Саратовский Государственный Университет
им. Н.Г.Чернышевского

КАФЕДРА ФИЗИКИ ТВЕРДОГО ТЕЛА

Методика

СОЗДАНИЯ УЧЕБНО-МЕТОДИЧЕСКИХ ДОКУМЕНТОВ ФОРМАТА *.WML

САРАТОВ
2008

УДК 621.382-181(075.8)
ББК 32.844.1 я73
С30

Семёнов А.А.

С30 Методика создания учебно-методических документов формата *.WML – Саратов. СГУ. – 58 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

WAP. Историческая справка.

Всего лишь чуть более пяти лет назад цивилизованный мир узнал о новой разработке гигантов телефонного хай-тека (**Nokia**, **Motorola**, **Phone.com**, **Ericsson**). Это был **WAP** (*Wireless Application Protocol*). Изначально задумывалось, что с помощью нового протокола люди смогут заказывать билеты на самолеты и поезда, совершать банковские операции и делать покупки в супермаркетах. Но, несмотря на все усилия разработчиков, WAP не прижился. Низкая скорость передачи данных, слабая защищенности каналов, дороговизна. Причин было много. WAP умирал. Но пришло спасение, и имя ему было **GPRS** — высокоскоростная пакетная передача данных. Скрестив эти технологии, разработчики WAP избавились практически от всех недостатков своего творения. А так как большинство моделей "трубок" уже имели в своем арсенале как GPRS, так и WAP, успех дуэту был обеспечен.

Мобильная телефония и мобильный Интернет все глубже и глубже входят в нашу жизнь. По статистическим данным, известным из средств массовой информации, мобильным аппаратом обладает практически каждый второй житель нашей страны и процесс насыщения ими весьма активно продолжается.

Мобильные Интернет-технологии в учебном процессе ВУЗа.

На сегодняшний день мобильный Интернет (как часто называют WAP) становится все более популярным. И по существующим наблюдениям учащиеся оснащены мобильными средствами связи в большей степени, нежели сами

преподаватели. Было бы, наверное, весьма недальновидно выпускать из средств учебного процесса такой резерв хоть и специфического, но всё же развитого компьютерного парка. Не говоря уже о том, что все учебные наработки по контролю знаний и тестированию для сети Интернет с небольшими изменениями пригодны для работы с сотовыми телефонами (следует только учесть особенности Интернет–средств для мобильного доступа). Сотовые телефоны, обладая графическими жидкокристаллическими дисплеями, вполне пригодны для работы с текстовыми материалами, моделирования физических и химических процессов, геометрических построений и манипуляций с трёхмерной графикой.

Опыт преподавания дисциплин специализации и проведения практических занятий для старших курсов показывает, что посещаемость лекций и практикумов снижается. В большинстве случаев это вызвано вполне объективными причинами: в современных условиях студенты старших курсов вынуждены зарабатывать, поскольку к этому времени некоторые уже имеют собственные семьи. Интернет технологии позволяют частично снять остроту проблемы снижения качества знаний, связанную с непосещаемостью. Так лекционные и другие учебные материалы можно разместить на файловом сервере учебного заведения для открытого доступа студентов, изучающих данную дисциплину. WAP добавляет к Интернету новое измерение – мобильность, и открывает новые возможности для мобильных телефонов, делая их не просто телефонами, а устройствами, оперативно связывающими своих абонентов с Интернетом вне зависимости от их места нахождения. Информация становится доступной там, где она необходима.

Мобильные сетевые технологии позволяют легко разрешить проблему с доступностью методической и учебной литературы по изучаемым дисциплинам. Тем более, что требования федерального агентства по образованию предписывают обеспечить наличие на кафедре, реализующей конкретную учебную дисциплину, современной литературы по указанной дисциплине, в том числе и в электронном виде. В том случае, если соответствующая литература специфична или труднодоступна, рекомендуется иметь конспекты соответствующих лекций или другие учебные материалы в электронном или печатном виде.

Для подавляющего большинства дисциплин, преподаваемых в высшем учебном заведении (ВУЗ), сотрудниками кафедр, обеспечивающих преподавание указанных дисциплин, разработаны так называемые учебно-методические

комплексы (УМК). УМК включают в себя рабочие программы преподаваемых дисциплин и различные методические материалы, в том числе пособия и лекционные материалы. Таким образом, для публикации на WAP-сайте в формате ***.wml** имеются исходные документы в форматах ***.doc (*.rtf)**, ***.pdf** или ***.htm** (если документы уже публиковались на WEB-сайтах ВУЗ-а).

Методика создания документов формата *.WML

Практика показывает, что распространенные модели современных мобильных телефонов и коммуникаторов безошибочно воспринимают ***.wml** контент размером до 10 страниц, сформированных в MS Word. Причем коммуникаторы, снабженные WEB–браузером, поддерживают полноценный протокол **HTTP** и способны успешно читать документы в форматах ***.htm (*.html)**. Но, поскольку WAP-сайт предназначен для чтения широким кругом мобильных устройств, и скорость передачи ограничена, исходные документы рекомендуется разбить на блоки меньшего размера. Различные источники определяют максимальный размер такого блока в пределах 1.4—2 КБайт, что обусловлено доступным объёмом оперативной памяти типичного мобильного браузера при работе с WAP-страницами. С учетом того, что WAP-шлюз осуществляет предварительное сжатие ***.wml** документов, можно порекомендовать разбиение исходных файлов на блоки размером, определяемым стандартной страницей, сформированной в MS Word (38-39 строк, шрифт с кеглем 12-14).

Постраничное разбиение с привязкой к формату MS Word удобно выполнить автоматически без участия автора документа, поскольку исходные материалы чаще всего формируются именно в Word, и сложные элементы форматирования (колонки и таблицы), а также рисунки уже логически размещены автором в переделах заданных страниц, что исключает из процесса преобразования процедуру синтаксического и логического анализа исходного материала.

Документы формата ***.htm (*.html)** с гипертекстовой разметкой представляют собой структуру, сформированную специальными элементами разметки - тэгами ('tag' - дословно признак). Большинство тэгов представляют собой парные элементы (открывающий - закрывающий) и разметка в общем случае образует сложную последовательность вложенных конструкций, что

затрудняет корректное разбиение ***.html** документа на страницы. Подавляющее большинство известных программ выполняют такую процедуру с потерей формата исходного текста. Положение также осложняется тем, что язык гипертекстовой разметки **html** не отличается строгостью, поэтому логичным и менее трудоёмким представляется преобразование ***.html** документа в формат MS Word с его дальнейший обработкой в среде последнего.

Документы формата ***.pdf** в принципе довольно трудно разделить на более мелкие части с сохранением исходного форматирования. И в этом случае для формирования WAP-контента следует преобразовать ***.pdf**-файл в документ формата MS Word посредством его обработки в пакете **ABBYY Fine Reader** версии не менее 4.0. Альтернативой может быть промежуточное преобразование ***.pdf** документа в ***.html** файл с помощью утилиты **PDF2HTML**. Недостатком подобного решения представляется то, что утилита **PDF2HTML** не всегда корректно обрабатывает фокументы в русской кодировке, а также тот факт, что последние доступные в Сети версии этого программного продукта являются условно-бесплатными ("shareware") и имеют ряд функциональных ограничений.

Таким образом, процесс формирования конечного ***.wml** файла в общем случае можно представить в виде последовательности преобразований:

{**документ.pdf** *или* **документ.htm**} —> **документ.doc** *или* **документ.rtf** —>

Документ1.doc		**Документ1. htm**		**Документ1. wml**
Документ2.doc	—>	**Документ2. htm**	—>	**Документ2. wml**
…		**…**		**…**
ДокументN.doc		**ДокументN.htm**		**ДокументN.wml**

причем документы 1…N снабжаются уже на стадии создания перекрестными гиперссылками. Здесь же удобно провести предварительный анализ документа и убрать элементы форматирования, не поддерживаемые WAP–браузерами (как, например, выравнивание текста по ширине страницы).

Рассмотрим процедуру полуавтоматического преобразования более подробно.

Предварительный этап: создание исходного документа MS Word.

1. Исходный документ - в формате ***.html**.

В среде MS Word последовательно выбрать опции <**Файл**> → <**Открыть**>. В окне "**Открытие документа**" установить "**Тип файлов**" –

"**Документ HTML** (***.htm, *.html**)". В таблице файлов найти нужный документ и выбрать его, подтвердив открытие нажатием кнопки "**Открыть**" или клавишей "**Enter**".

После того, как документ отобразится в окне MS Word, последовательно выбрать опции <**Файл**> → <**Свойства**>. В открывшемся окне MS Word 97 "**Свойства документа** " в разделе "**Кодировка HTML**" рекомендуется выбрать свойство "**Кириллица**" "**для отображения этой страницы:**", "**для сохранения этой страницы:**", "**для создания новых Web-страниц (основная кодировка):**", а также установить галочку для опции "**сохранять Web-страницы в основной кодировке**". Выполнив эти установки применить их к документу нажатием кнопки "**Ok!**".

В MS Word 2000 и более старших версий следует дополнительно отменить сохранение в документе сложного форматирования, формирование списков CSS, и выбрать предполагаемую версию браузера для просмотра Internet Explorer 3.0, что также снизит сложность форматирования сохраняемого документа (см. **Приложение 4**). Контекстная помощь по выбору этих опций в каждом конкретном случае предоставляется опцией локальной подсказки (нажатие [**?**] и указание курсором мыши на интересующее поле или опцию).

После того, как свойства сохраняемого документа определены, необходимо сохранить его как документ MS Word или файл формата ***.rtf** следующей последовательностью: <**Файл**> → <**Сохранить как…**>; в окне "**Сохранение документа**" выбрать "**Тип файла**" – "**Документ WORD** (***.doc**)" или "**Текст в формате RTF** (***.rtf**)", после чего нажать кнопку "**Сохранить**".

С точки зрения формирования конечного ***.wml** файла, сохранение в формате **RTF** можно считать более предпочтительным, поскольку в этом формате автоматически заменяются некоторые сложные элементы формата ***.doc** на более простые, что в дальнейшем обеспечит более корректную процедуру преобразования. Поэтому если MS Word предупреждает о такой возможности при сохранении, процедуру сохранения необходимо подтвердить, нажав "**Да**" в предупреждающем окне.

Установки, сделанные в свойствах файла для формата ***.html** также будут использованы в процессе преобразование в этот формат группы файлов ***.doc** .

2. Исходный документ - в формате ***.pdf**.

а) ***преобразование*** ***.pdf** ***документа с помощью утилиты*** **PDF2HTML.**

В меню "**Tools**" утилиты выбрать процесс преобразования "**PDF2HTM**". В настройках преобразования (<**File**> → <**Preferences**>) обычно выставлены по умолчанию все необходимые опции, и если есть необходимость что-то изменить, то лучше ознакомиться с описанием опций в меню помощи <**Help**>. Для начала преобразования документа следует выбрать его в меню <**File**> → <**Open**>, после чего программа предлагает в окне "**Сохранение**" утвердить название папки для формируемых ***.html** страниц документа (обычно по имени самого исходного документа). Если выбор подтвержден нажатием кнопки "**Сохранить**", запускается процедура преобразования ***.pdf** документа. Обычно по её завершению формируется ***.html** - файл с названием исходного документа, содержащий ссылки на файлы отдельных страниц, находящиеся в упомянутой ранее папке. Если сформированные ***.html** - страницы при просмотре отображаются корректно (что в случае русской кодировки - скорее исключение, чем ожидаемый результат), то следует приступить к их обработке с целью преобразования в документы MS Word согласно рекомендациям, приведенным в **разделе 1** (***Исходный документ - в формате*** ***.html**). В случае неудачной перекодировки, можно использовать следующий вариант.

б) ***преобразование*** ***.pdf** ***документа в OCR пакете*** **ABBYY Fine Reader.**

Если пакет **ABBYY Fine Reader** корректно установлен в системе, то ***.pdf** документ удобно открыть для преобразования следующим образом: указать на него одиночным кликом правой кнопки мыши в папке, где он находится, после чего, удерживая нажатой клавишу [**Shift**], левой кнопкой мыши вызвать контекстное меню, кликнув по выбранному файлу (см. Рис.1). В открывшемся меню выполнить опцию "**Open with Fine Reader**". Если такая опция отсутствует, то выполняем – "**Открыть с помощью**", после чего в появившемся меню находим всё тот же **Fine Reader** (или **Fine32**).

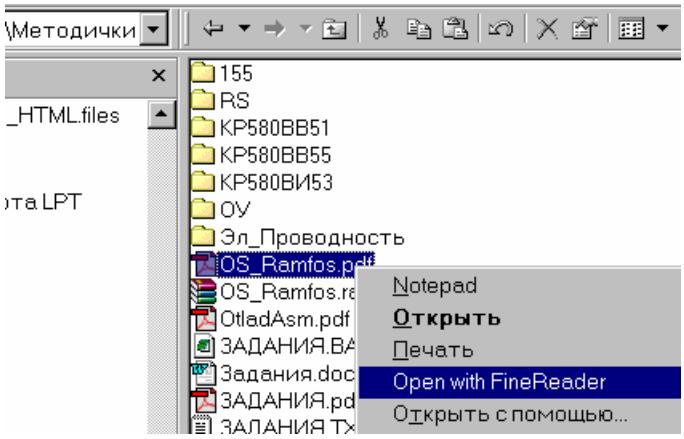


Рис. 1.

После запуска, **Fine Reader** начинает процесс преобразования документа ***.pdf** в пакет с постраничным набором картинок, которые далее следует разделить на области текста, картинок, таблиц и включить режим распознавания. Процесс можно выполнить полностью автоматически с помощью встроенного "**Мастера Scan & Read**" и, поскольку качество формируемых изображений страниц достаточно хорошее, в итоге получаем набор вполне корректных документов MS Word, которые могут быть сохранены как файлы или переданы для обработки непосредственно в Word.

Итак, в наличии имеется исходный документ MS Word (или набор документов).

Первый этап: создание комплекта документов в формате ***.html**, предназначенных для преобразования в группу документов формата ***.wml**.

а) ***полуавтоматическое преобразование исходного документа MS Word в комплект документов.***

На этом этапе формируется пакет коротких документов формата ***.doc**, объединённых коротким групповым именем и снабженных упорядоченными номерами 1…N, причем документы 1…N уже на стадии создания комплектуются корректными перекрестными гиперссылками. Здесь же корректируются элементы форматирования, не поддерживаемые WAP–браузерами.

Если исходные документы получены путём преобразования ***.pdf** или ***.html** файлов, их следует объединить в один общий документ, открыв в Word первый из документов и добавив в него все последующие, помещая курсор экрана в позицию за последней строкой файла и добавляя очередной файл следующим образом: <**Вставка**> → <**Файл**>, в открывшемся окне "**Вставка файла**" выбрать "**Тип файла**" – "**Документ WORD (*.doc)**" или "**Текст в формате RTF (*.rtf)**", после чего выбрать нужный файл и нажать кнопку "**OK**".

После того, как все файлы объединены в один общий, его необходимо сохранить как "**Текст в формате RTF"**, причем имя файла следует незначительно изменить, к примеру, добавив к нему символ "_". Такой прием позволяет в процессе создания нового файла удалить из его содержимого большое количество служебной информации MS Word, что положительно скажется на корректности дальнейшего процесса преобразования.

Поскольку MS Word поддерживает выполнение макросов на языке Microsoft Visual Basic for Applications (Microsoft VBA - Визуал Бэйсик для приложений), автоматическую обработку логично выполнить, используя логические конструкции данного языка. С этой целью сформируем управляющий элемент панели для макроса типа 'кнопка', выбирая пункты меню <**Сервис**> → <**Макрос**> → <**Начать запись**>; в открывшемся окне "**Запись макроса**" ввести "**Имя макроса**": **SavePages**. После подтверждения кнопкой "**OK**", начнется запись макроса и появится его панель управления (Рис. 2).

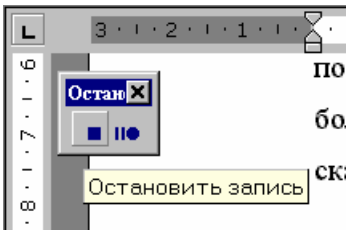


Рис. 2.

Нажатием специальной кнопки запись следует остановить, после чего перейти к редактированию созданного пустого макроса последовательностью команд: <**Сервис**> → <**Макрос**> → <**Макросы**>, в открывшемся окне "**Макрос**" выбрать "**Имя:**" записанного макроса – **SavePages** – и нажать кнопку [**Изменить**].

