь объекты также расположены иерархически. Главной отличительной чертой Word от Excel является наличие на рабочей области листа ячеек, которая обладает уникальным именем (имя столбца и номера строки).

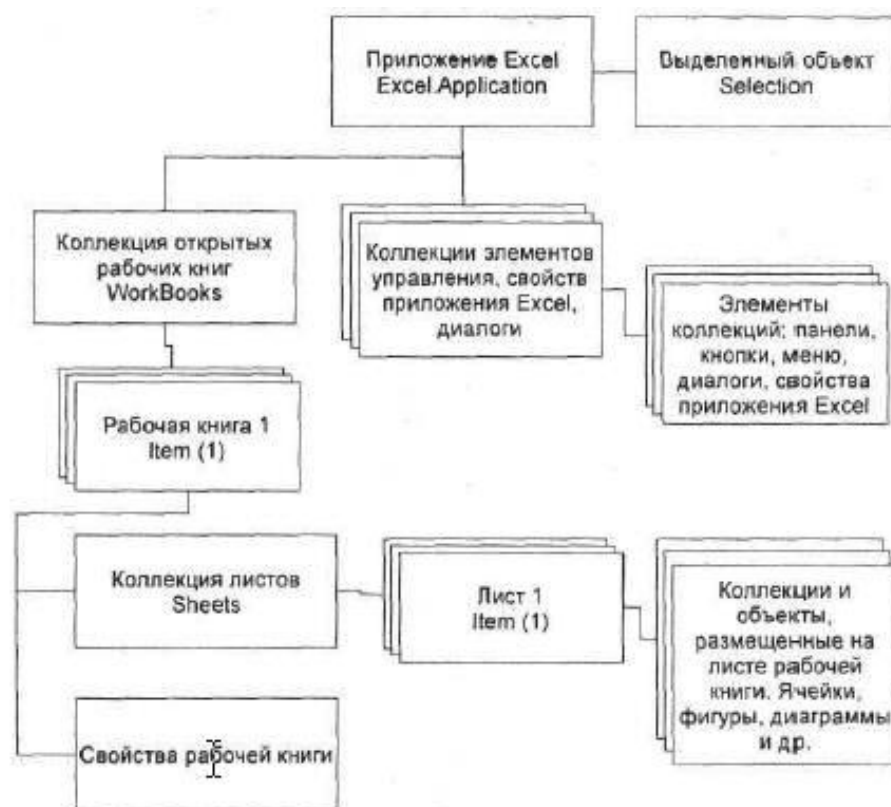


Рисунок 3 – Объектная модель Excel

2 Сценарии Windows Script Host как инструмент автоматизации офисных задач

Автоматизировать выполнение различных задач в пакете Microsoft Office можно не только макросами на языке VBA, но и с помощью имеющегося в операционной системе Windows стандартного инструмента автоматизации – сценариев Windows Script Host (WSH).

Сценарии WSH поддерживают два встроенных в Microsoft Windows языка программирования – Visual Basic Script Edition (VBScript) и Java Script Edition (JScript). Создаются и сохраняются такие сценарии в виде обычных текстовых файлов с расширением js или vbs.

Объем сценария ограничен только наибольшим размером файла в соответствующей файловой системе, поэтому некоторые сценарии могут

содержать в себе сотни строк. Для работы таких сценариев не нужен браузер, они запускаются в операционной системе как обычные исполняемые файлы. Каждая версия операционной системы Windows поддерживает определенные версии WSH (таблица 3).

Таблица 3 – Поддержка версий WSH в Windows

Версия	Операционная система
WSH 1.0	Windows 95, Windows 98, Windows NT 4.0
WSH 2.0	Windows 2000, Windows ME
WSH 5.6	Windows XP, Windows 2003 Server
WSH 5.7	Windows Vista, Windows 7, Windows 8.1
WSH 5.8	Windows 7, Windows 8.1

Основные преимущества сценариев WSH:

- Эти сценарии поддерживаются во всех версиях операционной системы Windows без установки дополнительного программного обеспечения.
- Языки VBScript и Jscript являются развитыми языками сценариев, которые применяются также при написании клиентских веб-приложений.
- Имеется возможность написание скриптов и на других языках сценариев (Perl, Python и т.д.), для которых необходимо устанавливать соответствующий модуль поддержки.
- В операционной системе зарегистрированы собственные объекты WSH, с помощью которых можно работать с сетевыми ресурсами, переменными среды, системным реестром, ярлыками и специальными папками Windows, а также запускать другие приложения и управлять их работой [7] (таблица 4).

- Можно использовать службы других приложений-серверов автоматизации, зарегистрировавших в операционной системе свои объекты (например, объектные модели приложений Microsoft Office).

Таблица 4 – Объекты WSH

Имя объекта	Описание
1	2
WScript	Это главный объект WSH, который служит для создания других объектов или связи с ними, содержит сведения о сервере сценариев, а также позволяет вводить данные с клавиатуры и выводить информацию на экран или в окно Windows.
WshArguments	Обеспечивает доступ ко всем параметрам командной строки запущенного сценария или ярлыка Windows.
WshNamed	Обеспечивает доступ к именованным параметрам командной строки запущенного сценария.
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.


```

//Создаем объект для работы с Word
var wrd = new ActiveXObject("Word.Application");

//Создаем новый документ
wrd.Documents.Add();

//Получаем указатель на активный документ
adoc=wrd.ActiveDocument;

//Указатель на область в документе
myRange = adoc.Range(Start=0, End=0)

//Активизируем Ворд
wrd.Visible = true;

wrd.Activate();

```

В результате выполнения листинга 1 в программе Word будет открыт новый документ (рисунок 4).

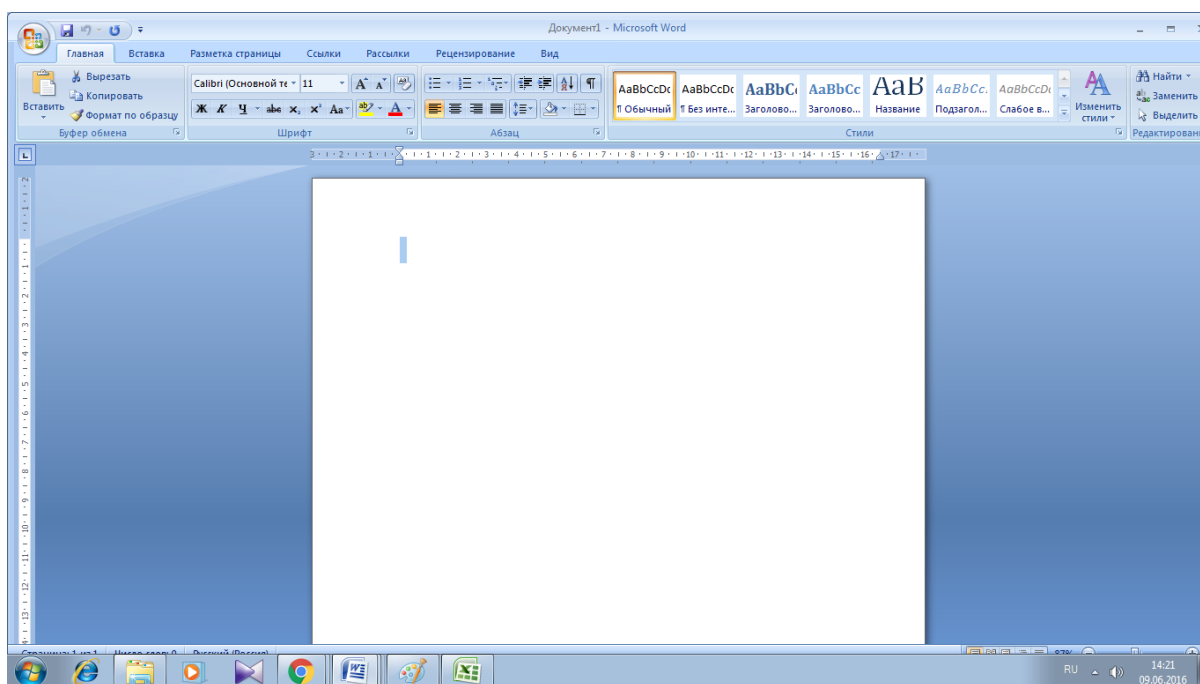


Рисунок 4 – Документ Word

Листинг 2. Создание Word файла с данными.

```
/* **** */
/* Имя: WDD.js */
/* Язык: JScript */
/* Описание: Создание файла Word с таблицей */
/* **** */

var wrd = new ActiveXObject("Word.Application");
wrd.Documents.Add();
adoc=wrд.ActiveDocument;
myRange = adoc.Range(Start=0, End=0)
wrd.Visible = true;
wrd.Activate();

//Создаем таблицу с одной строкой и тремя столбцами
tb0=adoc.tables.Add(Range=myRange, NumRows=1, NumColumns=3);
//Задаем ширину столбцов
tb0.Columns(1).Width = 80;
tb0.Columns(2).Width = 160;
tb0.Columns(3).Width = 160;
//Заполняем заголовок таблицы
tb0.Cell(1,1).Range.InsertAfter("Название свойства");
tb0.Cell(1,2).Range.InsertAfter("Значение");
tb0.Cell(1,3).Range.InsertAfter("Описание");
```

В результате выполнения сценария WDD.js в программе Word будет открыт новый документ и в нем будет напечатана таблица (рисунок 5).

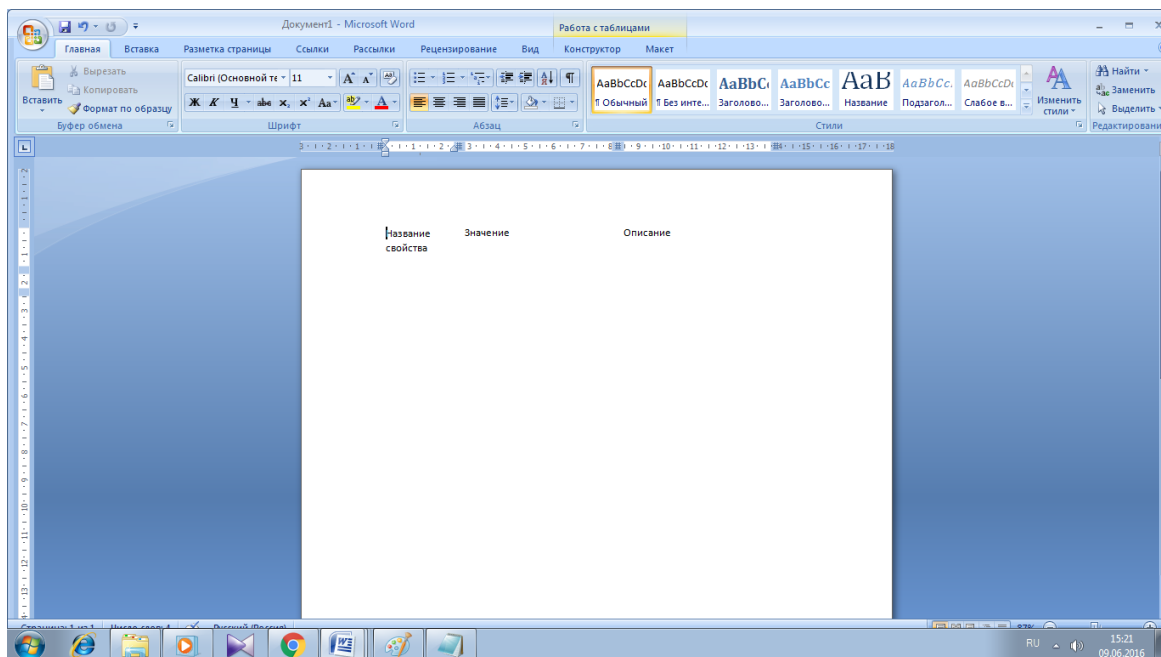


Рисунок 5 – Результат выполнения сценария WDD.js

Листинг 3. Создание Excel файла.

```

/*****/

/* Имя: EX.vbs */
/* Язык: VBScript */
/*Описание: Создание файла Excel */
/*****/

// Создаем объект для работы с Excel
Dim objXL
Set objXL = WScript.CreateObject("Excel.Application")
//Делаем окно видимым
objXL.Visible = TRUE
objXL.WorkBooks.Add

```

В результате выполнения сценария EX.vbs будет запущена программа Excel с пустой таблицей (рисунок 6).