В появившемся окне редактора Microsoft Visual Basic отобразится примерно следующий текст:

```
Sub SavePages()
'
' SavePages Макрос
' Макрос записан 10.01.08 'Name'
'
End Sub
```

Этот текст следует выделить полностью и вставить поверх него следующий программный листинг, предварительно скопированный ***отсюда*** в буфер обмена:

```
Sub SavePages()
'_____________для MS(c)Word_97'_____________________________________
'__Макрос постраничного сохранения документа_____________________
'__(c) 10.01.08 A.A.Semenov_____________________________________
If Documents.Count >= 1 Then
   If (SaveAsPages = True) Then
      MsgBox "Your Document is Successfully divided & saved!" &
Chr(13) _
      & Chr(10) & "    Группа документов успешно создана!"
   Else
      MsgBox "Error in division of Your Document!" & Chr(13) _
      & Chr(10) & " Ошибка сохранения документов!"
   End If
Else
    MsgBox "Sorry! No documents are open!" & Chr(13) _
    & Chr(10) & "Нет открытых документов!"
End If
End Sub

Function SaveAsPages() As Boolean
'
Dim PageCount As Integer, CurPage As Integer, CrnLine As Integer,
LineCount As Integer
Dim AcPane As Pane
Dim Message, Title, Default, Style, Response, strMes, filePath As
String

Message = "Введите короткое латинское имя группы для сохраняемых
файлов, типа L1 (для лекций) или M1 (для методичек). Короткое имя
более удобно для набора с клавиатуры мобильного устройства." '
Сообщение-подсказка.
  Title = " Имя группы сохраняемых файлов" '--- Заголовок окна ввода.
Default = "L1" '--- Значение по умолчанию.
'____Выводит на экран сообщение, заголовок и значение по умолчанию.
   strMes = InputBox(Message, Title, Default)
If strMes = "" Then strMes = "L0"

 Message = "Применить стиль <small> (мелкий шрифт)" & Chr(13) _
           & Chr(10) & "ко всем вновь создаваемым документам?"
   Title = " Стиль текста документа"
   Style = vbYesNo + vbQuestion + vbDefaultButton1  '--- Кнопки.

Response = MsgBox(Message, Style, Title, "", 1000)

On Error GoTo errr
```

```
With ActiveDocument
    '____определим текущий путь открытого документа___________
    filePath = ActiveDocument.Path & Application.PathSeparator

    '--- сохраним текущий документ, как временный ----
    ActiveDocument.SaveAs FileName:=filePath & "$temp$.doc",
FileFormat:=wdFormatDocument, _
    LockComments:=False, Password:="", AddToRecentFiles:=True, _
    WritePassword:="", ReadOnlyRecommended:=False,
EmbedTrueTypeFonts:=False, _
    SaveNativePictureFormat:=True, SaveFormsData:=False, _
    SaveAsAOCELetter:=False
    '--- количество параграфов документа
    LineCount = .ComputeStatistics(wdStatisticParagraphs)

    '--- количество страниц документа
    PageCount = .ComputeStatistics(wdStatisticPages)
    '--- переводим док. в режим разметки страницы
    ActiveWindow.View.Type = wdPageView

'    Selection.GoTo What:=wdGoToLine, Which:=wdGoToFirst, Count:=1,
Name:=""
'    переход на 1 строку

    Set AcPane = .ActiveWindow.Panes(1)
    '____курсор - в начало документа
       Selection.HomeKey Unit:=wdStory
    '____замена форматирования 'Justify' с начала документа
    For CrnLine = 1 To LineCount
      If Selection.ParagraphFormat.Alignment = wdAlignParagraphJustify
Then
         Selection.ParagraphFormat.Alignment = wdAlignParagraphLeft
      End If
         Selection.MoveDown Unit:=wdParagraph, Count:=1
    Next
    '____перевод курсора к концу документа ()
       Selection.EndKey Unit:=wdStory
    '____замена форматирования 'Justify' с конца документа
    For CrnLine = 1 To LineCount
      If Selection.ParagraphFormat.Alignment = wdAlignParagraphJustify
Then
         Selection.ParagraphFormat.Alignment = wdAlignParagraphLeft
      End If
         Selection.MoveUp Unit:=wdParagraph, Count:=1
    Next

    '____курсор - в начало документа
       Selection.HomeKey Unit:=wdStory
    '--- пробегаем по всем страницам
    For CurPage = 1 To PageCount
    '--- в зависимости от № текущий страницы
      Select Case CurPage
        Case 1
    '____перевод курсора на страницу вниз
        Selection.GoTo What:=wdGoToPage, Which:=wdGoToNext, Count:=1,
Name:=""

        Selection.TypeParagraph
        Selection.TypeParagraph
        Selection.TypeParagraph
```

```
        Selection.MoveUp Unit:=wdLine, Count:=1
        Selection.TypeText Text:="Next"
        Selection.HomeKey Unit:=wdLine, Extend:=wdExtend

        ActiveDocument.Hyperlinks.Add Anchor:=Selection.Range, _
        Address:=strMes & "_" & (CurPage + 1) & ".wml", _
        SubAddress:=""

        Selection.HomeKey Unit:=wdLine, Extend:=wdExtend
        Selection.ParagraphFormat.Alignment = wdAlignParagraphCenter
        Selection.MoveDown Unit:=wdLine, Count:=1
        Selection.HomeKey Unit:=wdLine

        Case 2 To PageCount - 1
    '____перевод курсора на страницу вниз
        Selection.GoTo What:=wdGoToPage, Which:=wdGoToNext, Count:=1,
Name:=""

        Selection.TypeParagraph
        Selection.TypeParagraph
        Selection.TypeParagraph
        Selection.MoveUp Unit:=wdLine, Count:=1
        Selection.TypeText Text:="Back"
        Selection.HomeKey Unit:=wdLine, Extend:=wdExtend
        ActiveDocument.Hyperlinks.Add Anchor:=Selection.Range, _
        Address:=strMes & "_" & (CurPage - 1) & ".wml", _
        SubAddress:=""
        Selection.TypeText Text:=" --- Next"
        Selection.MoveLeft Unit:=wdCharacter, Count:=4,
Extend:=wdExtend
        ActiveDocument.Hyperlinks.Add Anchor:=Selection.Range, _
        Address:=strMes & "_" & (CurPage + 1) & ".wml", _
        SubAddress:=""
        Selection.HomeKey Unit:=wdLine, Extend:=wdExtend
        Selection.ParagraphFormat.Alignment = wdAlignParagraphCenter
        Selection.MoveDown Unit:=wdLine, Count:=1
        Selection.HomeKey Unit:=wdLine

       Case Else '--- последняя страница
    '____перевод курсора к концу документа
        Selection.EndKey Unit:=wdStory
        Selection.TypeParagraph
        Selection.TypeParagraph
        Selection.TypeParagraph
        Selection.MoveUp Unit:=wdLine, Count:=1
        Selection.TypeText Text:="Back"
        Selection.HomeKey Unit:=wdLine, Extend:=wdExtend
        ActiveDocument.Hyperlinks.Add Anchor:=Selection.Range, _
        Address:=strMes & "_" & (CurPage - 1) & ".wml", _
        SubAddress:=""
        Selection.HomeKey Unit:=wdLine, Extend:=wdExtend
        Selection.ParagraphFormat.Alignment = wdAlignParagraphCenter
        Selection.MoveDown Unit:=wdLine, Count:=1
        Selection.HomeKey Unit:=wdLine
      End Select

    '____выделить до начала документа
        Selection.HomeKey Unit:=wdStory, Extend:=wdExtend

    '____вырезать выделенный текст
        Selection.Cut
```

```
'____создать новый документ
        Documents.Add
        Selection.MoveRight Unit:=wdCharacter, Count:=1,
Extend:=wdExtend
    '____указать шрифт и кегль
        Selection.Font.Name = "Times New Roman"
        Selection.Font.Size = 10
'____вставить выделенный текст в новый документ
        Selection.Paste
    '____выделить всю страницу и установить шрифт
        Selection.WholeStory
     If Response = vbYes Then '--- Нажата кнопка "Да" (Yes).
        Selection.Font.Subscript = wdToggle
       Else '---------------------- Нажата кнопка "Нет" (No).
        Selection.Font.Size = 10
    End If
    '____сбросить выделение___________
        Selection.HomeKey Unit:=wdLine, Extend:=wdExtend
'____сохранить новый документ WORD с заданным именем
        ActiveDocument.SaveAs FileName:=filePath & strMes & "_" &
CurPage & ".doc", FileFormat:=wdFormatDocument, _
        LockComments:=False, Password:="", AddToRecentFiles:=True, _
        WritePassword:="", ReadOnlyRecommended:=False,
EmbedTrueTypeFonts:=False, _
        SaveNativePictureFormat:=True, SaveFormsData:=False, _
        SaveAsAOCELetter:=False
    '____закрыть новый документ
        ActiveWindow.Close
Next
'____сохранить временный документ
ActiveDocument.Save
'____закрыть временный документ
ActiveWindow.Close

End With

SaveAsPages = True
Exit Function

errr:
SaveAsPages = False
End Function

Sub SaveHTM()
'_____________для MS(c)Word_97'_____________________________________
'____Макрос сохранения документов в формате HTML_____________________
'____(c) 10.01.08 A.A.Semenov_______________________________________
If Documents.Count >= 1 Then
   If (SaveAsHTM = True) Then
      MsgBox "    Your Documents are Successfully saved as *.HTM!" &
Chr(13) _
      & Chr(10) & "Документы успешно сохранены в формате *.HTM!"
    Else
      MsgBox " Error while saveing Your Documents!" & Chr(13) _
      & Chr(10) & "Ошибка сохранения документов!"
   End If
 Else
    MsgBox "Sorry! No documents are open!" & Chr(13) _
    & Chr(10) & "Нет открытых документов!"
End If
End Sub
```

```
Function SaveAsHTM() As Boolean
'
Dim DocumCount, CurDocum, TestDocum, pos, Ver As Integer
Dim myDocname, CurName, WinName, WordVer As String

'_____узнаем версию MS_Word (8.0 для 97, 11.0 для 2003)___
WordVer = Application.Version
Ver = Val(Left(WordVer, 2))
On Error GoTo ermet

DocumCount = Documents.Count

For CurDocum = 1 To DocumCount
If Ver = 8 Then
 If CurDocum = 1 Then
    MsgBox "     Выберите кодировку 'Кириллица' и нажмите 'Ok'" &
Chr(13) _
    & Chr(10) & "       столько раз, сколько предложит программа." &
Chr(13) _
    & Chr(10) & "(Модальные окна не дают закрыть себя программно.)"
 End If
End If
    Documents(1).Activate
    CurName = ActiveDocument.Name
    pos = InStr(CurName, ".")
 If pos > 0 Then
    myDocname = Left(CurName, pos - 1)
  Else
    myDocname = CurName
 End If

With ActiveDocument
If Ver = 8 Then
   '____сохранение HTML в Word_97_____________________
    ActiveDocument.SaveAs FileName:=ActiveDocument.Path &
Application.PathSeparator & myDocname & ".htm", FileFormat:=100, _
    LockComments:=False, Password:="", AddToRecentFiles:=True, _
    WritePassword:="", ReadOnlyRecommended:=False,
EmbedTrueTypeFonts:=False, _
    SaveNativePictureFormat:=False, SaveFormsData:=False,
SaveAsAOCELetter:=False
    '____здесь вылезает окно Word_97 с кнопкой и всё тормозит до
нажатия 'Ok'___________
    ActiveDocument.Close
  Else
    '____сохранение HTML (с фильтром) в Word_2003_____________________
    ActiveDocument.SaveAs FileName:=ActiveDocument.Path &
Application.PathSeparator & myDocname & ".htm",
FileFormat:=wdFormatFilteredHTML, _
        LockComments:=False, Password:="", AddToRecentFiles:=True,
WritePassword:="", _
        ReadOnlyRecommended:=False, EmbedTrueTypeFonts:=False, _
        SaveNativePictureFormat:=False, SaveFormsData:=False, _
        SaveAsAOCELetter:=False
    ActiveWindow.View.Type = wdWebView
    ActiveWindow.Close
End If
    TestDocum = Documents.Count
End With

Next
```

```
'---------------------------------------------------------------------
SaveAsHTM = True
Exit Function

ermet:
SaveAsHTM = False
End Function
'---------------------------------------------------------------------
```

После вставки следует выбрать опцию меню <**Файл**> → <**Сохранить**>.

Далее необходимо организовать управляющие элементы панели ('кнопки') для макросов **`SavePages()`** и **`SaveHTM()`**, для этого в меню <**Сервис**> выбираем пункт <**Настройка…**> и в окне "**Настройка** ", выбрав закладку <**Команды**>, в разделе "**Категории:**" открываем подраздел "**Макросы**". В окне "**Команды**:" при этом должны обнаружиться записанные макросы:

Normal.NewMacros.SavePages и **Normal.NewMacros.SaveHTM** .

Удерживая запись макроса левой кнопкой мыши, вытаскиваем новую кнопку управления макроса и помещаем её на выбранную панель (Рис. 3).

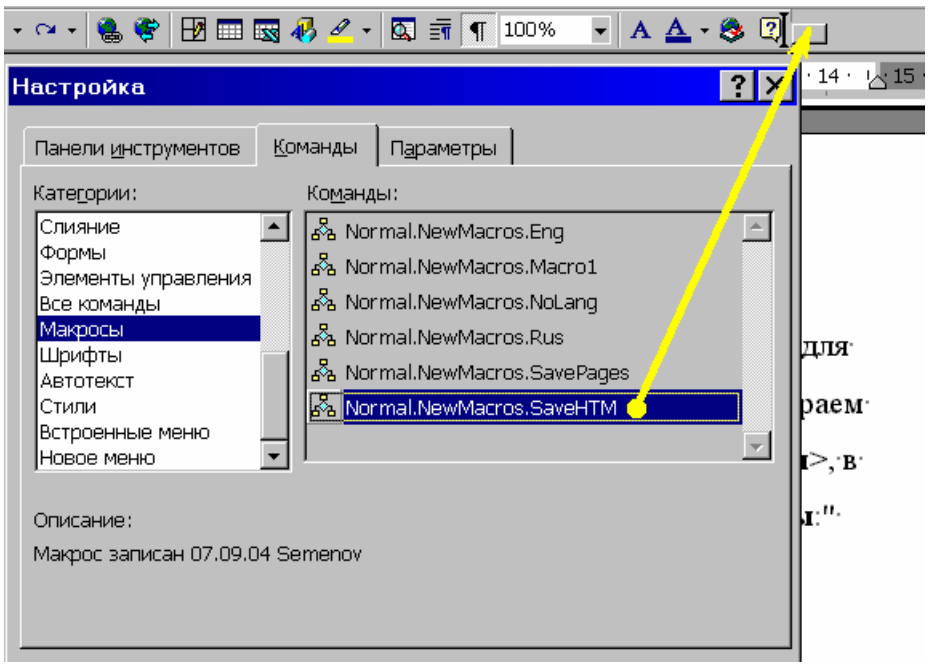


Рис. 3.

Для вновь организованного элемента управления с достаточно длинным заголовком "**Normal.NewMacros.SaveHTM**" следует выбрать короткое имя и соответствующий значок. Эти действия производятся выбором пунктов меню, которое появляется при клике правой кнопкой мыши по изображению созданного

элемента управления (Рис. 4). В меню "**Имя:**" задаем имя элемента: **SaveHTM**, а пункт меню "**Выбрать значок для кнопки**" позволяет украсить кнопку пиктограммой. Соответствующие манипуляции следует предпринять и для макроса **Normal.NewMacros.SavePages**, после чего в выбранной панели появятся две кнопки управления **SavePG** и **SaveHTM**, а среда MS Word будет готова для дальнейшей работы (Рис. 5).

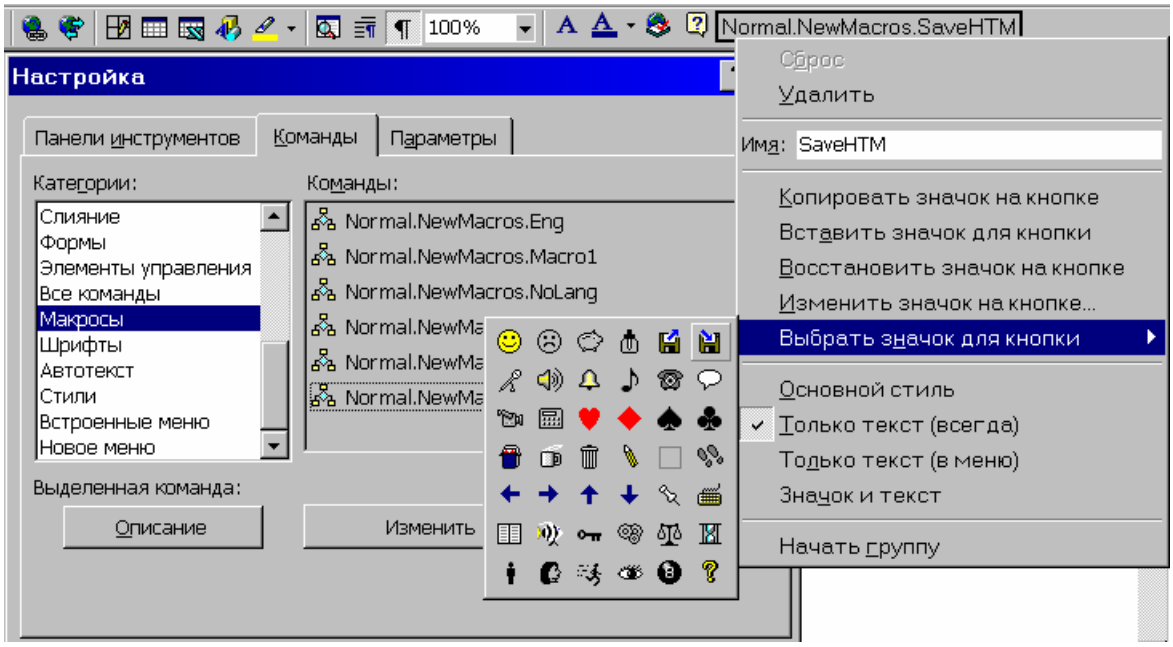


Рис. 4.

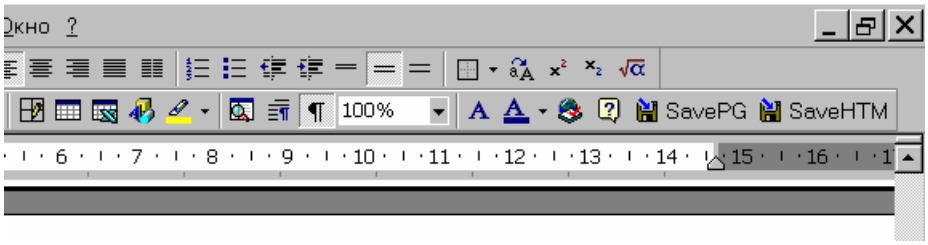


Рис. 5.

Открываем сохраненный ранее исходный документ MS Word (<**Файл**> → <**Открыть**>) для полуавтоматического преобразования. В этот момент удобно факультативно выполнить некоторые преобразования, реализация которых в автоматическом режиме затруднена.

Примером таких манипуляций может быть присвоение подрисуночным подписям гиперссылок, введение которых в преобразованный документ в дальнейшем будет сравнительно трудным действием. С этой целью в документе находятся все подписи к рисункам типа "**Рис. 1. - Рис. N**", которые выделяются как текст (**Рис. 1.**), и им присваиваются относительные ссылки на несуществующую пока папку с рисунками (например, pic/pic_1.gif – последовательность действий: <**Вставка**> → <**Гиперссылка**>; в окне "**Добавить гиперссылку**" ввести URL гиперссылки типа: pic/pic_1.gif, где номер в названии рисунка соответствует номеру в выделенном подрисуночном тексте; на этой же вкладке установить галочку в позиции "**Использовать для гиперссылки относительный путь**"). Смысл этой операции заключается в том, что сами рисунки и ссылки на них будут сгенерированы программой автоматически, но в момент просмотра WAP-браузером картинка может не открыться в окне мобильного устройства по причине какого-либо технического несоответствия (например, неподходящий размер). В таком случае принудительный вызов гиперссылки из подрисуночной подписи **Рис. 1.** позволит загрузить рисунок в файловую систему мобильного устройства и ознакомиться с ним отдельно.

К факультативным действиям можно отнести и замену некоторых привычных символов, которые неоднозначно отображаются на экранах мобильных устройств. Одним из таких символов является ' ÷ ' (например, "…в диапазоне 3÷5В."), логично заменить его по всему тексту на обычный дефис ' – '.

На этом же этапе желательно исправить все опечатки, грамматические ошибки и неточности, если таковые замечены. После чего необязательный этап мануальной обработки документа можно считать законченным и приступить к автоматическому преобразованию.

Для автоматической обработки документа и его постраничного разделения нажимаем кнопку включения макроса '**SavePG**' при любом расположении курсора в текстовом поле. Ход процесса отображается на экране и в совокупности не превышает по времени нескольких секунд. Макрос запрашивает уникальное короткое имя для группы ***.wml** файлов, которое удобно набирать с клавиатуры мобильного устройства при **on-line** тестировании. Так, например, для текста лекции логично ввести латинскую букву 'L' и порядковый номер лекции, к примеру – 5. В результате преобразования буду сформированы файлы **L5_1.doc** – **L5_N.doc**, где N – общее число сформированных файлов (не более 99). По

аналогии могут быть названы и другие методические материалы: пособие – 'P', сборник задач – 'Z' и т.д.

Кроме названия программа запрашивает у оператора подтверждение на преобразование всех символов документа в маленькие (тип <small>). Дело в том, что экраны мобильных устройств способны отображать лишь три размера шрифтов: большой, средний и маленький. Естественно, что при выборе маленького шрифта на небольшом экране мобильного устройства можно отобразить б*о*льшее количество символов, но при этом технические, химические символы и различные математические выражения будут лишены индексов (например, **C_2H_5OH** будет выглядеть как C2H5OH). Если такое преобразование не влияет на общий смысл текста (например, смысл фразы "…**резисторы R_1–R_7**…" фактически идентичен эквиваленту "…резисторы R1–R7…"), то на запрос о выборе шрифта <small> следует ответить утвердительно.

По успешному завершению процесса преобразования, когда в рабочей папке наряду с исходным документом сформирована группа новых файлов типа **L5_1.doc – L5_N.doc**, можно приступить к их пакетному конвертированию в формат ***.html**. Разделение процесса на два этапа позволяет просмотреть вновь созданные файлы на предмет различных ошибок, внесенных на этапе конвертирования, и устранить последние. Если таковых не обнаружено, то можно выполнить следующий этап.

б) ***автоматическое пакетное преобразование группы документов MS Word в комплект* *.html *документов.***

Выбрать в меню MS Word опции: <**Файл**> → <**Открыть**>, в окне "**Открытие документа**" выделить все файлы **L5_1.doc – L5_N.doc** в рабочей папке, после чего нажать кнопку '**Открыть**'.

После того, как все файлы **L5_1.doc – L5_N.doc** ***загружены в одно окно***, нажатием кнопки '**SaveHTM**' запускаем макрос преобразования файлов в формат ***.html**. Параметры ***.html** файлов были установлены заранее, но MS Word 97 потребует подтверждения сохранения в кодировке 'Кириллица' для каждого файла. Автоматизировать это подтверждение только средствами VBA практически невозможно, поскольку на время отображения модального окна

запроса все действия текущего процесса приостанавливаются. В ходе процесса конвертации MS Word может выводить окна с предупреждением, что некоторые символы документа могут некорректно отображаться в выбранной кодировке 'Кириллица' и для них следовало бы отдать предпочтение кодовой таблице UTF-8. В ответ на это предупреждение следует нажимать клавишу [**Enter**] или выбирать нажатием мыши кнопку, активную в окне предупреждения по умолчанию – "**OK**". В результате успешного пакетного конвертирования в рабочей папке появится набор файлов с расширением ***.htm** (**L5_1.htm – L5_N.htm**). Документы с гипертекстовой разметкой желательно просмотреть в имеющемся WEB-браузере (Microsoft Internet Explorer, Opera и т.д.) для оценки качества преобразования. Если выявлены какие-либо устранимые недостатки (например, не включен шрифт <small>, заголовок не выделен полужирным шрифтом, название таблицы не выровнено по центру документа и т.д.), следует внести необходимые изменения в исходных файлах **L5_1.doc – L5_N.doc**, удалить все файлы **L5_1.htm – L5_N.htm** и повторить процесс пакетного преобразования. В принципе, не возбраняется преобразовать отдельный файл, в котором обнаружены неточности, но при большом количестве нумерованных файлов в папке это может привести к сбою в нумерации. Если качество документов ***.htm** не вызывает нареканий, можно приступить к следующему этапу – преобразованию их в группу ***.wml** файлов.

в) ***преобразование группы* *.html *документов в* *.wml *формат.***

Процесс преобразования ***.html** документов в ***.wml** формат может быть выполнен в автоматическом режиме без потерь особенностей исходного форматирования весьма условно, несмотря на то, что язык разметки **wml** по сути представляет собой подмножество **html**. Основная причина заключается в значительной скупости изобразительных средств языка **wml**, а также – в более строгом его отношении к своим управляющим конструкциям. Так, если в ***.html** документе встретится последовательность типа:

Примечание: x - не имеет значения.

То она будет интерпретирована следующим образом:

Примечание: x - не имеет значения.

Точно такая же конструкция вызовет ошибку WAP-браузера, в связи с тем, что тэг закрывается раньше прекращения действия тэга . Для ***.wml** документа правильной будет более строгая конструкция:

<**i**>Примечание:</**i**> x - не имеет значения.

В ней корректно соблюдены правила вложения тэгов. В отличие от **html** все (даже одиночные) тэги **wml** парные. К примеру, **html** тэг
 в **wml** будет выглядеть как
 (с признаком завершения его действия). Положение усугубляется также тем, что, несмотря на наличие стандарта спецификации **wml**, различные производители мобильных аппаратов в коммерческих целях вносят в свои разработки некоторые отклонения, что приводит к ситуации, когда WAP-сайт отлично отображается телефоном **Nokia**, но приводит к ошибке при попытке посмотреть его через браузер **Samsung-**а.

Попытки программно обработать нестрогие конструкции **html** при автоматическом преобразовании контента в **wml** приводят к ситуации, когда исходное форматирование может быть полностью или частично утеряно. Так при конвертировании приведенного выше примера

Примечание: x - не имеет значения.

возможна следующая ситуация: программа учла открытие тэгов и ожидает закрытия тэга , но, поскольку ей встречается закрытие тэга , то она отбрасывает его, как непарный, приводящий к ошибке, и просматривает текст дальше; корректно обрабатывает закрытие тэга , а в конце абзаца обнаруживает ошибку – незакрытый тэг , которую устраняет, закрыв его самостоятельно. В результате исходная рассмотренная конструкция превратится в следующую:

Примечание: x - не имеет значения.

и будет отображена WAP-браузером, как:

Примечание: **x - не имеет значения.**

Если после слова ***Примечание:*** следует достаточно большой абзац, то весь он будет отображен полужирным шрифтом из-за ошибки преобразования.

Существующее доступное свободное программное обеспечение для преобразования форматов **html** документов в **wml** решает проблему несоответствия весьма простым способом, отбрасывая элементы сложного форматирования из состава формируемого документа.

Обзор доступных в Сети программ позволил рекомендовать для конвертирования **html** документов в **wml** формат лишь три из них. **WAP Tool** фирмы **Argogroup** – бесплатная программа, пытающаяся наиболее корректно

поддерживать исходное форматирование, но и совершающая заметное количество ошибок в связи с этим. Пакетный режим на поддерживается.

Web2wap 1.1.4 – условно–бесплатная программа (shareware), разработка **China Petroleum University**, поддерживает пакетный режим обработки файлов, обладает неплохим быстродействием. С точки зрения форматирования –старается поддержать выбор шрифтов исходного докуметнов, но весьма фривольно поступает с абзацами и разбиением на строки. Количество ошибок при обработке произвольного документа значительно меньше, чем у программы **WAP Tool**.

HtmlWmlEdit_1_3 (в комплекте с **MobileConverter_1_3_Trial.dll**) – условно–бесплатная программа (shareware), разработка **L.Akerman Software**. Наихудшая поддержка форматирования исходного документа (отбрасываются практически все более или менее сложные элементы), но практически полное отсутствие ошибок в формируемом файле, при обработке произвольного документа. Пакетный режим на поддерживается.

Следует отметить, что оценка количества ошибок, формируемых тестируемыми программами проводилась для файлов, не прошедших никакой предварительной обработки. Документы, сформированные по описываемой здесь методике, обрабатываются всеми тремя программами практически безошибочно, в саязи с чем преимущество имеет утилита **Argogroup's WAP Tool**, для исходных документов, где форматирование имеет значение (например, таблицы или коды программ). **Web2wap 1.1.4** удобна для пакетной обработки большого количества файлов, формат текста для которых не столь уж важен (например, описания приборов, литературные и исторические тексты). **HtmlWmlEdit_1_3** может спасти положение в тех случаях, когда первые две программы генерируют трудно устранимую ошибку (гипотетически очень маловероятная ситуация), а оператор не знаком с основами **wml** и не способен самостоятельно внести необходимые исправления.

Поскольку все рассмотренные программы созданы зарубежными производителями, они имеют общий недостаток – не совсем корректная работа с кодировкой 'Кириллица' (Windows–1251). **Argogroup's WAP Tool** преобразует символы 'Кириллицы' в UTF-8, но, судя по всему, выбирает неверную часть таблицы, похожая ошибка характерна и для **HtmlWmlEdit_1_3**. **Web2wap 1.1.4** не пытается конвертировать все символы 'Кириллицы', а преобразует в UTF-8, лишь отдельные специальные символы, но не прописывает кодовую таблицу Windows–

1251 в заголовок **wml** документа, в связи с чем последний некорректно отображается мобильными WAP-браузерами. Поскольку кодировку Windows–1251 не поддерживается и некоторыми моделями популярных сотовых телефонов имеет смысл предварительно корректно конвертировать документы ***.html** в UTF-8, после чего сформировать **wml** файл одним из перечисленных конверторов.

Поскольку перекодировка в UTF-8 не представляет собой уникального процесса, требующего недюжинной логики, к программе можно предъявить требования удобства интерфейса, самостоятельного принятия решений по большинству возникающих вопросов (размещение конвертируемых файлов, удаление или не удаление исходных документов), режим пакетной обработки.

Всем этим требованиям удовлетворяет бесплатная программа **WapTrans** отечественного производителя **WAP-Сервис**, обеспечивающая быструю и корректную перекодировку групп файлов. Кроме всего перечисленного программа имеет режим непосредствнной конвертации и возможность определения правил перевода. Внешний вид программы изображен на рис. 6.

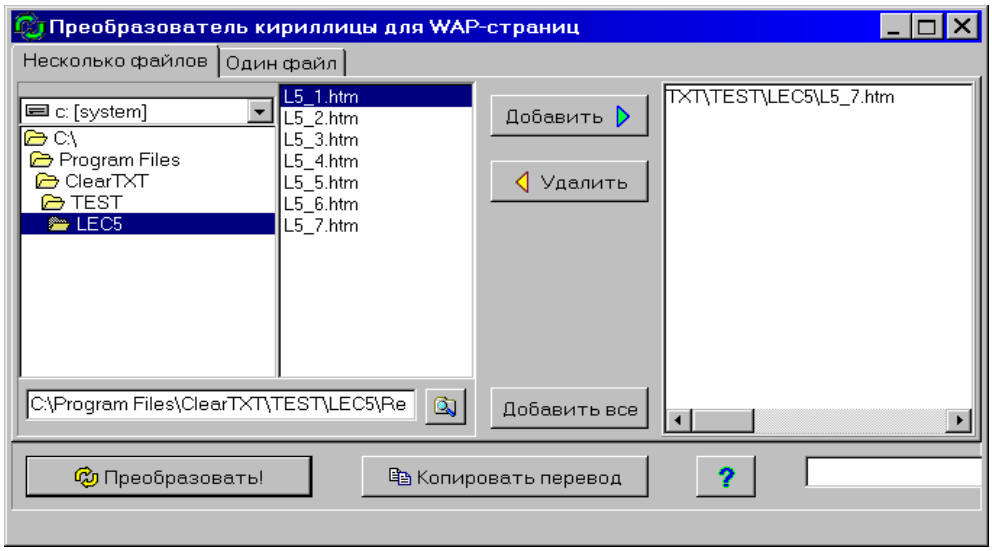


Рис. 6.

В режиме пакетной обработки программа отображает все доступные ***.html** файлы в открытой папке и позволяет добавлять их в список для перевода (справа) все или выборочно. После команды "Преобразовать" выполняется обработка файлов, индицируется статус прогресса обработки текущего файла, обработанные файлы удаляются из списка. По умолчанию программа помещает все

обработанные файлы в папку **Ready**, создаваемую в текущей папке, но место назначение может быть уточнено дополнительно. Исходные файлы по умолчанию не удаляются. Документы ***.html**, сформированные в папке **Ready** , готовы для перекодирования в **wml** формат.

Рассмотрим последовательность процесса преобразования в **wml** формат утилитой **Argogroup's WAP Tool** (Рис. 7.).