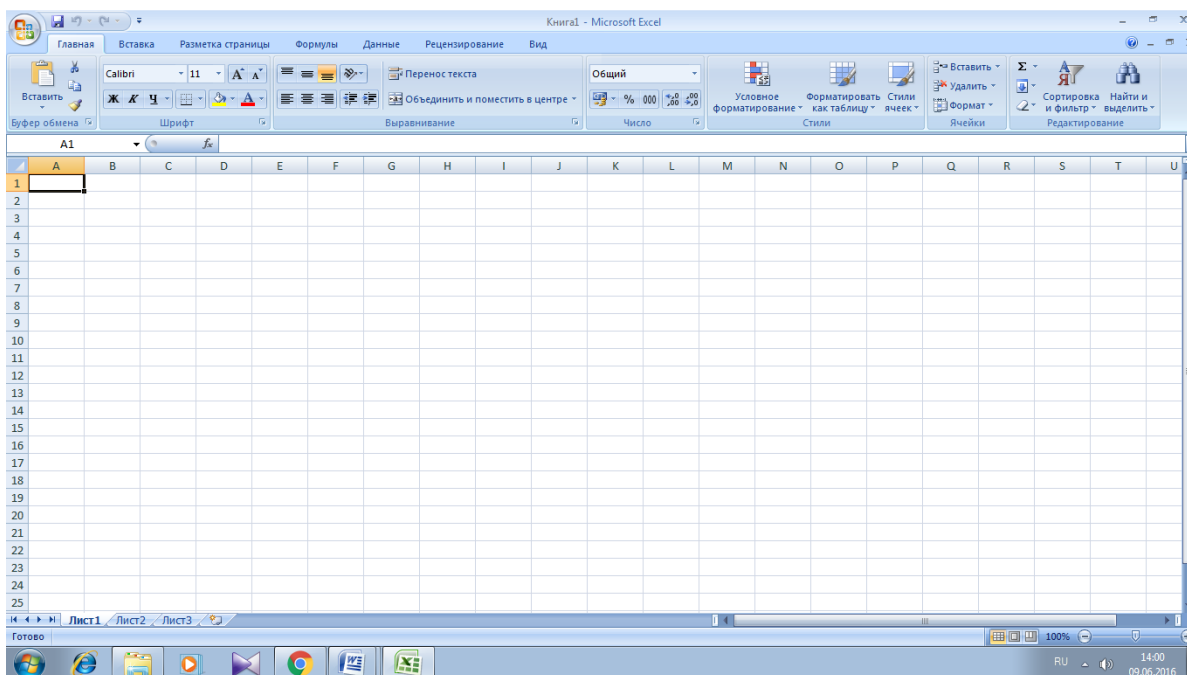


Рисунок 6 – Результат выполнения сценария EX.vbs

Листинг 4. Создание Excel файла с данными.

```

/*****/

/* Имя: EXX.vbs */
/* Язык: VBScript */
/*Описание: Создание файла Excel с данными */
/*****/

// Создаем объект для работы с Excel
Dim objXL
Set objXL = WScript.CreateObject("Excel.Application")
//Делаем окно видимым
objXL.Visible = TRUE
objXL.WorkBooks.Add
//Устанавливаем ширину первых трех колонок
objXL.Columns(1).ColumnWidth = 20
objXL.Columns(2).ColumnWidth = 30
objXL.Columns(3).ColumnWidth = 40

```

//Первая строка будет у нас шапкой таблицы

//Заполняем значения ячеек

objXL.Cells(1, 1).Value = "Название свойства"

objXL.Cells(1, 2).Value = "Значение"

objXL.Cells(1, 3).Value = "Описание"

objXL.Range("A1:C1").Select

objXL.Selection.Font.Bold = True

objXL.Selection.Interior.ColorIndex = 1

objXL.Selection.Interior.Pattern = 1

objXL.Selection.Font.ColorIndex = 2

objXL.Columns("B:B").Select

objXL.Selection.HorizontalAlignment = &hFFFFEFD

В результате выполнения сценария EXX.vbs в программе Excel будет создана таблица с тремя заголовками (рисунок 7).

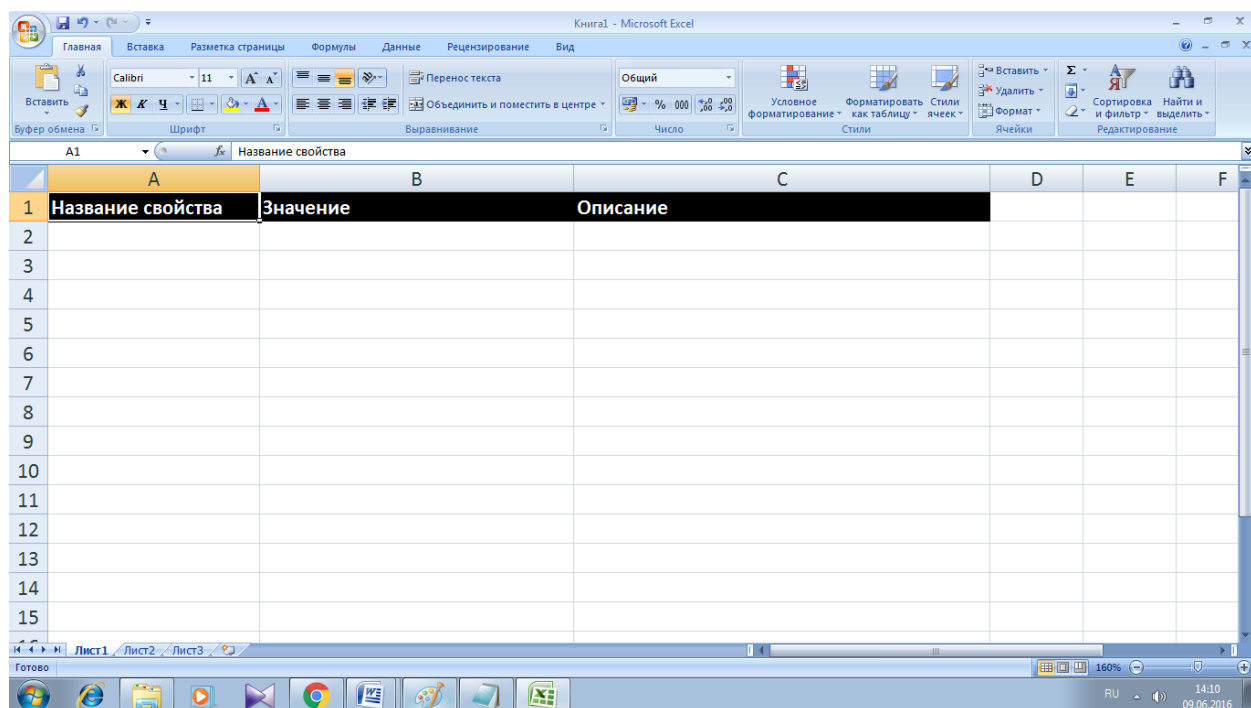


Рисунок 7 – Результат выполнения сценария EXX.vbs

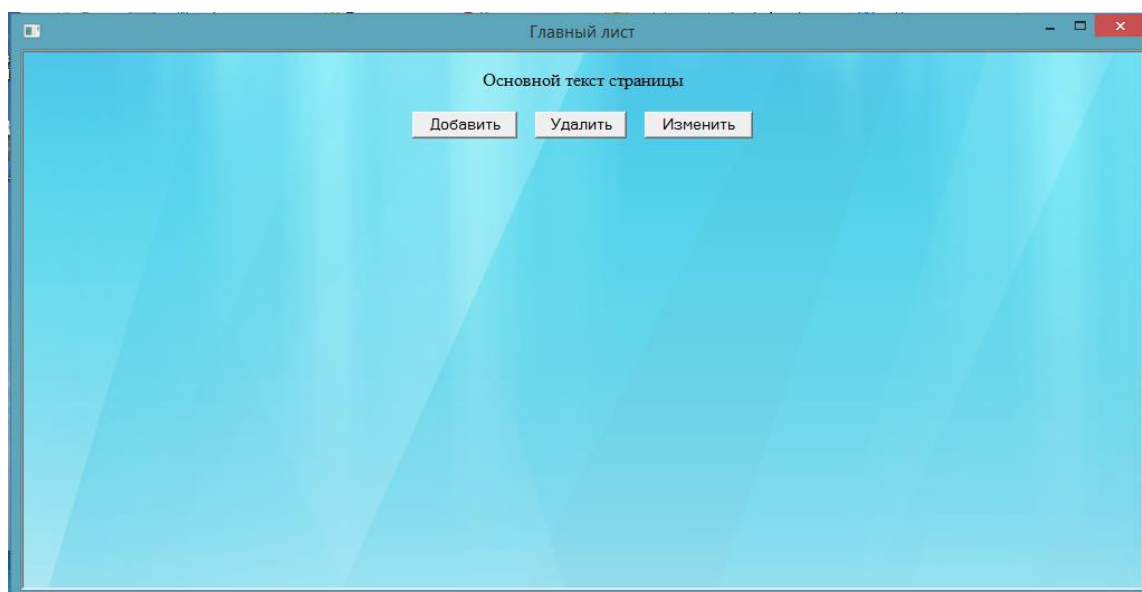


Рисунок 8 – Окно программы

HTA-файл через браузер открыть нельзя, так как он не интерпретирует расширение .hta. Сама система Windows определяет файл как исполняемый, и запуск происходит по щелчку на этот файл или через командную строку.

Благодаря тому, что стандарт HTA позволяет исполнять сценарии Java и VB, в этом формате можно создать много полезных приложений, от простого календаря и конвертора валют до ньюсридера и менеджера онлайн-доступа к блогам, сайтам и форумам. Поэтому, обладая даже небольшим опытом создания веб-страниц и написания простых Java-скриптов, пользователь может создавать HTA-файлы, которые ни по функциональности, ни по дизайну не уступают небольшим программам, написанным на более серьезных языках [9].