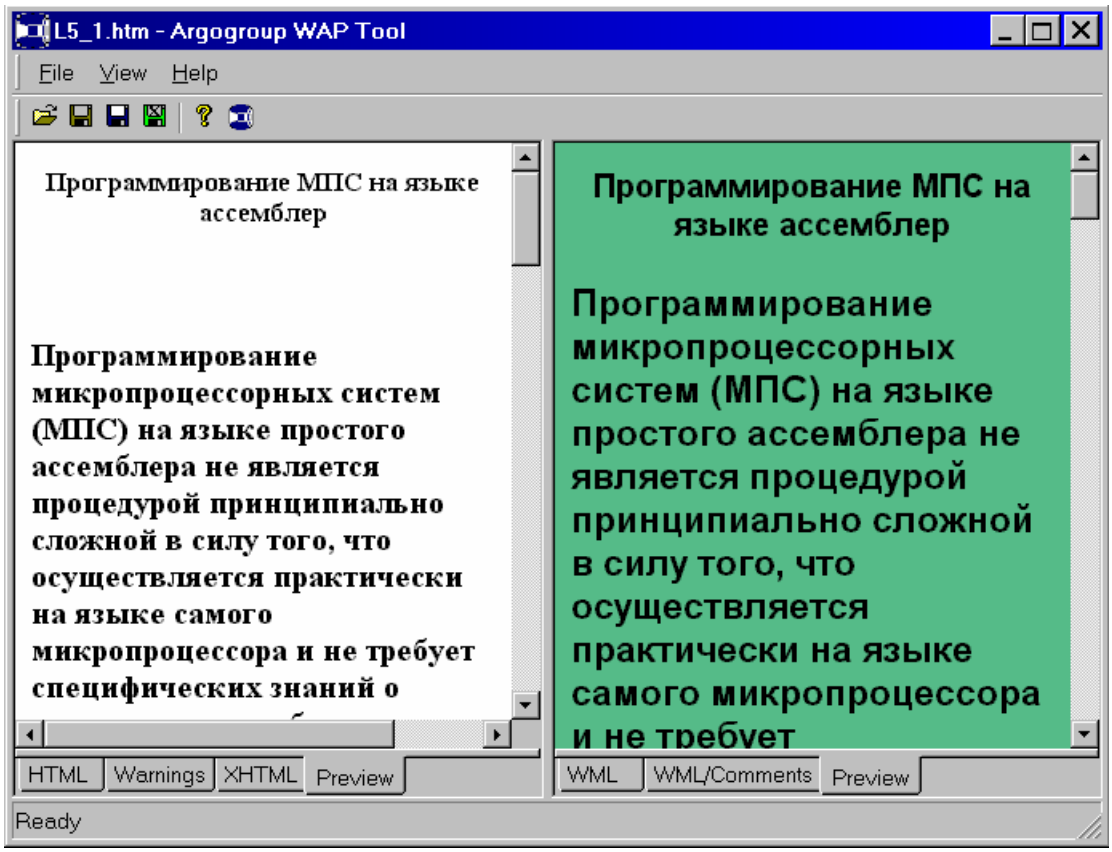


Рис. 7.

Собственно говоря, сама последовательность предельно проста: открыть ***.html** файл (в меню <**File**> → <**Open**>) и сохранить его ***.wml** эквивалент(в меню <**File**> → <**SaveWML**>) с тем же именем. Программа одновременно демонстрирует в разных окнах представление исходного ***.html** документа и его ***.wml**-аналога. Существует возможность просмотра исходных ***.html** и ***.wml** кодов, указаний на встреченные критичные ситуации в обоих документах. Но поскольку программа не позволяет редактировать исходные тексты документов, а вид ***.wml** документа в окне предпросмотра весьма условный, все дополнительные опции носят в значительной мере познавательный характер.

Работа с программой **HtmlWmlEdit_1_3** функционально полностью аналогична: открыть ***.html** файл (в меню <**File**> → <**Open**>) и сохранить его ***.wml** эквивалент(в меню <**File**> → <**Save WML**>) с тем же именем. Схожи и интерфейсы программ, за исключением того, что **HtmlWmlEdit_1_3** демонстрирует только исходные тексты ***.html** и ***.wml** документов, причем в каждый отдельный момент только одну вкладку.

Программа **Web2wap 1.1.4** и вовсе не балует пользователя богатством интерфейса, в том смысле, что совсем не демонстрирует никаких видов документа (Рис. 8.).

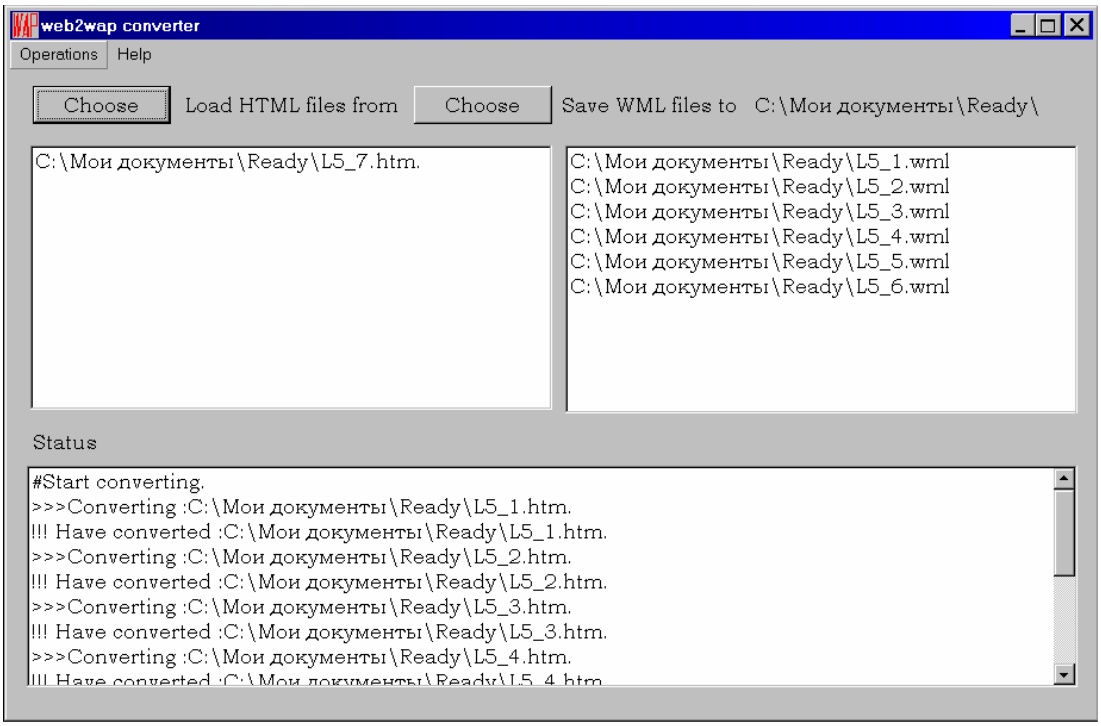


Рис. 8.

Кнопкой "**Choose**" программа предлагает составить список исходных ***.html** документов в окне "**Load HTML files from**", аналогичной кнопкой "**Choose**" выбирается папка для конечных файлов "**Save WML files to**" (по умолчанию '**C:**'). После этого процесс конвертирования запускается опцией <**Start**> в меню <**Operations**>. Ход процесса отображает окно "**Status**", а исходные файлы перемещаются в окно "**Save WML files to**", меняя расширение на **wml.**

Для очистки окон от результатов выполненного процесса и приведения программы в начальную готовность предназначена опция <**Reset**> в меню

<**Operations**>. Опция <**Settings**> того же меню позволяет в некоторых пределах влиять на процесс конвертирования, разрешая или запрещая некоторые условия перекодировки (в рассматриваемом случае значительного влияния на качество конечного файла не оказывающие).

Итак, пакет ***.wml** документов сформирован !

Второй этап: тестирование комплекта документов в формате ***.wml**, разрешение вопросов, связанных с отображением графики WAP-страниц.

а) Off-line ***и*** **on-line** ***тестирование комплекта сформированных*** ***.wml** ***документов, исправление найденных ошибок.***

Для **off-line** (без подключения к Сети) отладки рекомендуется использовать эмуляторы WAP, которые способны скачивать WML-файл как непосредственно с Web-сервера, так и открывать его из локальной файловой системы (как в рассматриваемом случае), и воспроизводить все содержимое WML-контента на дисплее эмулятора мобильного устройства.

Обзор и тестирование доступных в Сети свободно распространяемых бесплатных эмуляторов WAP позволяет рекомендовать для работы с документами формата **wml** следующие программы: **M3Gate** (разработчик **Numeric Algorithm Laboratories**) и **Deck-It** (разработчик **PyWeb.com**).

M3Gate представляет собой WAP-браузер на платформе IBM PC. Он позволяет просматривать страницы в формате WML в **on-line** и **off-line** режимах. M3Gate может работать как автономно, так и совместно с HTML WEB-браузером. Он также может быть интегрирован в среду таких популярных браузеров, как **Microsoft Internet Explorer** и **Netscape Communicator,** придавая им способность к просмотру WAP ресурсов.

M3Gate незаменим для отладки WAP-сайтов, поскольку без лишних настроек позволяет просматривать WML-контент в **off-line** режиме, и очень строго придерживается спецификации WAP, не игнорируя обнаруженные ошибки, как, в частности поступает аналогичная программа **Deck-It**, а индицируя на экране строку и столбец обнаруженной ошибки. WML-документы, протестированные в **off-line** режиме эмулятором **M3Gate,** практически безошибочно отображаются реальными мобильными аппаратами в режиме **on-line**. Программа имеет настраиваемый интерфейс, позволяющий оценить

отображение WML-контента как на экране простых мобильных устройств, так и на дисплеях развитых коммуникаторов (Рис. 9).

Рис. 9.

Работа с эмулятором **M3Gate** предельно проста. При установке программы, она вносит в системный реестр запись, согласно которой все файлы с расширением ***.wml** и ***.wbmp** (формат изображений в WAP) по умолчанию открываются **M3Gate**. Таким образом, для просмотра ***.wml** файла достаточно кликнуть по нему дважды левой кнопкой мыши, как он тут же будет открыт браузером **M3Gate**. Программа поддерживает относительные перекрестные гиперссылки в **off-line** режиме и корректно отображает WAP страницы в кодировке **windows-1251**, а также изображения в формате ***.wbmp**.

К сожалению, программа не обладает качеством визуального **wml–**редактора и в случае обнаружения ошибок использует для просмотра исходного текста **wml–**страницы стандартный Блокнот (Notepad) операционной системы Windows (вызывается кликом правой кнопкой мыши по "поверхности" эмулятора, и выбором опции <**View source**> во всплывающем меню, рис. 10).

Поскольку, как уже было сказано ранее, достаточно трудно корректно и безошибочно конвертировать **html** файл в формат **wml**, следует уделить внимание средствам правки **wml–**страниц в текстовом виде. Нет нужды говорить при этом о необходимости знаний хотя бы простых основ языка разметки **wml** (см. **Приложение 1. Основы WML**). И поскольку простых доступных средств визуальной редакции **wml–**страниц практически не существуют, следует

облегчить процесс работы хотя бы заменой стандартного Блокнота Windows развитым редактором с контекстной подсветкой текстов. В Сети доступно большое количество подобных программ, и, в частности, можно порекомендовать **A Tech Group Professional Notepad,** к достоинствам которого можно отнести автоматическую замену Блокнота, русскоязычный интерфейс и бесплатную регистрацию для граждан бывшего СССР. Сразу следует заметить, что это не единственная программа подобного типа, и, проведя обзор в Сети, каждый может найти для себя наиболее приемлемое решение.

Конечно, можно найти и исправить ошибку и в стандартном Блокноте, но использование специального редактора в значительной мере упростит этот процесс (см. рис. 10).

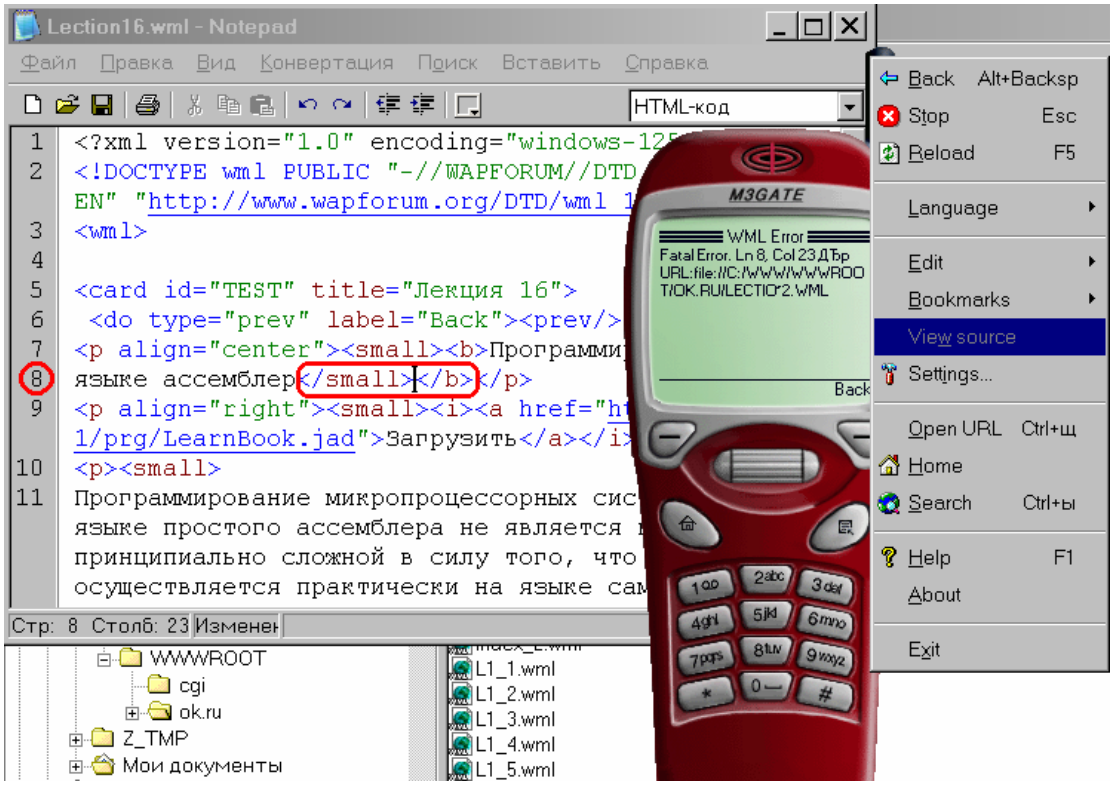


Рис. 10.

Изображенный на рисунке эмулятор индицирует ошибку: **WML Error Fatal Error. Ln 8, Col 23 URL: file://C:/WWW/WWWROOT/OK.RU/LECTIO~2.WML** (неустранимая ошибка в строке 8, колонка 23, файл LECTIO~2.WML). Вызванный опцией меню <**View source**> **Professional Notepad** демонстрирует необходимые строку и столбец, наряду с интеллектуальной контекстной подсветкой текста **wml**–файла. Анализ строки показывает, что при создании ***.wml** документа

нарушена последовательность закрытия тэгов </small> , закрытие следует проводить в обратном порядке: </small>. После исправления ошибки **wml–**файл необходимо сохранить и вновь протестировать в **M3Gate**.

Необходимо подчеркнуть, что поскольку количество **wml–**тэгов не велико (порядка тридцати) и в основном они эквивалентны соответствующим тэгам **html**, а сформированные нами файлы были сгенерированы по стандарту **html** автоматически, устранение в них небольшого количества несложных ошибок вручную представляется задачей сравнительно простой.

WAP-браузер **Deck-It** также позволяет просматривать страницы в формате WML, но, в отличие от **M3Gate,** предпочитает **on-line** режим. Для **off-line** просмотра WAP страниц с помощью **Deck-It** необходимо установить на компьютер локальный Web-сервер, с ролью которого успешно справляется утилита **Small HTTP сервер**, бесплатная для граждан бывшего СНГ (автор *М. Феоктистов*). Программа **Deck-It** эмулирует WAP-браузер мобильного телефона **Nokia** в связи с чем, вероятно, проводимый ей синтаксический анализ **wml–**контента менее строг. По крайней мере некоторые **wml–**страницы, успешно прошедшие тест с помощью **Deck-It,** вызывали ошибку при просмотре в режиме **on-line** реальным мобильным устройством, в то время как страницы, протестированные **M3Gate,** загружались безошибочно.

К достоинствам **Deck-It** следует отнести встроенные средства просмотра исходных текстов **wml** с указанием места и характера встреченной ошибки (рис. 11), а также возможность просмотра **html–**файлов с конвертацией их в **wml–**контент в режиме 'on fly' (на лету). В связке с **Small HTTP сервер***ом* **Deck-It** позволяет провести полноценную отладку структуры разрабатываемого WAP-сайта в **off-line** режиме.

Как недостатки **Deck-It** следует рассматривать невозможность автономной работы в **off-line** режиме, слабую поддержку мыши и полное отсутствие поддержки пользователя контекстной помощью (скупые сопроводительные документы отсылают за описанием и помощью на сайт компании **www.pyweb.com**, где какие-либо следы как помощи, так и самой компании **PyWeb.com** практически отсутствуют).