Таблица 5 – Атрибуты HTA документов

Атрибут	Описание
1	2
applicationName	Имя приложения, использующегося для идентификации при исполнении HTA-документа
border	Вид обрамления окна; среди возможных значений – thin, dialog, none, thick

Продолжение таблицы 5

1	2
borderStyle	Стиль обрамления окна; среди возможных значений - complex, normal, raised, static, sunken
caption	Указывает, будет ли показывать заголовок окна (yes/no)
icon	Путь к пиктограмме, которая будет показана в заголовке; допустимые параметры пиктограммы – 32x32 пикселей, формат ICO
maximizeButton	Указывает, будет ли активирована кнопка «восстановить» в заголовке документа (yes/no)
minimizeButton	Указывает, будет ли активирована кнопка «свернуть» в заголовке документа (yes/no)
showInTaskbar	Указывает, будет ли показан документ в панели инструментов Windows (yes/no)
windowState	Первоначальный размер окна; среди возможных значений – normal, minimize, maximize
innerBorder	Указывает, будет ли окно иметь внутреннюю границу (yes/no)
navigable	Указывает, будут ли ссылки открываться в отдельных окнах или в одном (yes/no)
scroll	Указывает, будет ли окно иметь полосу прокрутки (yes/no)
scrollFlat	Указывает вид полосы прокрутки: 3D или нет (yes/no)
singleInstance	Определяет, возможно ли одновременное открытие нескольких окон для НТА-документов
sysMenu	Указывает, будут ли показаны составляющие заголовка окна (yes/no)
contextMenu	Указывает на доступность контекстного меню, вызываемого правой кнопкой мыши (yes/no)

Окончание таблицы 5

1	2
selection	Указывает , можно ли выделять текст в НТА-окне (yes/no)
version	Версия НТА

3 Сценарий для формирования и учета почтовых отправлений

В данной главе будет рассмотрено решение одной офисной задачи – автоматизация оформления и учета почтовых отправлений в организации.

3.1 Формулировка задачи

Необходимо вести реестр отправленных писем и бандеролей, учитывая дату, наименование и адрес получателя, вес посылки, цену за отправку. Для каждой записи в реестре должен быть указан отдел-инициатор отправки.

Данная задача решалась на предприятии в ручном режиме: в каждом отделе заполнялся и распечатывался листов заказа на отправку корреспонденции, который отдавался вместе с посылкой в отдел управления делами. Там данные с листа заказа перебивали в xls-файл с реестром отправленной корреспонденции. При таком методе дублировалась работа по занесению в компьютер одной и той же информации, что увеличивало трудоемкость этой офисной задачи. При этом в силу человеческого фактора могли возникнуть какие-либо ошибки, которые нужны было исправлять, что в свою очередь увеличивало время обработки исходящей корреспонденции.

Для решения поставленной офисной задачи был написан сценарий WSH на языке JScript в формате НТА, который позволил автоматизировать процесс

формирования листов заказа на отправку и занесения соответствующей информации в единый xls-файл с реестром отправленных сообщений.

3.2 Структура приложения

Разработанное приложение состоит из нескольких файлов:

- 1) Главное HTA-приложение ПОЧТА.hta, содержащее основные функции на языке JScript и HTML-код для пользовательского интерфейса. Исходный текст приложения приведен в Приложении Б (листинг 6).
- 2) Excel-файл 1.xlsm с встроенным макросом на языке VBA для сохранения реестра отправлений и построения нужных отчетных форм.
- 3) Word-файл 1.doc, использующийся для распечатки на принтере листа заказа на отправку документов.

Алгоритм работы и графический интерфейс разработанной утилиты приведены в Приложении А.

ЗАКЛЮЧЕНИЕ

В деятельности организации любой формы собственности и любого вида деятельности постоянно приходится сталкиваться со стандартными офисными задачами. Эффективность их решения во многом зависит от выбранного офисного пакета и используемых технологий автоматизации рутинных работ.

В настоящей работе были рассмотрены наиболее распространенные офисные пакеты (как бесплатные, так и коммерческие), изучены возможности использования объектных моделей Microsoft Office во внешних приложениях и сценариях. В результате мы можем сделать вывод, что в качестве удобного и достаточно универсального инструмента для самостоятельного решения небольших задач, связанных с автоматизацией офисной деятельности, можно использовать сценарии Windows Script Host в совокупности с макропрограммами, встроенными в офисные документы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 CIT Forum [Электронный ресурс]. – [Б. м. : б. и.], [2010]. – Режим доступа : <http://citforum.ru/internet/html/hta/>
- 2 LibreOffice [Электронный ресурс]. – [Б. м. : б. и.], [2012]. – Режим доступа : <http://ru.libreoffice.org/home/>
- 3 WordPerfect [Электронный ресурс]. – [Б. м. : б. и.], [2016]. – Режим доступа : <http://www.wordperfect.com/rw/product/corel-office-software/?pid=prod3430104>
- 4 Википедия [Электронный ресурс]. – [Б. м. : б. и.], [2012]. – Режим доступа : https://ru.wikipedia.org/wiki/Microsoft_Office
- 5 Макарова Н. В. Информатика : учеб. пособие для вузов / Н. В. Макарова. – СПб. : Питер, 2011. – 576 с.
- 6 Общие сведения об объектной модели Word [Электронный ресурс]. – [Б. м. : б. и.], [2014]. – Режим доступа : <https://msdn.microsoft.com/ru-ru/library/kw65a0we.aspx>
- 7 Попов А. В. Windows Script Host для Windows 2000/XP: / А. В. Попов. – СПб. : БХВ–Петербург, 2003. – 640 с.
- 8 Программное обеспечение компьютеров [Электронный ресурс]. – [Б. м. : б. и.], [2010]. – Режим доступа : http://book.kbsu.ru/theory/chapter6/1_6_20.html
- 9 Специальные курсы по компьютерным технологиям [Электронный ресурс]. – [Б. м. : б. и.], [2008]. – Режим доступа : http://www.askit.ru/custom/vba_office/m1/01_02_basis_vba.htm
- 10 Уокенбах Д. Профессиональное программирование на VBA в Excel 2003:/ Д. Уокенбах. – СПб. : Диалектика, 2005. – 700 с.
- 11 Учебно-методические материалы для студентов кафедры АСОИУ [Электронный ресурс]. – [Б. м. : б. и.], [2013]. – Режим доступа : <http://www.4stud.info/ppp/lecture10.html>

12 Учебный материал ВГУЭС [Электронный ресурс]. – [Б. м. : б. и.],
[2000]. – Режим доступа :
http://abc.vvsu.ru/books/up_inform_tehnol_v_ekon/page0014.asp

ПРИЛОЖЕНИЕ А

(обязательное)

Пользовательский интерфейс

Графический интерфейс приложения ПОЧТА.hta представлен в соответствии с рисунком А.1.

Рисунок А.1 – Приложение ПОЧТА.hta

Данная форма ввода имеет несколько полей:

1. Управление/Отдел/Сектор. Пользователь выбирает из выпадающего списка нужное структурное подразделение (рисунок А.2).

Рисунок А.2 – Выпадающий список поля
Управление/Отдел/Сектор

Продолжение ПРИЛОЖЕНИЯ А

2. Поле для ввода фамилии и инициалов. При некорректном вводе информации выходит предупреждение (рисунок А.3).

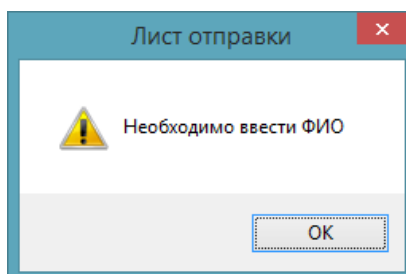


Рисунок А.3 – Предупреждение о некорректном вводе фамилии

3. Поле для выбора вида отправки (рисунок А.4).

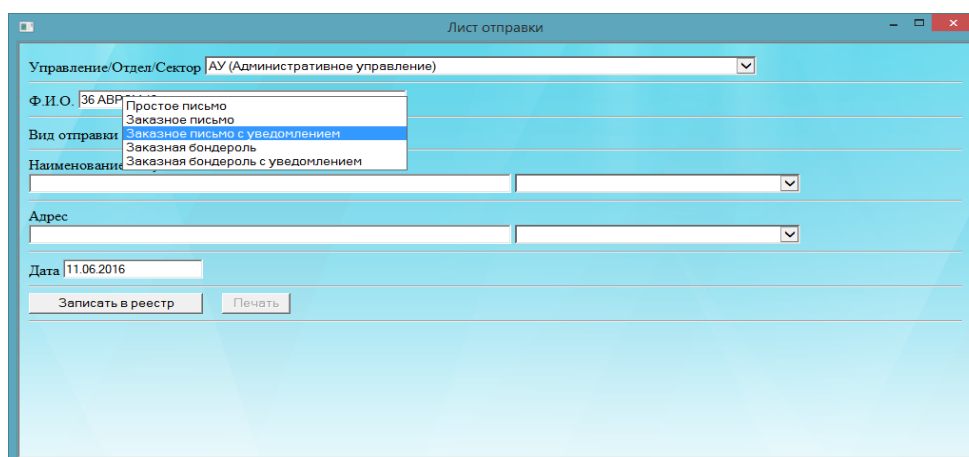


Рисунок А.4 – Выпадающий список доступных видов отправки

4. Наименование получателя. При некорректном вводе наименования получателя выходит предупреждение (рисунок А.5).

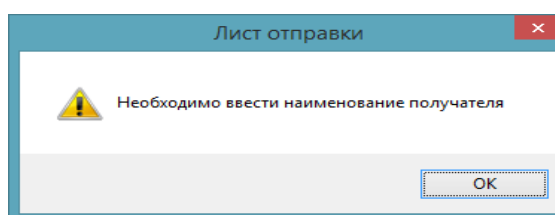


Рисунок А.5 – Предупреждение о некорректном вводе данных

Продолжение ПРИЛОЖЕНИЯ А

Если данному получателю ранее уже отправлялась корреспонденция, то его наименование можно выбрать из выпадающего списка (рисунок А.6).

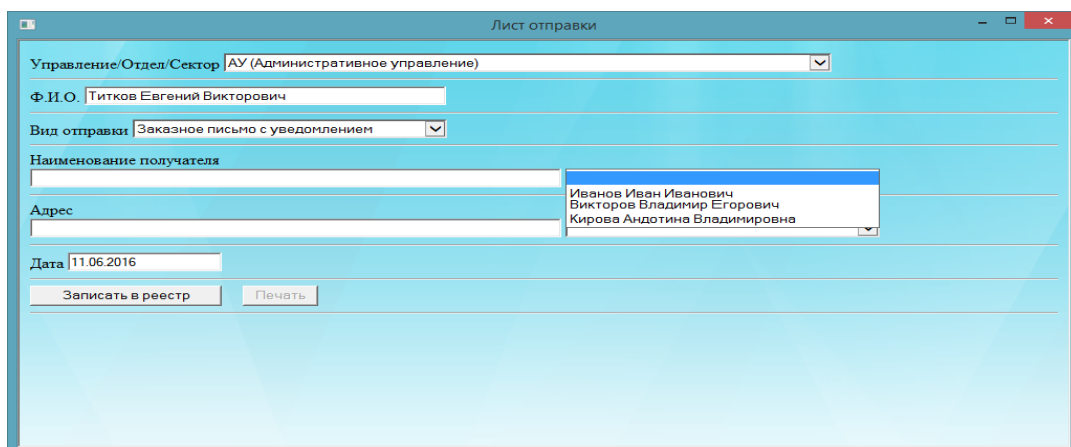


Рисунок А.6 – Выпадающий список получателей

5. Адрес получателя. При некорректном вводе выходит предупреждение (рисунок А.7).