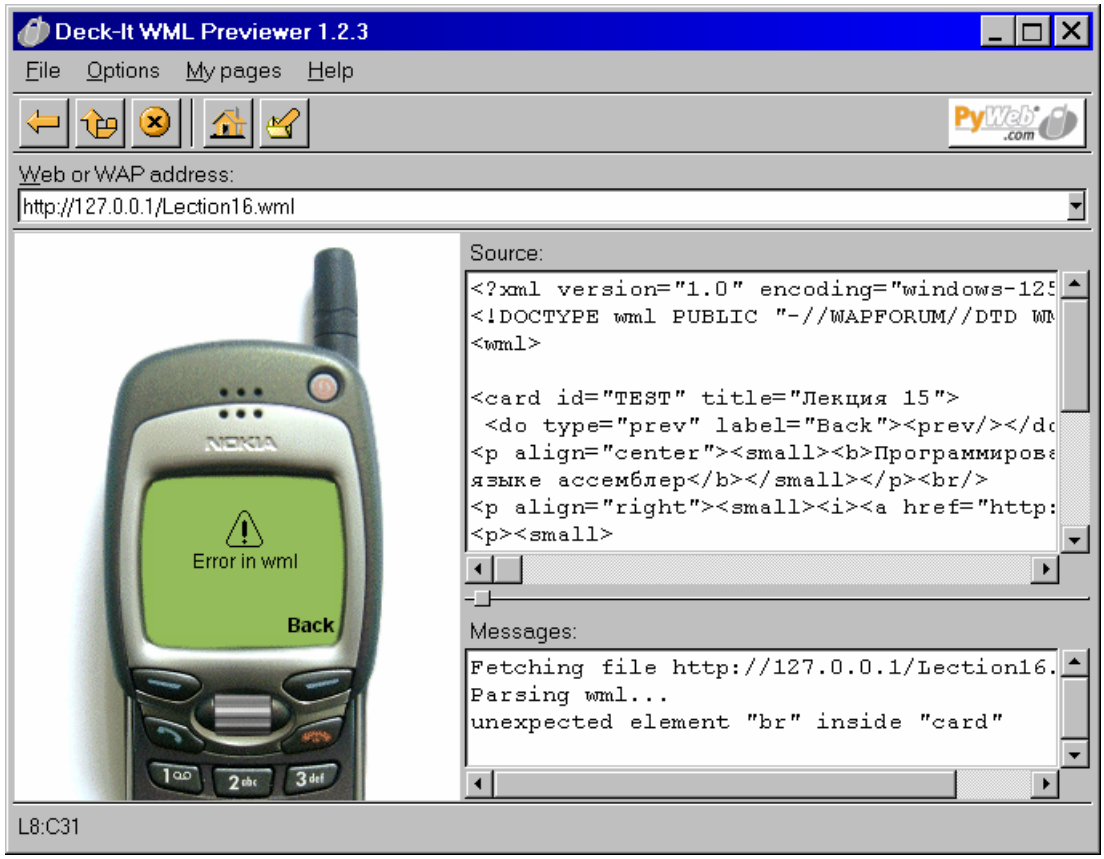


Рис. 11.

После установки и настройки **Small HTTP сервер***а* (сопроводительная документация к которому написана на русском языке и достаточно подробна, а процесс настройки подробно описан в Приложении 2.) локальные **wml**–файлы становятся доступны для просмотра из **Deck-It** по адресам **http://127.0.0.1/имя_файла.wml** . Адрес вводится в верней строке браузера побуквенно с клавиатуры, вставка с помощью мыши на поддерживается, 'горячие' комбинации клавиш (**Ctrl+x**, **Ctrl+c**, **Ctrl+v**) работают только в латинской раскладке клавиатуры и неактивной функции "**CapsLock**". Навигация осуществляется программными кнопками и 'джойстиком' эмулируемого телефона, которые нажимаются курсором мыши, алфавитно-цифровая клавиатура не действует и представляет собой муляж.

Для исправления обнаруженных ошибок необходимо запускать внешний текстовый редактор, к примеру, тот же **Professional Notepad**. Алгоритм анализа и исправления ошибки в этом случае не отличается от аналогичной процедуры при работе с браузером **M3Gate**.

Для создания сайта следует придерживаться определенной структуры расположения **wml–**ресурсов. Типичный пример организации локальных папок с этой целью приведен на рис. 12.

Рис. 12.

Корневая директория **WWWROOT** создается при настройке **Small HTTP сервер***а*. Папки **cgi**, **ok.ru**, **cgi-bin** организуются при его инсталляции. Папка **ok.ru** содержит некоторые служебные файлы, необходимые для работы сервера, и основное меню WAP-страницы, которое обычно организуется в файле **index.wml** (впрочем, он может иметь и другое имя). В этом файле, создать который изначально можно с помощью всё того же MS WORD, обычно содержится тематическое заглавие (например, Учебная дисциплина "Микропроцессорные системы"), авторский копирайт и краткая аннотация (если таковая необходима), а также список всех основных материалов сайта, снабженных относительными гиперссылками (например, **Лекция_1** - L1/L1_1.wml, **Лекция_2** - (L2/L2_1.wml) ... **Пособие_1** - P1/P1_1.wml и т.д.). Сформированные по описываемой методике **wml–**документы располагаются в соответствующих их названиям папках (**L1, L2, … , P1**). Конечно, папки могут иметь и другие имена, но желательно достаточно короткие. Директории **pic** расположенные в этих папках содержат картинки, используемые **wml–**документами каждой конкретной папки. И хотя структура локального сервера достаточно условна, тем не менее, именно в таком порядке **wml–**документы будут расположены на реальном WAP-сервере.

Заметим, что развитые мобильные устройства (например, наладонники или коммуникаторы) способны просматривать собственными программными средствами обычные **html–**документы. И комплекты таких документов были нами созданы на промежуточном этапе создания **wml–**файлов. Логично, что они вполне могут быть использованы для создания **html–**зеркала **wml–**сайта.

После того, как все **wml–**документы организованы в указанном порядке, можно протестировать их работу браузерами **M3Gate** и **Deck-It** на предмет корректной работы перекрестных ссылок и правильность навигации. После проверки и исправления ошибок, созданные файлы практически готовы к публикации в WEB, но для этого следует разрешить следующий вопрос.

б) ***разрешение проблем, связанных с корректным отображением графики WAP-страниц.***

Основная проблема с отображением графического контента **wml–**страниц связана с размерами экрана мобильных устройств. Исходя из чисто эргономических соображений, мобильный телефон не способен соперничать в этом вопросе с настольным или портативным ПК. Дисплей мобильного устройства попросту не может вместить красочные рисунки большого размера, поскольку изначально не был для этого предназначен. Строгость языка разметки **wml** связана в основном с тем, что при сравнительно скромной вычислительной мощности аппаратных средств, функции декомпрессии **wml–**потока выполняются аппаратно с помощью специальной микросхемы, в функции которой весьма трудно ввести обработку и масштабирование широкого круга форматов изображений.

Другим ограничивающим фактором является то, что изначально экраны мобильных устройств были преимущественно монохромными, что, вероятно, и предопределило стандарт спецификации WAP версии 1.1 для изображений. Графику рекомендовано создавать в специальном формате, называемом WBMP, которой определяет монохромную картинку, геометрические размеры которой (в пикселях) не должны превышать разрешения экрана используемого для просмотра мобильного устройства.

Таким образом, если строго придерживаться спецификации, все изображения на WAP-странце должны быть монохромными, без дополнительных градаций яркости, и размер их не должен быть больше чем 101x60 пикселей. Согласимся, что вместить в такие границы сколько–либо содержательную иллюстрация весьма затруднительно.

Современные мобильные устройства, безусловно, обладают как б***о***льшими размерами дисплеев (минимальным можно считать, по крайней мере, – 128x128 точек), так и возможностью отображать цветную графику, но в существующей спецификации WAP версии 1.1 это пока никак не отражено.

Для принятия компромиссного решения примем за основу, что изображение должно быть монохромным и рассмотрим варианты, как оно будет отображаться на экранах с различным разрешением. С этой целью к размерам 128x128 была

адаптирована иллюстрация реального документа, и корректность её аппаратного отображения была проверена на мобильном телефоне **Samsung**, с геометрией экрана 128x160 пикселей (рис. 13 а).

а) **б)** **в)**

Рис. 13.

Естественно, что на экране коммуникатора (рис. 13 в), геометрические размеры дисплея которого больше, изображение не претерпело никаких искажений. На экране телефона, геометрические размеры которого меньше по вертикали (рис. 13 б), картинка показана частично, но полоса прокрутки обозначила возможность просмотра изображения, используя кнопки джойстика ('вверх'–'вниз').

Следует подчеркнуть, что способность мобильного устройства к отображению графики обусловлена его как аппаратными, так и программными возможностями. Так конструкции фирмы **Nokia** зачастую снабжаются высокотехнологичными аппаратными решениями, которые фактически не соответствуют существующему стандарту, но предоставляют пользователю дополнительные удобства, и, в конечном итоге, находят поддержку у других производителей. В этом аспекте представляет интерес, как мобильные устройства

отображают картинку, размеры которой заведомо превышают геометрию их экрана (рис. 14).

а) **б)** **в)**

Рис. 14.

Телефоны со скромной аппаратной поддержкой картинку не отображают (рис. 14 а), выводя на экран только альтернативный текст, если он был предусмотрен при создании документа (**alt="diagram"**) и подсвечивают как гиперссылку подрисуночную подпись (если она была ранее оформлена как гиперссылка). Телефоны с развитой аппаратной поддержкой способны масштабировать изображение (рис. 14 б), а также осуществлять 'скроллинг' (прокрутку) по площади картинки (рис. 14 в), фактически просматривая её через окошко в размер собственного экрана.

Чтобы найти компромиссное решение, удовлетворяющее большинству моделей мобильных устройств, разберёмся с вопросом, насколько изображение теряет в информативности при сжатии его до размеров, подходящих для индикации на экранах мобильных устройств.

С этой целью было выбрано типичное насыщенное изображение принципиальной электрической схемы узла микропроцессорной системы (рис. 15 а), имевшее изначальный размер 852x798 пикселей, полноцветное (на рисунке уменьшено средствами MS Word для экономии места). Изображение было преобразовано к размеру 128x128 (рис. 15 б) и трансформировано в формат WBMP (рис. 15 г). Это же изображение было конвертировано в формат ***.jpg** и уменьшено до размеров 640x480, приемлемых для загрузки в упоминавшийся ранее телефон модели **Samsung** (рис. 15 в).

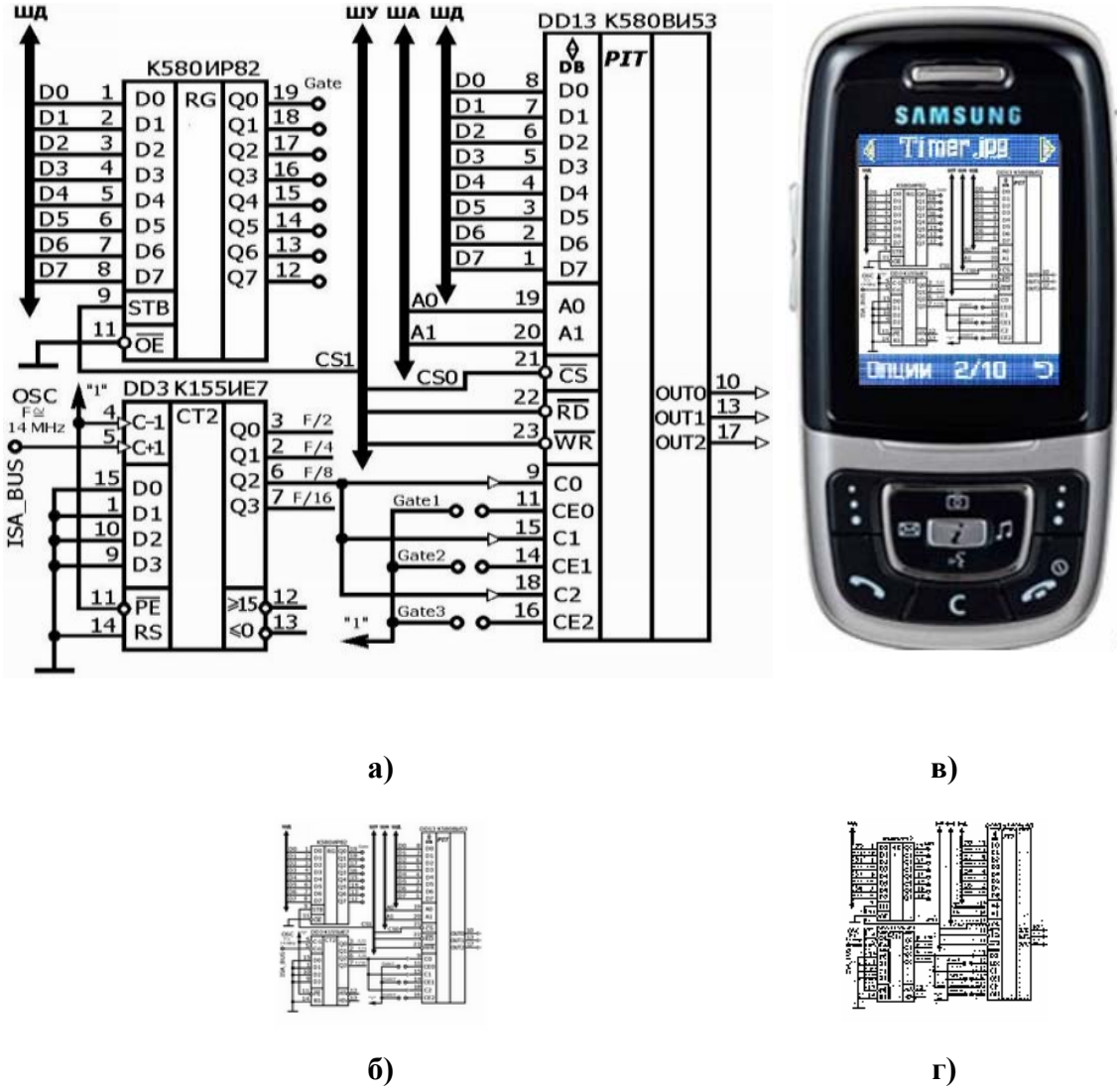


Рис. 15.

Анализ изображенных рисунков свидетельствует о том, что при сжатии полноцветной иллюстрации в маленький формат с полутонами картинка еще сохраняет некоторую информативность (рис. 15 б), но полученное из неё монохромное изображение WBMP можно рассматривать скорее как пиктограмму (рис. 15 г), логически обозначающую место иллюстрации в тексте.

Подытоживая проведенный анализ, можно рекомендовать следующий алгоритм, подходящий для большинства моделей мобильных устройств. Иллюстрации, сформированные средствами MS Word при создании комплекта документов **html,** конвертируются в картинки формата WBMP размером 128x128 пикселей. Как показывает рис. 14 б, б***о***льшее сжатие лишено всякого практического смысла, поскольку лишает информативности даже изображения специально адаптированные для маленького размера.

В **wml–**документах оформляются ссылки на WBMP-файлы следующим образом: **** , где **pic/graph.wbmp** относительная ссылка на файл рисунка в папке **pic** той директории, где размещен соответствующий документ; **diagram** – надпись, которая будет выведена на экран, в том случае, если отобразить иллюстрацию невозможно.

Поскольку на момент создания **html** документов у нас не было файлов иллюстраций (MS Word создает их в формате***.gif** во время сохранения создаваемого файла), ссылки на соответствующие файлы необходимо исправить в **wml–**документах вручную, задав поиск по тексту строки **.gif** и заменив найденные элементы на **.wbmp**. Здесь же следует проверить корректность самой ссылки, поскольку для документов, содержащих одну иллюстрацию создаёт её в той же папке, что и сам документ, если же иллюстраций несколько, для них создаётся отдельная папка. Все ссылки на иллюстрации необходимо привести в виду **pic/имя_картинки.wbmp**. Осуществить эту операцию автоматически не представлялось возможным, поскольку MS Word не позволяет изменять гиперссылки на рисунки в тексте даже вручную.

Исходные файлы картинок в формате ***.gif** следует также конвертировать в формат ***.jpg** размером 640x480 и поместить их в ту же папку **pic**. На эти файлы будут указывать ссылки из подрисуночной подписи, которые были нами заранее сформированы в виде: **Рис. 1.**. Такие ссылки необходимо преобразовать к виду **Рис. 1.**, что не представляет трудности, поскольку эти ссылки обычно расположены по тексту сразу после ссылок на картинки.

Таким образом, если картинка формата ***.wbmp** не отображается WAP-браузером мобильного устройства, или изображение слишком низкого качества, пользователь всегда имеет возможность скачать качественное ***.jpg** изображение по ссылке из подрисуночной подписи и рассмотреть его отдельно. Формат **.jpg** поддерживается большинством моделей мобильных устройств для обмена графическими данными, к тому же он обеспечивает хорошее сжатие изображения, что резко снижает время, необходимое для скачивания (так объём файла приведенной ранее схемы исходным размером 852x798 пикселей, после конвертирования в формат ***.jpg** 640x480 составил всего 78.5 КБайт).