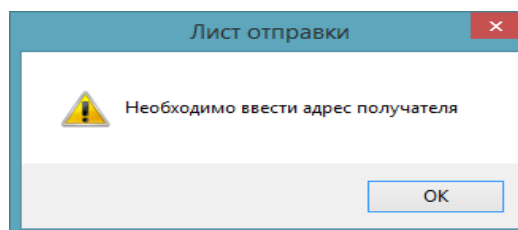


Рисунок А.7 – Предупреждение о необходимости ввода адреса

Если данному получателю ранее уже отправлялась корреспонденция, то его адрес можно выбрать из выпадающего списка.

6. Дата. При активации этого поля на форме появляется календарь, где можно выбрать нужную дату.

После заполнения всех полей формы в соответствии с рисунком А.8 можно сохранить данные в реестр (кнопка "Запись в реестр").

Продолжение ПРИЛОЖЕНИЯ А

Если все данные заполнены корректно, то выходит сообщение, показанное в соответствии с рисунком А.9.

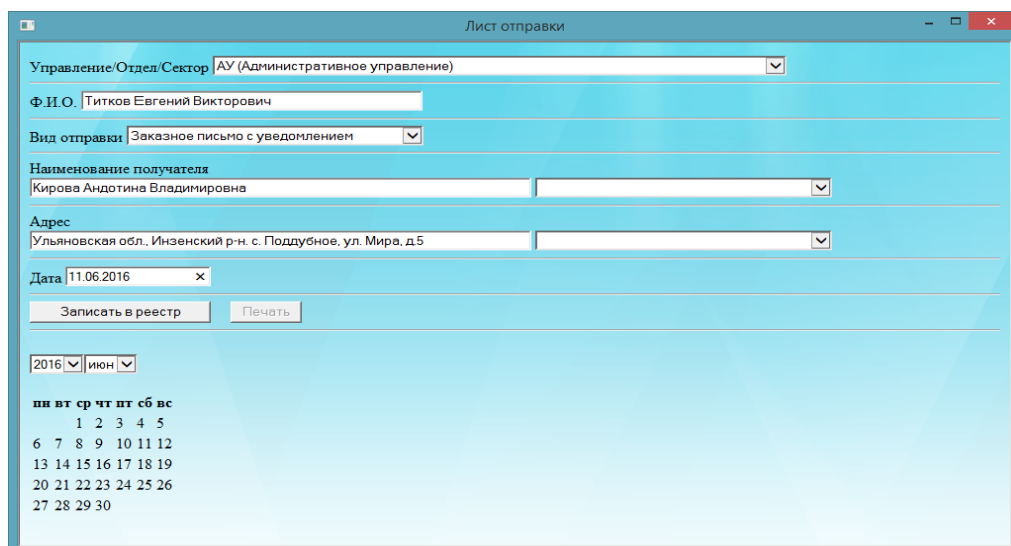


Рисунок А.8 – Правильно заполненное приложение.

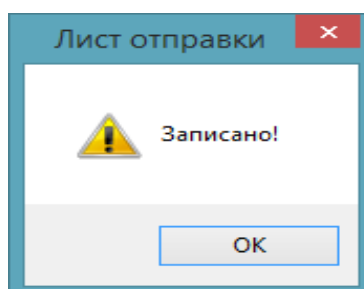


Рисунок А.9 – Уведомление о записи в реестр

Кнопка Печать позволяет распечатать лист заказа на отправку документов (рисунок А.10), который откроется в программе Microsoft Word .

Продолжение ПРИЛОЖЕНИЯ А

ЛИСТ ЗАКАЗА
на отправку документов

ОТПРАВИТЕЛЬ

Управление/отдел/сектор: _____ Контактный телефон: _____
 ФИО отправителя: _____ Подпись: _____
 Дата время: _____
 Документ: вид Письмо _____ Исх.№ _____ от "___" ____ 201_г.

Вид отправки						
Простое письмо	Заказное письмо	Заказное письмо с уведомлением	Факс	Телекс	Интернет	Нарочно
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ПОЛУЧАТЕЛЬ

(почтовый адрес/ код города и номер факса/ адрес электронной почты)

ХОД ОТПРАВКИ

ОТПРАВЛЕНО (Дата, время, подпись): _____

Рисунок А.10 – Лист заказа на отправку документов

При записи данных в реестр они сохраняются в Excel-файле 1.xlsm (рисунок А.11).

1	Дата	Наименование, Адрес	Вес	Цена	Вид отправки	Отдел
2	10.07.15	ОАО "Рухимаш" 431446, Мордовия, Рузаевский р-он, г. Рузаевка-б		15	Простое письмо	АУ (Административное управление)
3	11.06.16	Кирова Антонина Владимировна г.Саранск, ул. Московская, д.78	36	150	Заказная бондероль с уведомлением	АУ (Административное управление)
4	11.06.16	Поддубное, ул. Мира, д.5		60	Заказное письмо с уведомлением	АУ (Административное управление)
5	12.06.16	Вдовин Денис Евгеньевич г.Ульяновск, ул.Мира,32-15	13	100,00	Заказная бондероль	АУ (Административное управление)
6	12.06.16	Павлов Сергей Петрович г.Саранск, просп. Ленина, 5-32		60	Заказное письмо	АУ (Административное управление)
7	12.06.16	Петров Антон Владимирович г.Самара, ул. Авроры, 14-8		15	Простое письмо	ОВА (Отдел внутреннего аудита)
8	12.06.16	Петров Антон Владимирович г.Самара, ул. Авроры, 14-8		80	Заказная бондероль с уведомлением	ОТП (Отдел технической поддержки)
9	12.06.16	Кабанов Александр Александрович г.Самара, ул. Самарская, 34-78	10	70	Заказная бондероль	ОТП (Отдел технической поддержки)
10	12.06.16	Романов Петр Сергеевич г.Самара, ул. Товарная, 14-78	13	90	Заказная бондероль	ОТП (Отдел технической поддержки)
11	14.06.16	Вдовина Светлана Александровна г.Самара, ул. Товарная, 14-78	8	50	Заказная бондероль	ОТП (Отдел технической поддержки)

Рисунок А.11 – Записи об отправлениях в реестре

В данном реестре (рисунок А.12) мы видим информацию о всех отправленных посылках в хронологическом порядке.

Продолжение ПРИЛОЖЕНИЯ А

Для выделения посылок за текущий день нужно нажать на кнопку «В рабочий лист», при этом будет выполнен макрос, приведенный в листинге 7.

Дата	Наименование, Адрес	Вес	Цена	Вид отправки	Отдел
12.06.16	Вдовин Денис Евгеньевич г.Ульяновск, ул.Мира,32-15	13	100,00	Заказная бондероль	АУ (Административное управление)
12.06.16	Павлов Сергей Петрович г.Саранск, просп. Ленина, 5-32	60		Заказное письмо	АУ (Административное управление)
12.06.16	Петров Антон Владимирович г.Самара, ул. Авроры, 14-8	15		Простое письмо	ОВА (Отдел внутреннего аудита)
12.06.16	Петров Антон Владимирович г.Самара, ул. Авроры, 14-8	12	80	Заказная бондероль с уведомлением	ОТП (Отдел технической поддержки)
12.06.16	Кабанов Александр Александрович г.Самара, ул. Самарская, 34-78	10	70	Заказная бондероль	ОТП (Отдел технической поддержки)
12.06.16	Романов Петр Сергеевич г.Самара, ул. Товарная, 14-78	13	90	Заказная бондероль	ОТП (Отдел технической поддержки)

Рисунок А.12 – Рабочий лист с посылками за текущую дату

Для группировки информации по видам отправки необходимо нажать на кнопку «Распределить». В результате сформируются необходимые для отчетности листы со списками простых и заказных писем и бандеролей (рисунки А.13 – А.17).

СПИСОК № 1 _____ внутренних почтовых отправлений от 12.06.16

Вид и категория: ЗАКАЗНЫЕ ПИСЬМА

Отправитель: Мордовское отделение № 8589 Сбербанка РФ _____ ГСП_93_

Наименование и индекс места приема _____ 430033_

Всего ПИО _____ Всего листов _____

№ п/п	Куз., Кочу	Вес (гр)	Сумма платы за пересылку у (руб)	ШПИ
1	Петров Антон Владимирович г.Самара, ул. Авроры, 14-8		15,00	
			15,00	

Итого 1 ПРОСТОЕ ПИСЬМО 16,00 РУБ.
Служ. Старший инспектор Общего отдела Мордовского отделения № 8589 Сбербанка России
Полном. ФИО _____ Отчество КШ _____
-М.П. _____

Привлеч. оператор _____
Полном. ФИО _____
Отчество КШ _____

Рисунок А.13 – Список простых писем

Продолжение ПРИЛОЖЕНИЯ А

Итого 1 ЗАКАЗНОЕ ПИСЬМО 60,00 РУБ.

№ п/п	Куда, Кому	Вес (гр)	Сумма платы за пересылку (руб)	ШПН
1	Павлов Сергей Петрович г.Саранск, просп. Ленина, 5-32		60,00	
			60,00	

Итого 1 ЗАКАЗНОЕ ПИСЬМО 60,00 РУБ.
 Сдел Старший инспектор Общего отдела
 Мордовского отделения № 8589 Сбербанка России
 Подпись: _____ Полпись: _____
 ФИО: _____ ФИО: _____
 -М.П. _____ Оттиск КШП

Рисунок А.14 – Список заказных писем

Итого 0 ЗАКАЗНЫХ ПИСЕМ С УВЕДОМЛЕНИЕМ 0,00 РУБ.

№ п/п	Куда, Кому	Вес (гр)	Сумма платы за пересылку (руб)	ШПН
-------	------------	----------	--------------------------------	-----

Итого 0 ЗАКАЗНЫХ ПИСЕМ С УВЕДОМЛЕНИЕМ 0,00 РУБ.
 Сдел Старший инспектор Общего отдела
 Мордовского отделения № 8589 Сбербанка России
 Подпись: _____ Полпись: _____
 ФИО: _____ ФИО: _____
 -М.П. _____ Оттиск КШП

Рисунок А.15 – Список заказных писем с уведомлением

Окончание ПРИЛОЖЕНИЯ А

Итого 3 ЗАКАЗНЫХ БАНДЕРОЛИ 260,00 РУБ.

СПИСОК № 1

Вид и категория: ЗАКАЗНЫЕ БАНДЕРОЛИ
внутренних почтовых отправлений от 12.06.16

Отправитель: Мордовское отделение № 8589 Сбербанка РФ ГСП_93

Наименование и индекс места приема 430033

Всего РПО

№ п/п	Куда, Кому	Вес (гр)	Сумма платы за пересылку (руб)	ШПН
8	1 Вдовин Денис Евгеньевич г.Ульяновск, ул.Мира,32-15	13	100,00	
9	2 Кабанов Александр Александрович г.Самара, ул. Самарская, 34-78	10	70,00	
10	3 Романов Петр Сергеевич г.Самара, ул. Товарная, 14-78	13	90,00	
			260,00	

Итого 3 ЗАКАЗНЫХ БАНДЕРОЛИ 260,00 РУБ.
Служ Старший инспектор Общего отдела Мордовского отделения № 8589 Сбербанка России
Подпись: _____ ФИО: _____ -М.П. _____

Принял: оператор _____
Подпись: _____ ФИО: _____ Оттиск КШ

Рисунок А.16 – Список заказных бандеролей

Итого 1 ЗАКАЗНАЯ БАНДЕРОЛЬ С УВЕДОМЛЕНИЕМ 80,00 РУБ.

СПИСОК № 1

Вид и категория: ЗАКАЗНЫЕ БАНДЕРОЛИ С УВЕДОМЛЕНИЕМ
внутренних почтовых отправлений от 12.06.16

Отправитель: Мордовское отделение № 8589 Сбербанка РФ ГСП_93

Наименование и индекс места приема 430033

Всего РПО

№ п/п	Куда, Кому	Вес (гр)	Сумма платы за пересылку (руб)	ШПН
8	1 Петров Антон Владимирович г.Самара, ул. Авроры, 14-8	12	80,00	
			80,00	

Итого 1 ЗАКАЗНАЯ БАНДЕРОЛЬ С УВЕДОМЛЕНИЕМ 80,00 РУБ.
Служ Старший инспектор Общего отдела Мордовского отделения № 8589 Сбербанка России
Подпись: _____ ФИО: _____ -М.П. _____

Принял: оператор _____
Подпись: _____ ФИО: _____ Оттиск КШ

Рисунок А.17 – Список заказных бандеролей с уведомлением

ПРИЛОЖЕНИЕ Б

(обязательное)

Код программы

Листинг 6 Управляющее приложение ПОЧТА.hta

```
<html>
<title>Лист отправки</title>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
  <HTA:APPLICATION Border=Thick
    BorderStyle=Complex
    ShowInTaskBar=Yes
    ApplicationName="map"
    Scroll=No SingleInstance=No
    MaximizeButton=Yes
    MinimizeButton=Yes>

  <style>
    body {
      background-image: url("fon.png");
      background-attachment: fixed;
      background-position: center;
      background-repeat: no-repeat;
    }
  </style>
  <script type='text/javascript'>
  //создание таблицы под календарь
  if (!window.tempobj) window.tempobj = [];
  if (!window.dict) window.dict = {};
  dict.calendar = {
    div: '.',
    table: {
    }
  }
  </script>
```

Продолжение ПРИЛОЖЕНИЯ Б

```
};  
  
//функция для вывода календаря  
  
/*ниже функции написаны для правильной работы календаря*/  
  
-function() {  
    var d = document,  
        year_offset = 5,  
        red_day = 'red',  
        hover = 'hover',  
        lang = {  
            dn: ['нн', 'вт', 'ср', 'чт', 'пт', 'сб', 'вс'],  
            mm: ['янв', 'фев', 'мар', 'апр', 'май', 'июн', 'июл', 'авг', 'сен', 'окт', 'ноя', 'дек'],  
            correctDn: function(d) {  
                return (d == 0) ? 6 : d - 1  
            },  
            makeDate: function(s) {  
                s = s.split(/\D+/);  
                return new Date(s[2], s[1] - 1, s[0])  
            },  
            formatDate: function(d) {  
                d = [d.getDate(), d.getMonth() + 1, d.getFullYear()]  
                return d.join(dict.calendar.div || '.').replace(/b(\d)b/g, '0$1')  
            }  
        },  
        box = ce('div');  
        box.className = 'mm disnone';  
  
        var buildCArr = function(y) {  
            var d0 = new Date(y, 0, 1),  
                c = [],  
                dd, dn, mm;  
            for (var i = 0; i < 12; i++) {  
                c[i] = []  
            }  
            while (d0.getFullYear() == y) {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
    dd = d0.getDate()
    dn = lang.correctDn(d0.getDay())
    mm = d0.getMonth()
    c[mm][dd] = dn
    d0.setDate(dd + 1)
  }
  return c;
}

var setDate = function(e) {
  e = e || window.event
  var o = e.target || e.srcElement
  if(!o.tagName || o.tagName.toLowerCase() != 'div') return
  var d = o.firstChild.nodeValue;
  this.input.date = new Date(this.year, this.month, d)
  if(!this.input.getAttribute('disabled')) {
    this.input.value = lang.formatDate(this.input.date)
    cc(this, 'disnone')
    this.input.onblur = hideMonth
    this.input.onfocus = showMonth
  }}

var hideMonth = function(e, o) {
  e = e || window.event
  o = o || this
  o.onfocus = showMonth
  if(e && o.calendar && !hc(o.calendar.className, 'disnone'))
    setTimeout(function() {cc(o.calendar, 'disnone')}, 400)
}

var getEvent = function(d) {
  d = d.replace(/b(\d)b/g, '0$1')
  return dict.calendar.table[d]
}

var showMonth = function(e, dd, el) {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
e = e || window.event
var oInput = e && (e.target || e.srcElement)
dd = dd || this.date || new Date()
var o = el || this;
o.onfocus = "";

if(o.timer) {
    clearTimeout(o.timer);
    o.timer = null
}
if (e && e.type == 'focus') return (o.timer = setTimeout(function() {showMonth(null, dd, o)},
400))

if (e && e.type == 'click' && oInput == o && o.calendar && !hc(o.calendar.className,
'disnone'))
    return cc(o.calendar, 'disnone')

var Y = dd.getFullYear(),
c = buildCArr(Y),
m = dd.getMonth(),
arr = c[m],
cN, curr_day = (o.date) ? dd.getDate() : 0,
marr = [],
htm = '<table><col><col><col><col><col><col class="red"><col class="red">',
week = new Array(7),
coo = getTopLeft(o),
flush = function(arr) {
    marr.push('<td>' + arr.join('</td><td>') + '</td>');
    return new Array(7)
};
if (!o.calendar) {
    o.calendar = ac(box.cloneNode(true))
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
var c_div = o.calendar,
    title;

htm += '<tr><th>' + lang.dn.join('</th><th>') + '</th></tr>'

for (var i = 1, l = arr.length; i < l; i++) {
    cN = []
    title = getEvent(Y + '-' + (m + 1) + '-' + i) || ''
    if (title) title = ' title="' + title + '"'
    if (arr[i] > 4) cN.push(red_day)
    if (i == curr_day) cN.push(hover)
    if (title) cN.push('metka')
    cN = (cN.length) ? ' class="' + cN.join(' ') + '"' : ''
    week[arr[i]] = '<div' + cN + title + '>' + i + '</div>'
    if (arr[i] === 6) week = flush(week)
}
flush(week)

htm += '<tr>' + marr.join('</tr><tr>') + '</tr></table>';
c_div.innerHTML = htm

var p = ce('p'),
    months = buildField('select', {}, lang.mm),
    years = {},
    yl = year_offset * 2 + 1,
    y0 = Y - year_offset;

for (var i = 0; i < yl; i++) {
    years[y0] = y0++
}
years = buildField('select', {}, years)
ac(years, p)
years.selectedIndex = year_offset
years.onchange = function() {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
showMonth(null, new Date(this.value, m, 1), o)
}

ac(months, p)
cc(p, 'ri')

c_div.insertBefore(p, c_div.firstChild)
months.selectedIndex = m
months.onchange = function() {
    showMonth(null, new Date(Y, this.value, 1), o)
}

if (hc(c_div.className, 'disnone')) {
    cc(c_div, null, 'disnone')
    tempobj.push(o.calendar)

    var pageY = e && e.clientY || 0,
        c_height = c_div.offsetHeight + 10,
        th_height = o.offsetHeight + 10,
        c_top = (pageY > c_height) ? coo.top - c_height : coo.top + th_height;
    o.calendar.style.left = coo.left + 'px'
    o.calendar.style.top = c_top + 'px'
}

c_div.input = o
c_div.month = m
c_div.year = Y
c_div.onclick = setDate
c_div.onmouseover = function() {o.onblur = "";}
c_div.onmouseout = function() {o.onblur = hideMonth; }
o.focus()}
var buildField = function(el, params, idxs) {
    var el = ce(el),
        name = params.name;
```

Продолжение ПРИЛОЖЕНИЯ Б

```
for (var id in params) el[id] = params[id]
if (idxs) {
  var params;
  for (id in idxs) {
    params = {
      value: id
    }
    ac(buildEl2('option', params, idxs[id]), el)
  }
}
return el
}

var buildEl2 = function(el, params, txt) {
  /** элемент с вложенным textNode */
  var el = ce(el);
  for (var id in params) el[id] = params[id]
  if (txt != undefined) ac(ct(txt), el)
  return el
}

function initCalendar(e, el) {
  d.onkeyup = cancel
  var inputs = (el) ? [el] : gt("input"),
  inp;
  for (var e in inputs) {
    inp = inputs[e]
    if (!hc(inp && inp.className, "date")) continue;
    inp.onclick = showMonth
    inp.onfocus = showMonth
    inp.onblur = hideMonth
  }
  addLoadEvent(initCalendar)
}()
```

Продолжение ПРИЛОЖЕНИЯ Б

```
//получение DOM-элемента
function gi(i) {
  return document.getElementById(i)
}

//создание DOM-элемента с тегом «t»
function ce(t) {
  return document.createElement(t)
}

//создание текстового узла, который можно вставить в DOM-элемент
function ct(t) {
  return document.createTextNode(t)
}

/*получение массива всех DOM-элементов с тегом «t», которые находятся в
элементе «e» (если "e" не задан, то поиск идёт по всему DOM-дереву, т.е. в глобальной
переменной "document")*/
function gt(t, e) {
  e = e || document;
  return e.getElementsByTagName(t)
}

/*добавление дочернего элемента "n" в элемент "e" (если "e" не задан, элемент
добавляется в конец элемента body (т.е. перед закрывающимся тегом </body>))*/
function ac(n, e) {
  e = e || document.body;
  return e.appendChild(n);
}

/*возвращение позиции слова "c" в строке "s" (знак тильды "~" в этой функции
означает "побитовое НЕ". простыми словами, например, для числа "0" будет "~0 = -1", для
числа "1" - "~1 = -2", "~5 = -6", и т.д.)*/
function hc(s, c) {
  return ~(' ' + s + ' ').indexOf(' ' + c + ' ')
}

// добавление или удаление класса у DOM-элемента "o"
function cc(o, add, del) { /*changeClass*/
  var o = o || {},
```

Продолжение ПРИЛОЖЕНИЯ Б

```
n = 'className',
cN = (undefined != o[n]) ? o[n] : o,
ok = 0
if ('string' !== typeof cN) return false
var re = new RegExp('(\\s+|^)' + del + '(\\s+/$)', 'g')
if (add) /*addClass*/
if (!hc(cN, add)) {
    cN += ' ' + add;
    ok++
}
if (del) /*delClass*/
if (hc(cN, del)) {
    cN = cN.replace(re, ' ');
    ok++
}
if (!ok) return false
if ('object' == typeof o) o[n] = cN
else return cN
}
```