Для различных преобразований графических форматов во всех современных версиях популярных графических пакетов, таких как **Adobe PhotoShop**,

Corel DRAW и др., существуют фильтры, позволяющие осуществлять изменение формата рисунков. Из отдельных программ специально предназначенных для этих целей наиболее популярной является программа **ACDSee**, но в последнее время она приобрела статус условно-бесплатной, что накладывает ограничение на возможность её использования. Хорошими возможностями обладает свободно распространяемая бесплатная программа **XnView для Windows** (Copyright 1991-2002 Pierre-e Gougelet). Программа имеет русскоязычный интерфейс, доступна для скачивания из Сети, и способна как преобразовывать подавляющее большинство форматов файлов, так и изменять прочие атрибуты графики: размеры, насыщенность, цветность, четкость изображения и др. Немаловажным достоинством полной версии программы является возможность пакетной обработки графических файлов. Из других доступных программ похожими, но более скромными возможностями обладает **Microsoft Photo Editor** из пакета **Microsoft Office**, и если Вы являетесь счастливым обладателем этого лицензионного пакета, то, возможно, другой конвертор Вам и не понадобится, хотя опции групповой обработки файлов **Photo Editor** не предлагает.

Не рекомендуется использовать различные утилиты типа **ALL2WBMP** или **PIC2WBMP**, поскольку в силу свойственного им жесткого алгоритма, создаваемое ими изображение чаще всего посредственного качества (см. рис. 14 б), тем более, что большинство таких утилит не бесплатны.

Русскоязычный интерфейс программы **XnView** интуитивно понятен (рис. 16). В процессе установки программа принимает на себя функции просмотра файлов большинства графических форматов, позволяя оставить обработку файлов, зарегистрированных в системе за закрепленными приложениями.

Открытие графического файла осуществляется двойным кликом левой кнопкой мыши по его имени, либо через всплывающее меню при клике правой кнопкой. Если программа **XnView** уже запущена, порядок открытия файла стандартный: опция меню <**Файл**> → <**Открыть**>. Для преобразования файла в другой формат достаточно выполнить последовательность: <**Файл**> → <**Сохранить как…**>; в окне "**Сохранить рисунок**" выбрать необходимый "**Тип файла**", после чего нажать кнопку "**Сохранить**" (рис. 16).

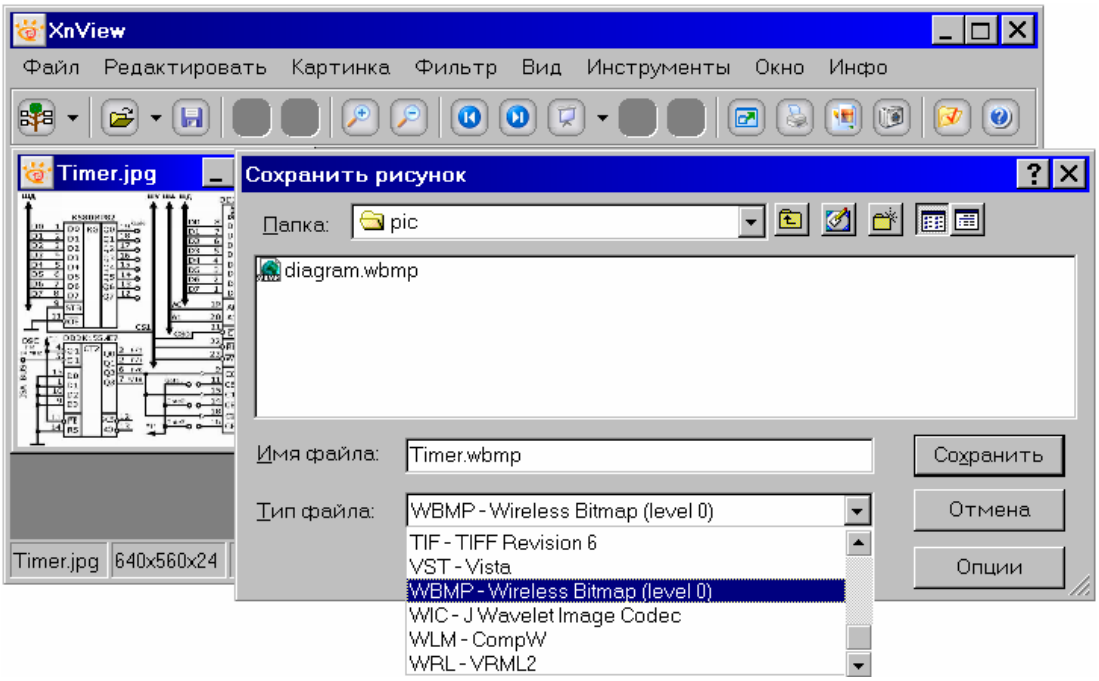


Рис. 16.

Программа имеет огромное количество опций для работы с графическими изображениями, описывать которые в данной методике было бы излишним.

Остановимся на алгоритме, позволяющем получить довольно качественное изображение в формате **wbmp**. Исходный файл должен быть полноцветным (24 бит), даже если это чертеж или схема в черно-белом варианте. При преобразовании полноцветного изображения **XnView** имеет возможность при сжатии использовать полутона, что повышает разборчивость конечного рисунка. Если исходный файл монохромный его следует конвертировать в полноцветный опциями <**Картинка**> → <**Преобразовать в цветное**> (или <**Картинка**> → <**Преобразовать в серое**>), выбрав далее максимальное число цветов или градаций серого.

Для конвертирования исходного изображения в картинку формата **wbmp** размера 128x128 пикселей достаточно изменить его размер (<**Картинка**> → <**Изменить размер**>), выставив параметр "**Ширина**" 128 пиксел и активировав пункт "**Сохранять пропорции**" (рис. 17). После чего сохранить картинку как WBMP (рис. 16). Но более качественный результат может дать иная последовательность действий.

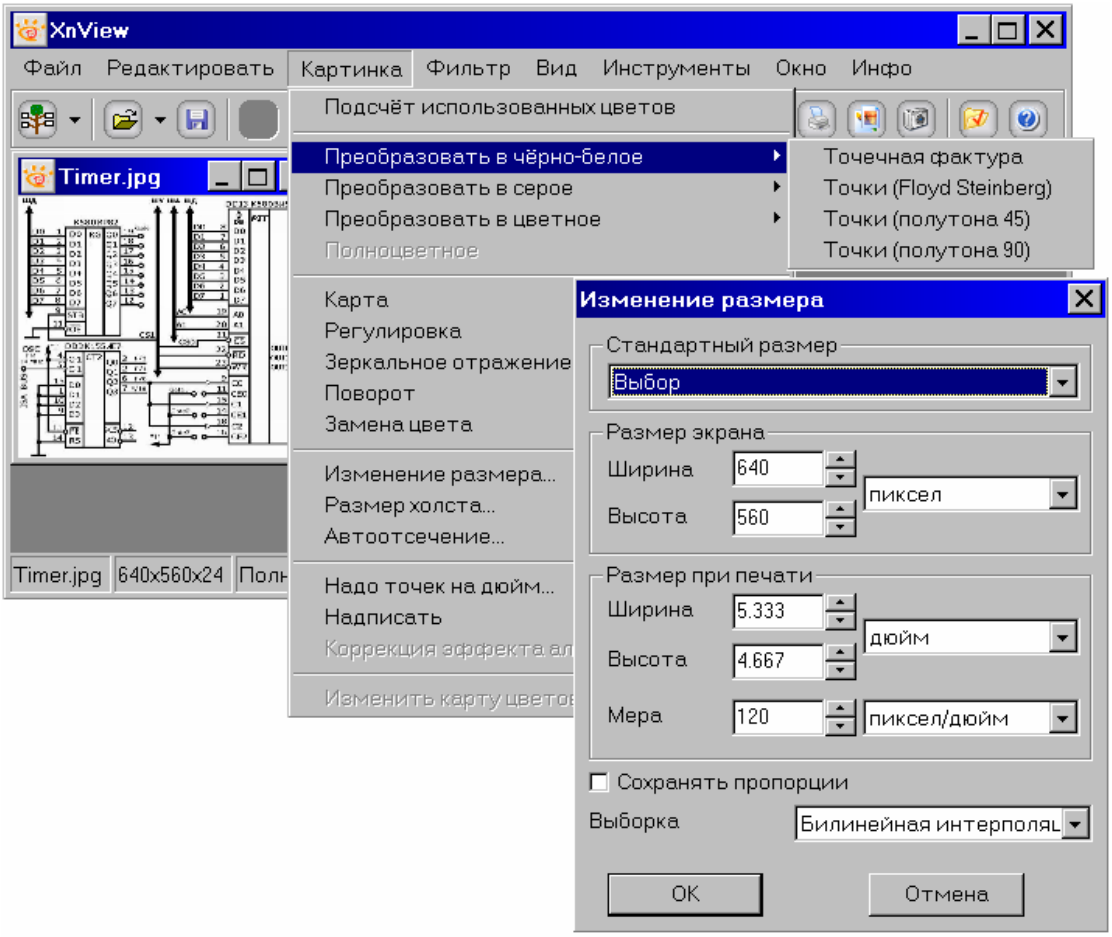


Рис. 17.

Исходное изображение, сохраняя пропорции, уменьшить до ширины 256 пиксел, после чего в меню <**Фильтр**> выбрать пункт <**Деталировка**>, где последовательно провести над рисунком операции "**Проработка деталей**" и "**Проработка краёв**". После того, как эти операции выполнены, часть полутоновых деталей рисунка должны стать более контрастными. Полученное изображение, сохраняя пропорции, уменьшить до ширины 128 пиксел, и преобразовать его в черно-белое (что, собственно, и происходило бы по умолчанию при конвертировании в формат WBMP). **XnView** предложит методы преобразования (рис. 17), из которых довольно неплохой результат даёт алгоритм Floyd Steinberg (хотя можно попробовать и другие). Конечный результат следует сохранить как WBMP (рис. 16).

Возможно, данный алгоритм не является оптимальным для любого изображения, но после проведения каждой операции результат её воздействия очевиден на экране и может быть отменен. Комбинируя различные фильтры и

методы обработки картинки, в программе **XnView** всегда можно добиться вполне приемлемого качества итогового рисунка.

К сожалению, предложенная методика создания комплекта методических **wml–**документов для публикации на WAP-сайте довольно далека от той степени автоматизации, о которой поётся в песне известной в недавнем прошлом группы: "Нажми на кнопку - получишь результат…". Но, следует отметить, что и создание обычного HTML WEB-сайта является процессом в значительной мере творческим. Тем более, что даже простое и, казалось бы, полностью отработанное преобразование документа Word в документ формата HTML происходит не 100%–но идентично. В данном же случае мы имеем дело с кросс–платформенным преобразованием документов для микропроцессорного устройства с ограниченной производительностью, причем сами документы представляют собой не сиюминутные новости или рекламу, а учебные и методические материалы качество и достоверность которых вполне оправдывают потраченные усилия и затраты.

Заключение

В заключение хотелось бы отметить, что если по ряду причин описанная здесь методика представляется громоздкой и не вполне удобной, существует и более простое решение, требующее значительно меньших трудозатрат, хотя и не обеспечивающее высокого качества. Таким решением являются фильтры HTML–контента непосредственно на WEB-сервере, при обращении к нему WAP-браузером. Примером такого фильтра является сервлет **h2wPackage.jar**, который устанавливается на WEB-сервере и требует его специальной настройки, после чего фильтр способен преобразовать на лету контент HTML в WML формат. Похожими возможностями обладает **Mobile Converter**, работа которого основана на интеграции в работу сервера уже упоминавшейся ранее **MobileConverter_1_3_Trial.dll** в качестве фильтра. Оба фильтра не бесплатны, а о качестве преобразования можно судить хотя бы по работе описанной ранее программы **HtmlWmlEdit_1_3,** функционирующей в комплексе с данной динамической библиотекой.

И, наконец, следует упомянуть ещё один весьма неоднозначный способ WAP-публикации учебных документов, который подробно описан в **Приложении 3** – Подготовка методических материалов в формате JAVA-книг. JAVA-книги или электронные книги (e-book) в настоящий момент являются довольно популярным носителем для чтения художественной литературы, а учебные и методические тексты не отличаются от последней какой либо спецификой, не позволяющей использовать указанный формат

При разработке представленной здесь методики создания электронных документов в форматах, приспособленных для чтения с помощью мобильных коммуникационных устройств, преследовалась цель создания достаточно простого алгоритма, не требующего значительных финансовых затрат, основанного на программном обеспечении, доступном и распространяемом свободно (“freeware”) и условно–бесплатно (“shareware”). Причем условно–бесплатные программные продукты использовались лишь в случаях, когда характерные для них ограничения не становились препятствием для их продуктивного использования.

Основываясь на предложенной методике, даже учебные заведения, не имеющие собственных WEB-ресурсов, способны создать эффективную и доступную сеть распространения учебных материалов, опираясь на бесплатные серверы доступные в Сети. По мнению автора предложенной методики, материалы, доступ к которым осуществляется с применением столь популярных современных, широко–доступных и, можно сказать, “модных” технологий, будут более востребованы учащимися.

Б***о***льшая часть программного обеспечения, используемого в методике, находится в Сети в свободном доступе. Программы, распространяемые авторами и создателями бесплатно, а также условно–бесплатно, не предназначены для коммерческого распространения и собраны на прилагаемом к методике компакт–диске лишь для удобства тестирования.

Все упомянутые в тексте названия фирм и программных продуктов являются собственностью их владельцев.

Список литературы.

1. Краснов А. WAP — новая технология сетевой коммуникации // “Игромания” №4/67 2003.

2. Кустов А. WAP. Весь мир в вашем телефоне // http://ad.adriver.ru/top-beauty_im/wap.htm .

3. Семёнов А.А. Интернет-технологии как средство обучения и мониторинга качества знаний в учебном процессе ВУЗа // Методическое обеспечение профессионального саморазвития ВУЗов и ССУЗов.

4. Садоян Р. Язык WML // http://prosto.pp.ru/Docum/DocumShow.asp? DocumID=106.

5. Andy Taler Введение в WML // http://www.inode.ru/articles/xml/2006-02-17/304.

6. WML (Wireless Markup Language) // http://www.ua-admin.com/ru/mobilend/primary.php? only_general=1.

7. Мобильный Интернет, или WAP-сайт своими руками // http://kek.ksu.ru /EOS/Tests/WAP/1_M_FirstSteps.html .

8. Алексей Доля Собственный WAP-сайт? Нет проблем! // Мобильные Компьютеры, № 021, стр. 21-54.

9. Легченков А. Как сделать WAP. Основы языка WML // http://www.sitepoint.com/print/351 .

10. Что такое WAP? // http://www.javaportal.ru/mobiljava/wapwmlandother/wap.html .

11. Paul Howard Converting HTML to WML // http://www.topxml.com/wap/articles/htmlwml/default.asp .

12. WAP редакторы // http://For-Xakep.narod.ru.

13. Пишем на WML // http://bs.yandex.ru/resource/narod/40.htm .

14. WAP эмулятор Deckit // http://mirsofta.ru/index.php?id=1058021775.

15. WAP Tool Support // http://www.argogroup.com/software/.

16. Small HTTP server for Win32 // http://www.wplus.net/pp/mrdoors/srv/ .

17. Txt2Bmp - APf72 Home Page // http://www.apf72.narod.ru/txt2bmp.htm .

ПРИЛОЖЕНИЕ 1.

Основы WML

WML (**Wireless Markup Language**) это сетевой язык для создания WAP-сайтов для мобильных устройств; дословно: язык разметки для беспроводных систем. Официальная спецификация WML разработана и поддерживается **WAP Forum**, производственным консорциумом, основанным **Nokia**, **Phone.com**, **Motorola** и **Ericsson**. Эта спецификация определяет синтаксис, переменные и элементы, используемые в файлах WML.

WAP (**Wireless Application Protocol**) – это протокол для беспроводных приложений, позволяющий просматривать через Сеть специально созданные сайты на экранах мобильных устройств. WAP-страницы очень похожи на обычные HTML-страницы. Лежащий в их основе язык WML представляет собой, по сути, некоторое подмножество HTML. В языке WML используются тэги, так же как в HTML, только синтаксис WML строже и должен полностью соответствовать стандарту XML 1.0. Под тэгами (tag – признак, маркер, метка), как и в HTML, понимаются специальные ключевые слова, заключенные в угловые скобки **<…>**. В WML все тэги парные (** **–открывающий…** **–закрывающий), одиночные тэги закрываются сразу: **
**.

WML скуп на возможности представления: набор поддерживаемых тэгов весьма невелик, и язык основан на нетипичной метафоре колоды карт, являющихся краеугольным камнем WML-дизайна и представляющих собой набор показываемых по отдельности карточек (экранов), хранящихся в памяти. То есть, если для обычного сайта единицей является **Web-page**, то единица WML – это т.н. **deck** (колода, хотя встречаются термины 'палуба', 'папка' и 'дека'), состоящая из одной или нескольких **card** ('карта'), составляющих вместе WML-документ (wml-файл).