//возвращение координат ВЕРХНЕЙ ЛЕВОЙ позиции у DOM-элемента

```
function getTopLeft(el) {
    var top = 0,
        left = 0;
    while (el) {
        top = top + parseInt(el.offsetTop)
        left = left + parseInt(el.offsetLeft)
        el = el.offsetParent
    }
    return {
        top: top,
        left: left
    }
}
```

Продолжение ПРИЛОЖЕНИЯ Б

/ отслеживание нажатия на клавиатуре клавиши "Escape" (27 - это её код, а переменная "e" - это объект DOM-события). И если эта клавиша нажата - запускается функция hidTemp() */*

```
function cancel(e) {  
    e = e || window.event  
    if (27 == e.keyCode) {  
        hidTemp()  
    }  
}
```

*/*прячем всплывающее окно (которое хранится в глобальной переменной window.tempobj) посредством добавления к нему CSS-класса "disnone" */*

```
function hidTemp() {  
    tempobj = window.tempobj  
    if (!tempobj) return  
    var el, t = (tempobj.length) ? tempobj : [tempobj]  
    for (var i = 0; i < t.length; i++) {  
        if (t[i]) cc(t[i], 'disnone')  
    }  
}
```

/ добавление функции "func", которая будет запускаться в момент загрузки нашего .hta-файла (в момент открытия .hta-файла и загрузки его компонентов, запускается функция "window.onload", в которую и "помещаем" нашу функцию "func")*/*

```
function addLoadEvent(func) {  
    var old = window.onload  
    if (typeof window.onload != 'function') window.onload = func  
    else window.onload = function() {  
        old();  
        func();  
    }  
}
```

//функция получает путь к текущей папке, откуда мы запустили наш .hta-файл

```
function getPath() {  
    var relpath = window.location.pathname.split("\\");
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        relpath.pop();
        relpath = relpath.join('\\');
        return relpath;
    }

</script>

<script language="javascript">
    var ForReading = 1;
    var ForWriting = 2;
    var ForAppending = 8;
    var TristateUseDefault = -2;
    var TristateTrue = -1;
    var TristateFalse = 0;
    var wshShell = new ActiveXObject("WScript.Shell");
    var wshEnv = wshShell.Environment("PROCESS");
    /*в переменной path1 хранится путь к папке AppData, в которой хранятся данные
    приложений для текущего пользователя Windows. Данный путь мы получаем при
    помощи заранее определенного объекта ActiveX - WScript.Shell */
    var path1 = (wshEnv("LOCALAPPDATA"));
    var path = getPath(); // общий путь
    //Функция удаляет из начала и конца строки пробелы
    String.prototype.trim = function() {
        return this.replace(/^\s+|\s+$/g, "");
    };

    function change() // объекты на форме скрыть/показать
    {
        var m = new Array("name", "np", "dat", "vid", "otd", "id_zr", "id_p", "adr", "adres",
" name_pp")
        for (var i = 0; i <= m.length - 1; i++) {

            if (document.getElementById(m[i]).disabled == false) {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
document.getElementById(m[i]).disabled = true;
} else document.getElementById(m[i]).disabled = false;}
}

function file(p) // Проверяем существование папки и файла, если нет, то создаем
{

var fso = new ActiveXObject("Scripting.FileSystemObject");
if (!fso.FolderExists(path1 + "\\expedition")) {
fso.CreateFolder(path1 + "\\expedition");
if (!fso.FileExists(path1 + "\\expedition\\" + p)) {
fso.createtextfile(path1 + "\\expedition\\" + p);
}
} else {
if (!fso.FileExists(path1 + "\\expedition\\" + p)) {
fso.createtextfile(path1 + "\\expedition\\" + p);
}}}

function p(p1, j, k) // для формирования списка из файла
{
var fso = new ActiveXObject("Scripting.FileSystemObject");
var g = fso.getFile(path1 + "\\expedition\\" + p1);
var flag = 0;
var f = fso.OpenTextfile(path1 + "\\expedition\\" + p1, 1);
var i = 0;
var adr = document.forms[0].elements[k].value;

while (!f.AtEndOfStream) {
var st = f.Readline();
var ob = document.getElementById(j); // обращение p("adress.txt", "adres", "adr")
ob.options[i] = new Option(st, i);
i = i + 1;
if (adr == st) {flag = 1 }
}
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
if (flag == 0) {
    var f = fso.OpenTextFile(path1 + "\\expedition\\" + p1, 8);
    f.WriteLine(adr);
    f.Close();
    }}
function z(j, k) // передает выбранное значение в input
{
    var ob = document.getElementById(j);
    if (ob.selectedIndex != -1) {
        document.forms[0].elements[k].value = ob.options[ob.selectedIndex].text
    }
}

function zapis(p, d, a, vid, otd) // запись в файл Excel
{
    // создаем экземпляр ActiveX-объекта Excel.Application
    var ExcelApp = new ActiveXObject("Excel.Application");
    var ExcelSheet = ExcelApp.Workbooks.Open(path + "\\1.xlsm"); // открываем файл
    "1.xlsm" из текущей папки

    var str = ExcelSheet.Application.Caption;
    // проверяем, есть ли вхождение строки '[Только для чтения]' в
заголовке
    if (str.indexOf('[Только для чтения]') != -1) {
        alert("Файл открыт другим пользователем, попробуйте позже");
        ExcelSheet.Application.Quit();//выход из приложения
    }else {
        var i = 1;
        /* проходим цикл while до тех пор, пока на первом рабочем листе (Sheets(1)) в
        первом столбце в i-ой строке (Cells(i, 1)) значение не будет определено*/
        while (ExcelSheet.Sheets(1).Cells(i, 1).Value != undefined) {
            i++;
        }
    }
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
    // разбиение строки "d" на массив "s" по разделителю "."
    var s = d.split(".");
    ExcelApp.Application.DisplayAlerts = false;
    ExcelSheet.Sheets(1).Cells(i, 2).Value = p + " " + a;
    ExcelSheet.Sheets(1).Cells(i, 1).FormulaR1C1 = parseInt(s[1], 10) + "/" + parseInt(s[0],
10) + "/" + parseInt(s[2], 10);
    ExcelSheet.Sheets(1).Cells(i, 5).Value = vid;
    ExcelSheet.Sheets(1).Cells(i, 6).Value = otd;
    ExcelSheet.Sheets(1).Activate;
    ExcelSheet.SaveAs(path + "\\1.xlsm");
    ExcelApp.Application.DisplayAlerts = true; //включение отображение ошибок
    ExcelSheet.Application.Quit();
    alert("Записано!");
    document.getElementById("id_p").disabled = false;
}
}

function reestr() // нажатие на кнопку "записать в реестр"
{
    p("adress.txt", "adres", "adr");
    p("name_p.txt", "name_pp", "np");
    if (index > 8) {
        alert(index);
        change();
    }
    // Считываем в переменную n ФИО
    var n = theForm.name1.value;
    // Считываем в переменную dn дату
    var dn = theForm.cal.value;
    //+++++

    var testd = ^d{1,2}\\.d{1,2}\\.d{4}/;
    if (!testd.test(dn)) {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
alert("Неверный формат даты! Воспользуйтесь календарем");  
return;  
}
```

```
var z1 = st.split('.');  
var d1 = new Date(z1[0], z1[1] - 1, z1[2]);  
var z2 = dn.split('.');  
var d2 = new Date(z2[0], z2[1] - 1, z2[2]);
```

```
switch (z2[1]) {  
case "01": // Январь  
case "03": // Март  
case "05": // Май  
case "07": // Июль  
case "08": // Август  
case "10": // Октябрь  
case "12": // Декабрь  
if ((parseInt(z2[0]) >= 0) && (parseInt(z2[0]) <= 31)) {break; }  
case "04": // Апрель  
case "06": // Июнь  
case "09": // Сентябрь  
case "11": // Ноябрь  
if ((parseInt(z2[0]) >= 0) && (parseInt(z2[0]) <= 30)) {break; }  
case "02": // Декабрь  
if ((parseInt(z2[0]) >= 0) && (parseInt(z2[0]) <= 29)) {break; }  
default:  
alert("Не корректная дата! Воспользуйтесь календарем");  
return;  
}
```

```
if (d2 < d1) {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
alert("Введенная Вами дата уже прошла");
theForm.cal.value = "";
return;
}
//+++++

// Считываем в переменную пр наименование получателя

var pr = theForm.name_p.value;
var adr = theForm.adr.value;
    // Проверка корректности ввода ФИО
if (n.trim() == "") {      alert("Необходимо ввести ФИО");
    return;
}
pr = pr.trim();
// Проверка корректности ввода наименования
if (np == "") {
    alert("Необходимо ввести наименование получателя");
return; }
    // Проверка корректности ввода
if (adr.trim() == "") {
    alert("Необходимо ввести адрес получателя");
return;
}
// Проверяем правильность ввода даты
    // Проверка непутого ввода даты
if (dn.trim() == "") {
    alert("Необходимо ввести дату");
return;
}
var vv = document.getElementById("vid");
var v = vv.options[vv.selectedIndex].text;

var oo = document.getElementById("otd");
```

Продолжение ПРИЛОЖЕНИЯ Б

```
var o = oo.options[oo.selectedIndex].text;

zapis(np, dn, adr, v, o);

stroka2 = stroka2 + index + ". Адрес: " + adr + ";" + '\n' + "Наименование: " + np +
'\n\n';

index = index + 1; }
// записать данных в файл и напечатать лист учета
function печат() {
var n = theForm.name1.value;
var dn = theForm.cal.value;
var np = theForm.name_p.value;
var adr = theForm.adr.value;
var vv = document.getElementById("vid");
var v = vv.options[vv.selectedIndex].text;
var ot = document.getElementById("otd");
var o = ot.options[ot.selectedIndex].text;
var word = new ActiveXObject("Word.Application");
    // общий путь образцу листа
var c = word.Documents.Open(path + "\\1.doc");
var fm = c.Bookmarks("fio").Range.Start;
c.Range(fm, fm + n.length).Text = (n);
var dt = c.Bookmarks("date").Range.Start;
c.Range(dt, dt + dn.length).Text = (dn);
var ap = c.Bookmarks("adres").Range.Start;

c.Range(ap, ap + adr.length).Text = (stroka2);

var ol = c.Bookmarks("otdel").Range.Start;
c.Range(ol, ol + o.length).Text = (o);
word.ActiveDocument.SaveAs(path1 + "\\expedition\\" + n + ".doc"); // путь локальный
```

Продолжение ПРИЛОЖЕНИЯ Б

```
c.Close();
alert("Записано");
var d = word.Documents.Open(path1 + "\\expedition\\" + n + ".doc"); // проверить
d.Application.Visible = false;

d.printOut(Background = true);

stroka2 = "";
index = 1;

d.Close();
//d.Application.Quit(); //диспетчер задач...
change();
}
</script>
</head>

<body BGCOLOR="silver" l="no">
<form name="theForm">
<span>Управление/Отдел/Сектор</span>

<select id="otd" style="width : 580">
</select>
<hr/>
<span>Ф.И.О.</span> <input type="text" size="53" id="name" name="name1" Value=""
/><br/>
<hr/>
<span>Вид отправки</span>
<select id="vid" style="width : 300">
<option value="1">Простое письмо</option>
<option value="2">Заказное письмо</option>
<option value="3" selected>Заказное письмо с уведомлением</option>
<option value="4" >Заказная бондероль</option>
```


Продолжение ПРИЛОЖЕНИЯ Б

```
var da = new Date();
var day = "";
var mon = "";
if (da.getDate() < 10) day = "0" + da.getDate().toString()
else day = da.getDate();
if ((da.getMonth() + 1) < 10) mon = "0" + (da.getMonth() + 1).toString()
else mon = (da.getMonth() + 1);
var st = day + "." + mon + "." + da.getFullYear();
theForm.dat.value = st;
```

// Определяем массив с "Управлениями/Отделами/Секторами"

```
var as = new Array("ОППО (Отдел планирования и отчетности)", "УК (Управление
кредитования)", "ЦУНДО (Центр управления наличным денежным обращением)", "ФО
(Финансовый отдел)", "ОВА (Отдел внутреннего аудита)", "ОПМБ (Отдел продаж малому
бизнесу)", "ОРП (Отдел по работе с персоналом)", "ЮО (Юридический отдел)", "ОСИМКО
(Отдел сопровождения и мониторинга кредитных операций)", "ГСМИК (Группа
специалистов маркетинга и коммуникаций)", "ОТБ (Отдел транзакционного бизнеса)",
"ОРИСКБ (Отдел развития и сопровождения корпоративного бизнеса)", "ОРЗ (Отдел по
работе с залогами)", "ГСПРПСС (Группа специалистов по реализации ПСС)", "Отдел «Банк
ХХI»", "ОПРСП (Отдел по работе с партнерами)", "ОБ (Отдел безопасности)", "МП
(Менеджер проектов)", "ОПРСПАЮЛ (Отдел по работе с проблемными активами
юридических лиц)", "ГСУФС (Группа специалистов управления филиальной сетью)",
"ОРПЗФЛ (Отдел по работе с проблемной задолженностью физических лиц)", "УПКК
(Управление продаж корпоративным клиентам)", "ГСПРНФР (Группа специалистов по
работе на финансовых рынках)", "ОПУД (Отдел по управлению делами)", "АУ
(Административное управление)", "ОПИОВСВСП (Отдел продаж и обслуживания в сети
ВСП)", "ОТП (Отдел технической поддержки)", "ОККО (Отдел качества клиентского
обслуживания)", "ОПП (Отдел прямых продаж)", "ООИПППБ (Отдел организации и
поддержки продаж продуктов благосостояния)", "ЦОПП (Центр оперативной поддержки
продаж)", "ДО/099", "Депозитарий");
```

// Все элементы массива "as" вставляем в DOM-элемент "select" с идентификатором "otd"

```
var as1 = as.join("%&^").split("%&^");
var as2 = as1.sort();
```

Продолжение ПРИЛОЖЕНИЯ Б

```
for (var i = 0; i < as2.length; i++) {  
    var obj = document.getElementById("otd");  
    obj.options[i] = new Option(as2[i], i);  
}  
/* открываем (или создаем, если их нет) файлы adress.txt и name_p.txt и считываем из них  
данные для вставки в форму где открывающие списки Наименования получателя и Адрес*/  
file("adress.txt");  
var adr = theForm.adr.value;  
p("adress.txt", "adres", "adr");  
file("name_p.txt");  
var np = theForm.name_p.value;  
p("name_p.txt", "name_pp", "np");  
  
var stroka2 = "";  
var index = 1;  
</script>  
</html>
```

Листинг 7. Распределение данных в реестре и на рабочий лист по дате

```
Sub Кнопка1_Щелчок() 'сформировать реестр  
    Sheets(2).Select 'очистить рабочий лист и фильтр на сегодняшнюю дату  
    Columns("A:F").Select  
    Selection.ClearContents  
    Sheets(1).Select  
    Columns("A:A").Select  
    Selection.AutoFilter  
    ActiveSheet.Range("A:A").AutoFilter Field:=1, Criteria1:= _  
        xlFilterToday, Operator:=xlFilterDynamic  
  
    Columns("A:F").Select 'копируем и вставляем на рабочий лист записи  
    Selection.Copy  
    Sheets(2).Select
```

Продолжение ПРИЛОЖЕНИЯ Б

```
Range("A1").Select
ActiveSheet.Paste
Sheets(1).Select
Range("F14").Select
Application.CutCopyMode = False
Columns("A:A").Select
Selection.AutoFilter
Sheets(2).Select
Range("D2").Select
Columns("A:F").EntireColumn.AutoFit

i = 2
Do While Sheets(2).Cells(i, 1).Value <> ""
i = i + 1
Loop
Range(Cells(2, 5), Cells(i - 1, 5)).Select 'чтобы был выпадающий список, и не допускалось
свободного ввода чтоб не нарушить условие
With Selection.Validation
.Delete
.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
xlBetween, Formula1:= _
"Простое письмо,Заказное письмо,Заказная бондероль,Заказное письмо с
уведомлением,Заказная бондероль с уведомлением"
.IgnoreBlank = True
.InCellDropdown = True
.InputTitle = ""
.ErrorTitle = ""
.InputMessage = ""
.ErrorMessage = ""
.ShowInput = True
.ShowError = True
End With
Range(Cells(1, 1), Cells(i - 1, 6)).Select 'граница ячеек
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
```

Продолжение ПРИЛОЖЕНИЯ Б

Selection.Borders(xlDiagonalUp).LineStyle = xlNone

With Selection.Borders(xlEdgeLeft)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlEdgeTop)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlEdgeBottom)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlEdgeRight)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlInsideVertical)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlInsideHorizontal)

.LineStyle = xlContinuous

.ColorIndex = 0

Продолжение ПРИЛОЖЕНИЯ Б

.TintAndShade = 0

.Weight = xlThin

End With

End Sub

Листинг 8. Распределение по виду отправки

Function ArrayOfValues(ByVal txt\$) As Variant

arr = Split(Replace(txt\$, " ", ""), ","): Dim n As Long: ReDim tmpArr(0 To 0)

For i = LBound(arr) To UBound(arr)

Select Case True

Case arr(i) = "", Val(arr(i)) < 0

Case IsNumeric(arr(i))

tmpArr(UBound(tmpArr)) = arr(i): ReDim Preserve tmpArr(0 To UBound(tmpArr) + 1)

Case arr(i) Like "#:#*"*

spl = Split(arr(i), ":")

If UBound(spl) = 1 Then

If IsNumeric(spl(0)) And IsNumeric(spl(1)) Then

For j = Val(spl(0)) To Val(spl(1)) Step IIf(Val(spl(0)) > Val(spl(1)), -1, 1)

tmpArr(UBound(tmpArr)) = j: ReDim Preserve tmpArr(0 To UBound(tmpArr) + 1)

Next j

End If

End If

End Select

Next i

On Error Resume Next: ReDim Preserve tmpArr(0 To UBound(tmpArr) - 1)

ArrayOfValues = tmpArr

End Function

'-----

Sub raspredelenie(l As Variant, m As String, c As Variant)

Sheets(2).Activate

ActiveSheet.Range(Cells(1, 2), Cells(1, 4)).Select

Selection.Copy

Sheets(2).Paste Destination:=Sheets(m).Range(Sheets(m).Cells(8 + c, 2), Sheets(m).Cells(8 + c, 4))

Sheets(m).Cells(8 + c, 1).Value = c + 1

Продолжение ПРИЛОЖЕНИЯ Б

Sheets(m).Activate

Sheets(m).Cells(8 + c, 1).Select

With Selection.Font

.Name = "Arial Cyr"

.Size = 10

End With

Selection.Font.Bold = True

Sheets(m).Cells(8 + c + 1, 4).FormulaR1C1 = "=SUM(R[" & CStr(-c - 1) & "]C:R[-1]C)"

Sheets(m).Cells(8 + c + 1, 4).NumberFormat = "0.00"

End Sub

'-----

Sub obedinenie(k As String) 'вставлять в каждый лист текущую дату объединение ячеек

Sheets(2).Activate

Sheets(2).Cells(2, 1).Select

Selection.Copy

Sheets(2).Paste Destination:=Sheets(k).Cells(2, 3)

Sheets(k).Activate

ActiveWorkbook.Sheets(k).Range("C2:E2").Select

Selection.Copy

With Selection

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlBottom

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Selection.Merge

Selection.Borders(xlDiagonalDown).LineStyle = xlNone

Продолжение ПРИЛОЖЕНИЯ Б

```
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
```

End Sub

'-----

```
Sub Кнопка2_Щелчок() 'распределить
Dim massiv() As String
Dim j As Long
ReDim massiv(ActiveWorkbook.Sheets.Count)
For j = 1 To ActiveWorkbook.Sheets.Count
massiv(j) = ActiveWorkbook.Sheets(j).Name
If (j > 2) Then
Sheets(massiv(j)).Activate

ActiveWorkbook.Sheets(massiv(j)).Range("A8:E250").Select
Application.DisplayAlerts = False
With Selection
    .HorizontalAlignment = xlGeneral
    .VerticalAlignment = xlBottom
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = True
End With
Selection.UnMerge
```

Продолжение ПРИЛОЖЕНИЯ Б

Selection.ClearContents

Selection.RowHeight = 19.5

Application.DisplayAlerts = True

*If Sheets(massiv(j)).Range("C2:E2").MergeCells = False Then
obedinenie (massiv(j))*

Else

Sheets(massiv(j)).Activate

ActiveWorkbook.Sheets(massiv(j)).Range("C2:E2").Select

With Selection

.HorizontalAlignment = xlGeneral

.VerticalAlignment = xlBottom

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = True

End With

Selection.UnMerge

obedinenie (massiv(j))

End If

End If

Next

i = 2

k = 0

m = 0

Z = 0

p = 0

l = 0

Sheets(2).Activate

Do While Sheets(2).Cells(i, 5).Value <> ""

Sheets(2).Cells(i, 4).NumberFormat = "0.00"

Продолжение ПРИЛОЖЕНИЯ Б

```
If (Sheets(2).Cells(i, 5).Value = "Простое письмо") Then
Call raspredelenie(i, massiv(3), k)
k = k + 1
ElseIf Sheets(2).Cells(i, 5).Value = "Заказное письмо" Then
Call raspredelenie(i, massiv(4), m)
m = m + 1
ElseIf Sheets(2).Cells(i, 5).Value = "Заказное письмо с уведомлением" Then
Call raspredelenie(i, massiv(5), Z)
Z = Z + 1
ElseIf Sheets(2).Cells(i, 5).Value = "Заказная бондероль" Then
Call raspredelenie(i, massiv(6), p)
p = p + 1
ElseIf Sheets(2).Cells(i, 5).Value = "Заказная бондероль с уведомлением" Then
Call raspredelenie(i, massiv(7), l)
l = l + 1
End If
i = i + 1
Loop
Application.CutCopyMode = False
'-----
Sheets(1).Activate
Sheets(1).Columns("A:A").Select
Selection.AutoFilter Field:=1, Criteria1:= _
xlFilterToday, Operator:=xlFilterDynamic
Dim r1, r
Dim mas() As String
Sheets(2).Select
For k = 3 To 5
r1 = Sheets(1).Columns(k).SpecialCells(xlCellTypeVisible).Address
If (k = 3) Then
r = Replace(r1, "$C$", "")
ElseIf k = 4 Then
r = Replace(r1, "$D$", "")
ElseIf k = 5 Then
```