WML был разработан для устройств с низкой пропускной способностью (мобильные телефоны передают данные со скоростью 14,4 КБайт/с) и маленьким дисплеем. Когда сотовый телефон загружает WML–страницу, загружаются вся колода с картами. Переход по ссылкам внутри страницы (между картами) осуществляется быстро, без обращения к серверу.

WML, в основном, описывает текст и ссылки. Использование таблиц и графики весьма ограничено. Так как WML соответствует XML стандарту, то имеет значение регистр букв. Это значит, что <wml> и <WML> - разные тэги. Другой важный момент – это жесткое требование, что каждый тэг должен закрываться (аналогично XML).

Структура типичного WML-файла выглядит следующим образом:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>

<card id="TEST1" title="HELLO!">
<p>
Hello world!<br/>
</p>
</card>

<card id="TEST2" title="BYE!">
<p>
Do_svidanya!<br/>
</p>
</card>

</wml>
```

Самая первая фраза внутри любого XML–документа называется пролог. Поскольку пролог стандартен, он содержит две строчки кода: определение версии XML и DTD (указатель на файл, содержащий DTD – Document Type Definition, т.е. определение типа документа для нашего документа).

Следом за прологом, в каждом XML-документе содержится элемент, который содержит в себе остальные подэлементы. Этими элементами являются угловые скобки: **< >** и **</ >**. В документе должен содержаться только один элемент описывающий сам документ. В WML этим элементом является **<wml> </wml>**. Все остальные элементы содержатся уже внутри него.

Нетрудно заметить, что такая структура во многом родственна HTML: тэги **<wml>...</wml>** начинают и завершают документ, а тело карт заключается в **<card>...</card>**. Абзацы разделяются тэгом ** ... **, весть текст в WML должен содержаться внутри этих тэгов; **
** - перевод строки. **Русский текст** можно включать в **wml**-документы в кодировках **Windows-1251** и (предпочтительно) **UNICODE UTF-8**.

Не разрешается вставлять тэги ** ... ** друг в друга. На экран сотового телефона одновременно выводится только один документ.

Прежде чем дать краткий обзор основных элементов языка необходимо напомнить, что в отличие от HTML, где браузер отображает страницы даже с неправильным кодом, WML браузер отвергнет страницу с кодом, в котором есть ошибка.

Строительные блоки и функциональные средства WML можно условно разделить на четыре категории: форматирование, навигация, ввод данных и управление действиями/событиями.

Форматирование

Выравнивание текста.

Выравнивание текста в WML практически такое же, как и в HTML. Все выравнивания должны быть сделаны, используя следующие тэги **<p>:**

По центру:
```
<p align="center">
```
Справа:
```
<p align="right">
```
Слева:
```
<p align="left">
```

Форматирование шрифта

наклонный

жирный

жирный

наклонный

<u>подчеркнутый</u>

<big>увеличенный шрифт</big>

<small>уменьшенный шрифт</small>

Таблицы

```
<table columns="3"><!--- комментарий: таблицы аналогичны HTML --->
<tr>
<td>Cell 1</td><!--- комментарий: это первая ячейка --->
<td>Cell 2</td><!--- комментарий: это вторая ячейка --->
<td>Cell 3</td><!--- комментарий: это третья ячейка --->
</tr>
</table><!--- комментарий: ...такой же как в HTML ☺ --->
```

Изображения

```
<p>
Это картинка
<img src="/img/ball.wbmp" alt="Шарик" />
</p>
```

WAP поддерживает только изображения в специальном формате .wbmp;

img src="/ img/ball.wbmp" - ссылка на файл изображения;

alt="Шарик" - альтернативный текст, обычно выводится на экран, если картинку отобразить не удалось.

Навигация

Для навигации и установки гиперссылок предусмотрены тэги **<anchor>** и **<a>**. В элементе **anchor** используется либо подэлемент **go** с атрибутом **href**, задающим адрес ссылки, либо **prev** - вернуться к предыдущему экрану.

Ссылки.

Есть три способа ссылки на другой WAP сайт, файл или карту. Как и в HTML это может быть прямой путь к файлу, либо относительная ссылка. Для ссылки на другой WAP сайт используйте следующее:

```
<a href="http://gowansnet.waphosts.net/">Gowansnet</a>
```

Для ссылки на файл на вашем сайте используйте:

```
<a href="links.wml">My Links</a>
```

и для ссылки на другую карту в этой же странице:

```
<a href="#about">About Me</a>
```

```
<a href="test.wml">Дальше</a>:
<anchor>
Перейти на страницу Тест
<go href="test.wml"/>
</anchor>
```

В примерах выше ссылки устанавливаются при помощи традиционного тэга **<a>** и 'якоря' **anchor** и. У **anchor** два атрибута: "**href**" определяет объект, на который мы ссылаемся, и "**title**" - экранная надпись, идентифицирующая связь, которую браузер может опционально показывать.

```
<anchor>
Предыдущая страница
<prev/>
</anchor>
```

Элементы ввода

Формы
Текстовое поле:

```
<p>Имя: <input name="name" size="15"/><br/></p>
```

Выбор из одного пункта списка:

```
<select>
<option value="male">мужской</option>
<option value="female">женский</option>
</select>
```

Выбор одного или нескольких пунктов из списка:

```
<select multiple="true">
<option value="v1">Soup</option>
<option value="v2">Meat</option>
<option value="v3">Tea</option>
</select>
```

Управление действиями

"Действия" (**tasks, do**) используются для структур типа меню и для описания отдельных действий, например, для выбора новой карты/ссылки. Типовые "исполнительные" тэги действий - это **go, prev, refresh** (перечитать текущую карту) или **noop** (ничего не делать). Действия на уровне конкретной карты описываются в тэгах **do,** а на уровне всей колоды - в элементе **template.**

При помощи тэга **timer** и атрибута **ontimer** тэга **card** можно, например, на время 'заморозить' страницу. На практике часто применяется **ontimer="url"**, говорящий, что некое время карта должна отображаться, а затем должен загрузиться ресурс с адреса **url**. Время в **value** выражается в 0,1 сек.

```
<wml>
<card id="variable" title="Variable"
ontimer="WMLScript.wml">
 <timer value="5"/>
```

```
 <p align="left"> Wait for 5 seconds ... </p>
</card>

<!-- Card and deck linking -->
<card id="input1" title="Input">
<do type="Next input properties">
  <go href="#input2"/> </do>
<do type="Task properties">
  <go href="Task.wml"/> </do>
  ...
</card>
</wml>
```

В WML есть также возможность назначать клавиши навигации. Тэг **<onevent type="button"><действие></onevent>** позволяет приписать к одной из кнопок телефона (зависит от модели) пользовательское событие. В качестве действия обычно используется **<go href="url"/>**. Так, добавив к карте строку **<onevent type= "GO"><go href="#card2"/></onevent>**, мы назначим кнопке **GO** переход непосредственно в меню.

Вариант

```
<do type="accept" label="Search">
<go href="table.asp?srch=$varsearch"/>
</do>
```

иллюстрирует другой вариант переназначения стандартного меню телефона и способ передачи переменных в сервер–ориентированные скрипты. Значение параметра **type** показывает, какая именно клавиша будет переопределена (в данном случае кнопка, соответствующая **OK**). Префикс $ перед именем переменной говорит о том, что будет использовано значение переменной, а не само слово "**varsearch**".

Приведенное выше краткое описание основных конструкций и управляющих элементов языка WML не претендует на справочную полноту, а выполняет, скорее, функцию быстрой подсказки при поиске ошибок в WML-документах.

ПРИЛОЖЕНИЕ 2.

Настройка Small HTTP сервера

Small HTTP сервер – это небольшая утилита, которая превращает Ваш компьютер в полностью функциональный Web-сервер. Сама программа требует минимального комплекта системных ресурсов, так что функционирование сервера не влияет на производительность вашего компьютера. Этот сервер может функционировать под LAN сетями или даже под Dial-Up сетью. Веб-мастера могут использовать эту утилиту в их локальных компьютерах для отладки их CGI-сценариев, не переходя в онлайн.

Установить сервер, в принципе, довольно просто, выполнив программу **sethttp.exe** (**shttp2.exe** или **shttp3.exe**) и введя в процессе установки основные настройки.

Предположим, что мы устанавливаем сервер на диск C:\. Создадим для этого следующие папки: **C:\WWW**, **C:\WWW**MyServer, **C:\WWW\WWWROOT** и **C:\WWW\WWWROOT\ok.ru**. Файл конфигурации **shttp2.exe** разместим в каталоге **C:\WWW**MyServer и запустим его.

В окне настройки на вкладке **General** укажем следующие параметры:

- каталог **C:\WWW\Small_server**, где находится исполняемый модуль web-сервера (**http.exe**);
- имя файла для записи протокола: **http.log** (факультативно);
- строку "**Гражданин бывшего СНГ**" в качестве значения поля **Name** и
- число **30 (24-26)** в качестве кода.

На вкладке **HTTP** укажем следующие параметры:

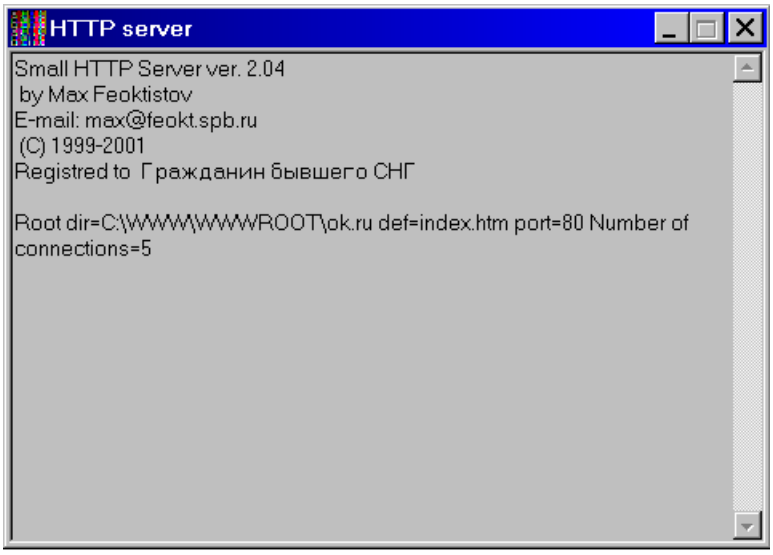
- **Default file: index.htm** (по умолчанию, или другое имя, см.далее);
- **корневой каталог C:\WWW\WWWROOT\ok.ru**: (в этом каталоге будут храниться **cgi-сценарии** на **Perl**, служебные файлы и папки с нашими собственными файлами);
- **адрес к интерпретатору Perl**: **C:\Perl\BIN\Perl.exe** (если он есть);
- **каталог с cgi-файлами**: **\cgi** (их пока нет, но могут быть);
- **MIME** типы - параметр **ext**: .wml **text/vnd.wap.wml**, (он показывает, что серверу следует ожидать от файла с расширением **wml**).

На вкладке **HOSTS** укажем:

- **IP:** 127.0.0.1;
- **Host:** localhost;
- **Dir:** C:\WWW\WWWROOT\ok.ru.

Можно указать и псевдонимы, для обращения к каталогам (**wap.ok.ru**), тогда вместо **http://127.0.0.1/filename.htm**, к сайту можно будет обращаться **http://wap.ok.ru/filename.htm**.

После того, как настройка выполнена, останется нажать кнопку [**SET&EXIT**], а затем запустить сам сервер **http.exe** из папки **C:\WWW\MyServer\HTTP**. Если сервер успешно запустился, должно появиться следующее окно:



которое удобно свернуть в системный трей.

В папке **C:\WWW\WWWROOT\ok.ru** необходимо разместить служебный файл **index.htm**. Этот файл вызывается автоматически тогда, когда запрашиваемый адрес заканчивается на слэш (например, **http://127.0.0.1/**), а также в тех случаях, когда запрашиваемый контент на сервере не найден. Файл может содержать соответствующие сообщения и какое-либо меню или ссылки.

Для проверки работы сайта в каталоге **C:\WWW\WWWROOT\ok.ru** размещаем произвольный файл (например, **hello.htm**) и вызываем его WEB-браузером, набрав URL: **http://127.0.0.1/hello.htm,** запрошенный файл должен отобразиться в окне WEB-браузера.

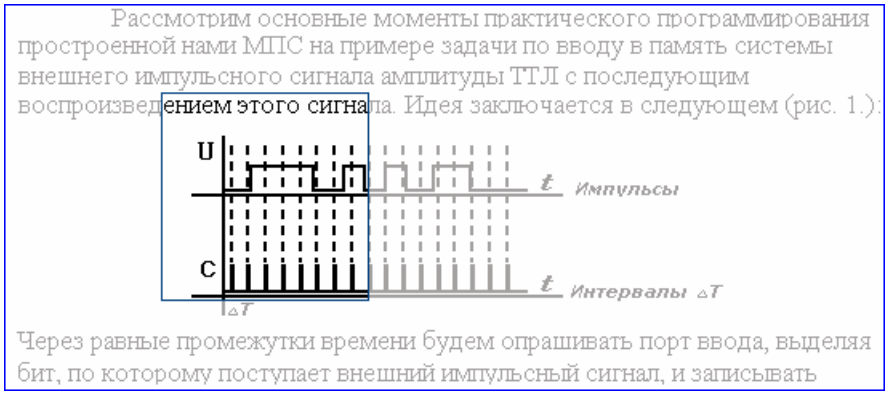
ПРИЛОЖЕНИЕ 3.

Подготовка методических материалов в формате JAVA-книг

Предложенная методика создания документов в формате ***.wml** рассчитана на широкий круг мобильных устройств, поддерживающих WAP. Развитые модели телефонов способны исполнять загружаемое программное обеспечение. Разработчики снабдили их межплатформенной виртуальной машиной, позволяющей выполнять **Java** байт-код и обеспечивающей программную совместимость различных моделей. Многочисленные рекламируемые игрушки для сотовых телефонов и различные Интернет–развлечения представляют собой программы на языке **Java**, так называемые, Java-апплеты, которые в случае мобильных телефонов носят название мидлеты. Мидлеты создаются специальной средой разработки носящей название J2M (**Java** для мобильных устройств).

Телефоны, способные исполнять **Java** байт-код, способны обрабатывать Интернет–контент программно, что значительно расширяет возможности программных WAP–браузеров. Наличие в программном обеспечении мобильных телефонов Java-машины позволяет предложить для создания методических WAP-документов весьма специфический метод, заключающийся в оформлении их в формат Java–книг.

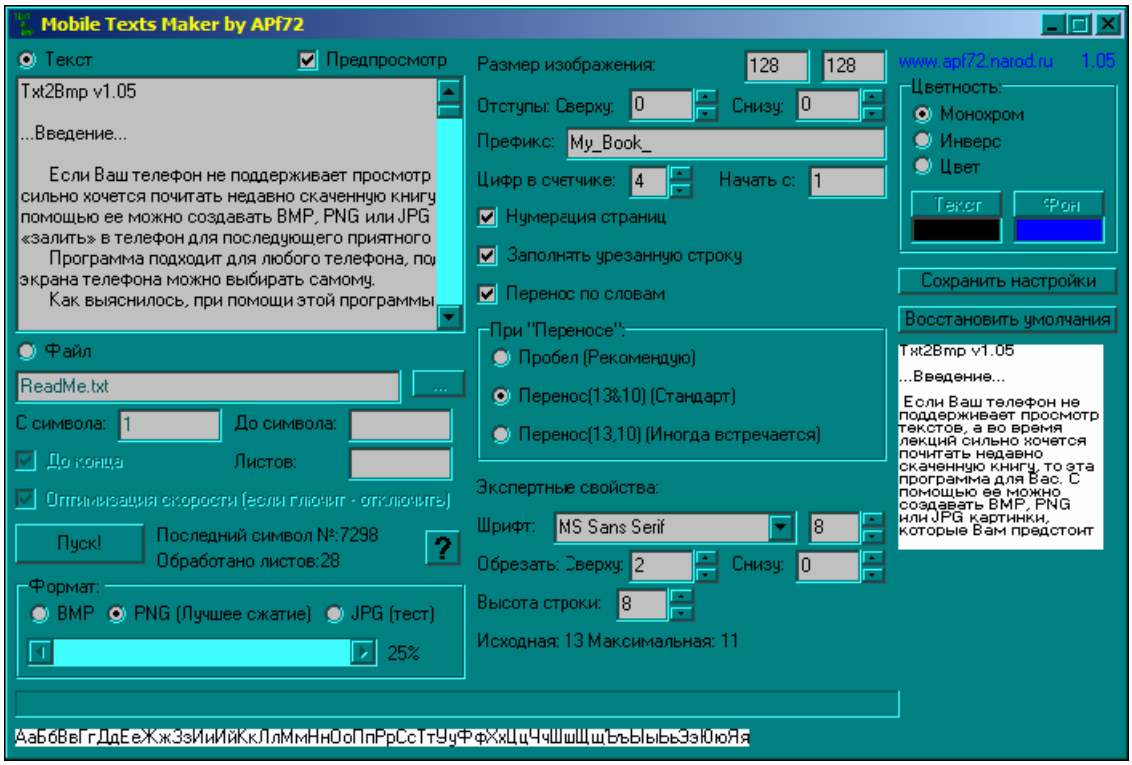
Основная идея метода заключается в том, что исходный текст документа оформляется в виде набора графических файлов распространенных форматов *.gif, *.jpg, *.png, причем размеры картинки не лимитированы размерами экрана мобильного аппарата. Программное обеспечение позволяет отображать картинку, фактически просматривая её через окошко в размер собственного экрана (см. рис.).



Указанный метод широко применяется для создания так называемых Java-книг, в которых исходный текст конвертируется в набор картинок, которые вместе с программой обработки загружаются в мобильный телефон. Достоинством методики является фактическая независимость от шрифтов, установленных в мобильном аппарате, поскольку изображение текста формируется в персональном компьютере, способность просматривать иллюстрации крупного размера, а также возможность работать с документом в режиме off-line после его закачки в аппарат из сети.

Для создания Java-книг существует бесплатное программное обеспечение доступное в Сети, из числа которого можно рекомендовать две весьма удачные удилиты: **Txt2Bmp** (Mobile Texts Maker by APf72) и **mjBookMaker** (© mjSoftware).

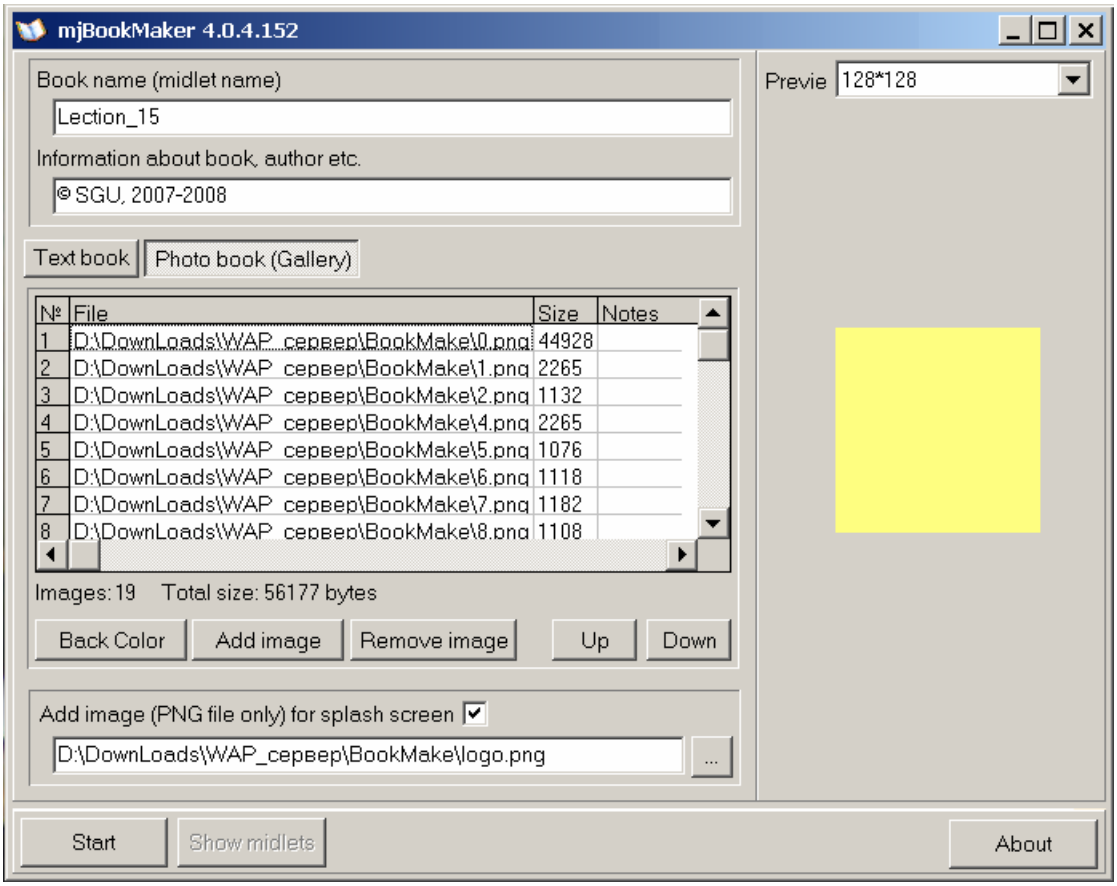
Mobile Texts Maker представляет собой конвертер текстовых файлов в картинки форматов *.bmp, *.jpg, *.png размерами кратными размерам экранов популярных мобильных аппаратов. Русскоязычный интерфейс программы интуитивно прост (см. рис) и обладает гибкой системой настроек.



После загрузки текстового файла и установки премлемых настроек программа формирует комплект картинок заданного формата и размера. Опции настройки подробно описаны в файле **readme.txt**, прилагающемся к программе. Файлы

иллюстраций добавляются к набору текстовых картинок уже на этапе создания Java-книги.

Сформировать электронную Java-книгу удобно с помощью утилиты **mjBookMaker** (см. рис.), которая формирует мидлет, совместимый с большим количеством моделей сотовых телефонов и записывает в него комплект рисунков в заданной последовательности.



После занесения кнопкой "**Add image**" имен графических файлов в список программы в нужной последовательности и введения дополнительных сведений (о названии книги – "**Book name**" и авторской информации – "**Information about book**"), нажатием кнопки "**Start**" в папке "**midlets**" текущего каталога формируется мидлет с именем, внесенным в поле "**Book name**".

Исходный файл размером в 9 стандартных страниц MS Word объемом 67.5 Кбайт был преобразован программой в Java-мидлет **Lection_15.jar** размером 39.9 Кбайт. Сформированный мидлет можно разместить в папке **prg** на WAP-сайте и добавить ссылку для его вызова в **wml**-файл следующей строкой:

```
<p align="right"><a href="prg/LearnBook.jad">Загрузить</a></p>
```

Ссылка позволит скачать соответствующий электронный документ при просмотре **wml**-страницы и просматривать его в режиме **off-line**.

Для тестирования получаемых мидлетов до размещения их на сайте очень удобна программа **Kwyshell Midlet2Exe Convert Tool 1.0** (**Midp2Exe**). Она позволяет трансформировать ***.jar** файлы в exe-модули, исполняемые в среде MS Windows без какого-либо дополнительного программного обеспечения. Для упрощения работы с утилитой и расширения возможностей удобно использовать графическую оболочку **GUIMidp2Exe.exe**, которая позволяет:

- преобразовать как единичные файлы, так и каталоги (включая подкаталоги), содержащие **jar** файлы.

- показывает полную информацию о выбранном **jar** файле и иконку (если она есть)

- генерирует **jad** файл

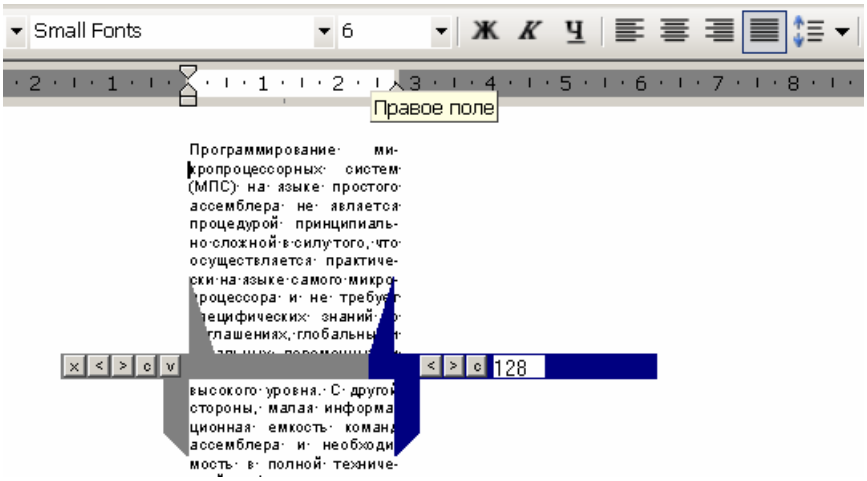
- помогает запустить для проверки сгенерированный **exe**–файл.

Следующие иллюстрации демонстрируют возможности программы **Midp2Exe** по тестированию созданного мидлета электронной книги.

Программа **Midp2Exe** представляет собой полнофункциональный эмулятор сотового телефона, предоставляющий пользователю возможность предварительного тестирования Java-мидлетов в режиме **off-line.**

Как видно на втором рисунке, конвертер текстовых файлов Mobile Texts Maker при создании картинки не обеспечивает выравнивание формируемого текста или его литературный перенос. Для того чтобы текст в картинках имел более аккуратное форматирование, следует прибегнуть к следующему алгоритму формирования исходного текста.

Текст загружается в MS Word и ему присваивается шрифт "**Small fonts**", после чего выбирается правильный язык для текста (в меню <**Сервис**> → <**Язык**> → <**Выбрать язык**>) и разрешается автоматическая расстановка переносов (<**Сервис**> → <**Язык**> → <**Расстановка переносов**>). Когда эти опции для текста установлены, размер страницы уменьшается до необходимой величины (~2.5 см для размеров экрана 128x128). MS Word форматирует текст к заданным размерам и расставляет знаки переносов (см. рис.).



Для более точной установки размера правого поля удобно использовать утилиту **Calipers**, которая представляет собой визуальную экранную линейку (или штангенциркуль), отображающую экранные размеры в пикселях.

После того как исходный текст отформатирован MS Word, его необходимо сохранить как **текстовый документ с разбиением на строки**. Полученный таким образом файл, обрабатывается программой Mobile Texts Maker, которая сохранит форматирование при формировании графических файлов.

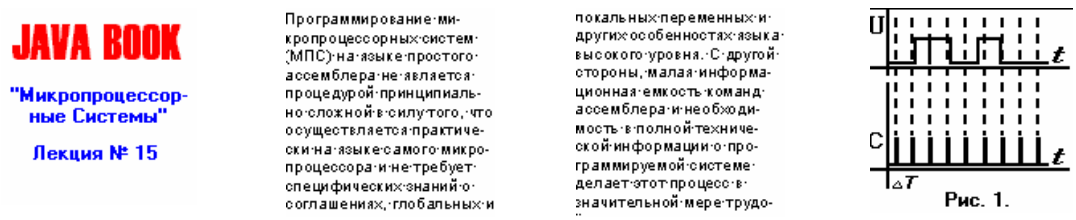
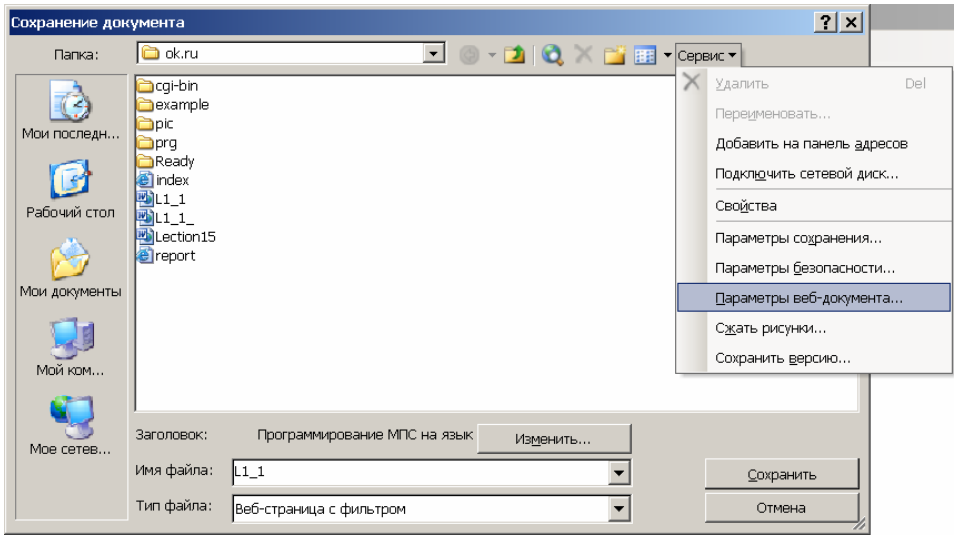


Рис. 1.

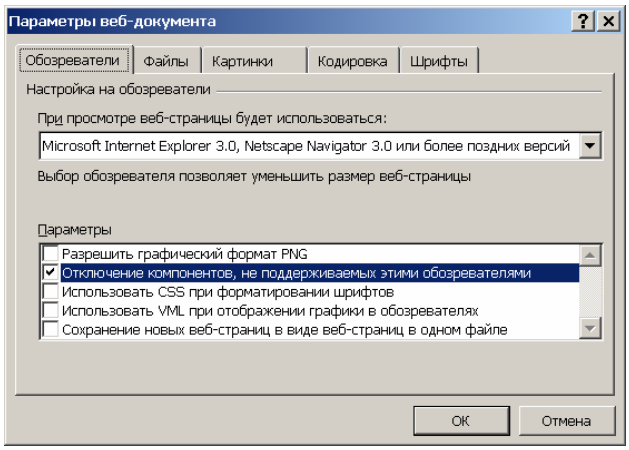
ПРИЛОЖЕНИЕ 4.

Настройка MS Word 2003 для работы с файлами HTML

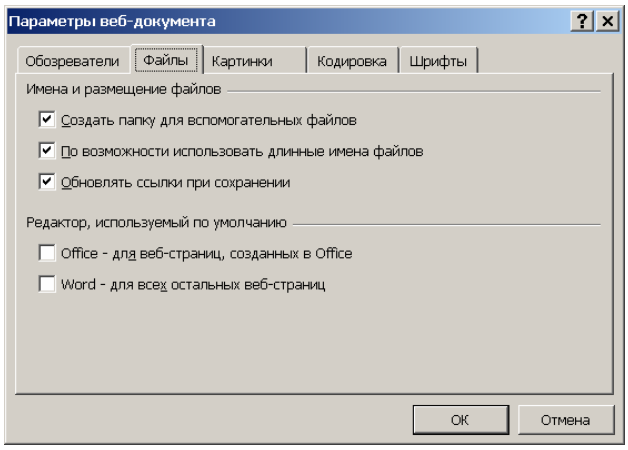
MS Word 2000 и более старших версий предлагает ряд дополнительных настроек по сохранению документов в формате HTML. Настройки доступны в меню <**Сервис**> окна "**Сохранение документа**" → <**Параметры веб–документа**> (см. рисунок.)



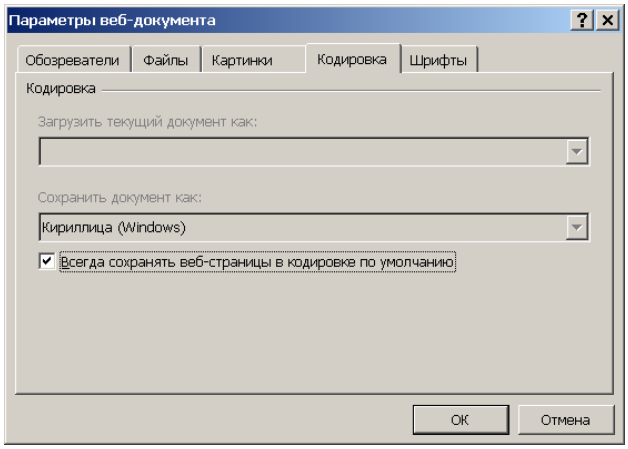
Параметры, рекомендуемые для применения с предлагаемой методикой формирования HTML файлов, указаны на рисунках ниже.



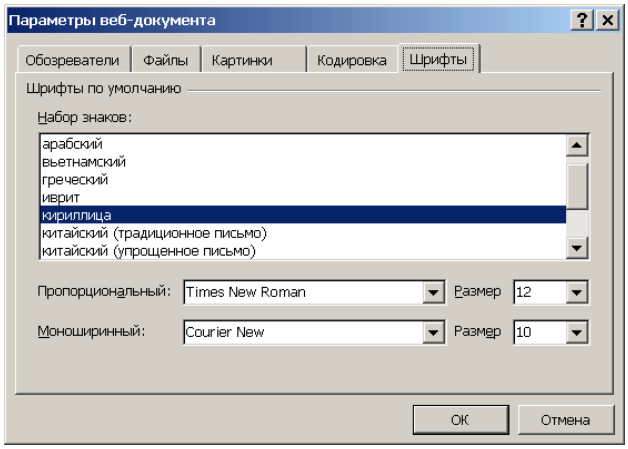
Установка параметров Интернет–обозревателя.



Опции, выставляемые для сохраняемых файлов.



Опции, выставляемые для кодировки сохраняемых файлов.



Опции, выставляемые для шрифтов сохраняемых файлов.