Продолжение ПРИЛОЖЕНИЯ Б

r = Replace(r1, "\$E\$", "")

End If

a = ArrayOfValues(r)

mas() = (Split(Join(a, ","), ","))

For j = 2 To i - 1

Sheets("Рабочий лист").Select

Sheets(2).Cells(j, k).Select

Selection.Copy

Sheets(1).Paste Destination:=Sheets(1).Cells(Val(mas(j - 1)), k)

' ++++++

Sheets(1).Activate

Sheets(1).Cells(Val(mas(j - 1)), k).Select

Selection.Borders(xlDiagonalDown).LineStyle = xlNone

Selection.Borders(xlDiagonalUp).LineStyle = xlNone

Selection.Borders(xlEdgeLeft).LineStyle = xlNone

Selection.Borders(xlEdgeTop).LineStyle = xlNone

Selection.Borders(xlEdgeBottom).LineStyle = xlNone

Selection.Borders(xlEdgeRight).LineStyle = xlNone

Selection.Borders(xlInsideVertical).LineStyle = xlNone

Selection.Borders(xlInsideHorizontal).LineStyle = xlNone

Sheets("Рабочий лист").Select

' ++++++

Next

Next

'-----

For j = 3 To ActiveWorkbook.Sheets.Count

Sheets(massiv(j)).Activate

Dim y

y = 8

Do While Sheets(massiv(j)).Cells(y, 2).Value <> ""

Продолжение ПРИЛОЖЕНИЯ Б

y = y + 1

Loop

y = y + 1

*Sheets(massiv(j)).Range(Cells(8, 1), Cells(y, 5)).Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone*

With Selection.Borders(xlEdgeLeft)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlEdgeTop)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlEdgeBottom)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlEdgeRight)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With Selection.Borders(xlInsideVertical)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

Продолжение ПРИЛОЖЕНИЯ Б

.Weight = xlThin

End With

With Selection.Borders(xlInsideHorizontal)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

'=====

Dim stroka, t, ok1, ok2, ok3, ok4

stroka = ""

If ((y - 9) = 11) Or ((y - 9) = 12) Or ((y - 9) = 13) Or ((y - 9) = 14) Then

ok1 = "БХ"

ok2 = "ЕМ"

ok3 = "БХ"

ok4 = "ЕЙ"

Else

Select Case ((y - 9) Mod 10)

Case 1

ok1 = "ОЕ"

ok2 = "БМО"

ok3 = "АЯ"

ok4 = "Б"

Case 2 To 4

ok1 = "БХ"

ok2 = "БМА"

ok3 = "БХ"

ok4 = "И"

Case 0, 5 To 9

ok1 = "БХ"

ok2 = "ЕМ"

ok3 = "БХ"

ok4 = "ЕЙ"

End Select

Продолжение ПРИЛОЖЕНИЯ Б

End If

If y <> 9 Then

If j = 3 Then

stroka = "Итого " & y - 9 & " ПРОСТ" & ok1 & " ПИС" & ok2 & " " & Cells(y - 1, 4).Text & " РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела Принял: оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

ElseIf j = 4 Then

stroka = "Итого " & y - 9 & " ЗАКАЗН" & ok1 & " ПИС" & ok2 & " " & Cells(y - 1, 4).Text & " РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела Принял: оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

ElseIf j = 5 Then

stroka = "Итого " & y - 9 & " ЗАКАЗН" & ok1 & " ПИС" & ok2 & " С УВЕДОМЛЕНИЕМ " & Cells(y - 1, 4).Text & " РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела Принял: оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

ElseIf j = 6 Then

stroka = "Итого " & y - 9 & " ЗАКАЗН" & ok3 & " " & "БАНДЕРОЛ" & ok4 & " " & Cells(y - 1, 4).Text & " РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела Принял: оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

ElseIf j = 7 Then

stroka = "Итого " & y - 9 & " ЗАКАЗН" & ok3 & " " & "БАНДЕРОЛ" & ok4 & " С УВЕДОМЛЕНИЕМ " & Cells(y - 1, 4).Text & " РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела Принял: оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

End If

ElseIf y = 9 Then

If j = 3 Then

stroka = "Итого " & 0 & " ПРОСТЫХ ПИСЕМ 0,00 РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела Принял:

Продолжение ПРИЛОЖЕНИЯ Б

оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

ElseIf j = 4 Then

stroka = "Итого " & 0 & " ЗАКАЗНЫХ ПИСЕМ 0,00 РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела

Принял:

оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

ElseIf j = 5 Then

stroka = "Итого " & 0 & " ЗАКАЗНЫХ ПИСЕМ С УВЕДОМЛЕНИЕМ 0,00 РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела

Принял: оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

ElseIf j = 6 Then

stroka = "Итого " & 0 & " ЗАКАЗНЫХ БАНДЕРОЛЕЙ 0,00 РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела

Принял:

оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

ElseIf j = 7 Then

stroka = "Итого " & 0 & " ЗАКАЗНЫХ БАНДЕРОЛЕЙ С УВЕДОМЛЕНИЕМ 0,00 РУБ." & vbCrLf & "Сдал Старший инспектор Общего отдела

Принял: оператор_____ " & vbCrLf & "Мордовского отделения № 8589 Сбербанка России"

End If

End If

ActiveWorkbook.Sheets(massiv(j)).Cells(y, 1).Value = stroka & vbCrLf & "Подпись_____

Подпись_____ " & vbCrLf & "ФИО
ФИО

_____ " & vbCrLf & "- М.П.

Оттиск КШ"

Range(Cells(y, 1), Cells(y + 2, 5)).Select

With Selection

Окончание ПРИЛОЖЕНИЯ Б

```
.HorizontalAlignment = xlLeft  
.VerticalAlignment = xlCenter  
.WrapText = True  
.Orientation = 0  
.AddIndent = False  
.IndentLevel = 0  
.ShrinkToFit = False  
.ReadingOrder = xlContext  
.MergeCells = True
```

End With

Cells(y, 1).Select

With Selection.Font

```
.Name = "Times New Roman"  
.Size = 8  
.Strikethrough = False  
.Superscript = False  
.Subscript = False  
.OutlineFont = False  
.Shadow = False  
.Underline = xlUnderlineStyleNone  
.ColorIndex = xlAutomatic  
.TintAndShade = 0  
.ThemeFont = xlThemeFontNone
```

End With

Selection.Font.Bold = True

Rows(y).RowHeight = 18

Rows(y + 1).RowHeight = 9

Rows(y + 2).RowHeight = 104.25

Next

'=====

Sheets(1).Select

Columns("A:A").Select

ActiveSheet.Range("\$A:\$A").AutoFilter Field:=1

End Sub

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.П. ОГАРЁВА”

**ОТЗЫВ РЕЦЕНЗЕНТА
О ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ**

Студента Вдовиной Яны Евгеньевны
Факультет математики и информационных технологий
Кафедра фундаментальной информатики
Группа: 402
Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии
Квалификация (степень): Бакалавр
Наименование темы: Практическое применение сценариев Windows для решения офисных задач
Рецензент: Сухарев Л. А., зав. кафедрой алгебры и геометрии, к.ф.-м.н.

ОЦЕНКА ВЫПУСКНОЙ РАБОТЫ

№ п/п	Показатели	Оценка				
		5	4	3	2	0*
1.	Актуальность тематики работы		X			
2.	Степень полноты обзора состояния вопроса и корректность постановки задачи		X			
3.	Уровень и корректность использования в работе методов исследования, математического моделирования		X			
4.	Степень комплексности работы, применение в ней знаний естественнонаучных, социально-экономических, общепрофессиональных и специальных дисциплин			X		
5.	Ясность, четкость, последовательность и обоснованность изложения		X			
6.	Применение современного математического и программного обеспечения, компьютерных технологий в работе		X			
7.	Качество оформления (общий уровень грамотности, стиль изложения, качество иллюстраций, соответствие требованиям стандарта)		X			
8.	Оригинальность и новизна полученных результатов (научных, конструкторских и технологических решений)			X		
9.	Тип работы	фундаментальная с оригинальными результатами				
		реферативная				
		прикладная		X		
10.	Рекомендации	к опубликованию				
		к внедрению			X	
ИТОГОВАЯ ОЦЕНКА		Хорошо				

* – не оценивается (трудно оценить)

Отмеченные достоинства:

1) Автор достаточно аккуратно подобрал материал, четко выдерживая логическую структуру и взаимосвязь между разделами и приложениями. Все разработанные сценарии размещены в Приложении Б, основной текст работы свободен от листингов, иллюстрирующих работу приложения, что делает работу более доступной для понимания.

2) Работа имеет практическое значение, разработанный сценарий можно модифицировать для работы в другой организации.

Отмеченные недостатки:

1) Алгоритм работы созданного сценария для оформления реестра почтовых отправок описан недостаточно подробно.

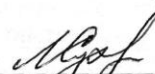
Заключение:

Учитывая отмеченные выше достоинства и недостатки, считаю, что выпускная работа Я. Е. Вдовиной удовлетворяет всем требованиям, предъявляемым к выпускным квалификационным работам, и заслуживает оценки "хорошо".

Автор работы, Вдовина Яна Евгеньевна, достойна присвоения степени бакалавра по направлению «Фундаментальная информатика и информационные технологии».

14 июня 2016 г.

Рецензент


(подпись)

Отзыв на выпускную бакалаврскую работу
«Практическое применение сценариев Windows
для решения офисных задач»
студента 4 курса дневного отделения
факультета математики и информационных технологий
Я.Е. Вдовиной

В настоящее время одним из стандартных инструментов, позволяющих самостоятельно автоматизировать повседневную работу в операционной системе Windows, является сервер сценариев Windows Script Host (WSH). Обычно эти сценарии используются в качестве утилит для управления операционной системой, но их также можно применять для автоматизации различных офисных задач (при этом фактически идет обращение к объектным моделям пакета Microsoft Office).

В данной бакалаврской работе была поставлена задача изучения сценариев WSH и практического использования этих сценариев для решения задачи по автоматизации процесса учета почтовых посылок, отправляемых из организации.

Автор работы в основном раскрыл заявленную тему, был проведен сравнительный анализ наиболее распространенных офисных пакетов, рассмотрены примеры сценариев, использующих объектные модели Microsoft Word и Microsoft Excel, разработана утилита для учета почтовых отправлений. Считаю, что бакалаврская работа Я.Е. Вдовиной заслуживает оценки "хорошо".

Руководитель дипломной работы
к.ф. – м.н., доцент кафедры
фундаментальной информатики



А.В. Попов