

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Петербургский государственный университет путей сообщения
Императора Александра I»
(ФГБОУ ВО ПГУПС)

Петрозаводский филиал ПГУПС

ОДОБРЕНО

на заседании цикловой комиссии

протокол № 11 от 23.06.2017

Председатель цикловой комиссии:

И. А. Ковалева

УТВЕРЖДАЮ

Начальник УМО

А. В. Калько

«23» 06

А.В. Калько

2017 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по организации и проведению практических занятий

По МДК 02.01. Программное обеспечение компьютерных сетей

Специальность: 09.02.02 Компьютерные сети

Разработчик: преподаватель ПФ ПГУПС Усков Алексей Андреевич

2017

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания по организации и проведению лабораторных работ и практических занятий разработаны в соответствии с рабочей программой МДК 02.01. Программное обеспечение компьютерных сетей для специальности 09.02.02 Компьютерные сети и предназначено для выполнения лабораторных работ и практических занятий обучающимися.

Практические занятия и лабораторные работы по МДК 02.01. Программное обеспечение компьютерных сетей направлены на приобретение практического опыта, усвоение знаний, освоение умений и формирование элементов общих и профессиональных компетенций, предусмотренных рабочей программой МДК.

В результате освоения МДК, обучающийся должен

иметь практический опыт:

- настройки сервера и рабочих станций для безопасной передачи информации;
- установки web-сервера;
- организации доступа к локальным и глобальным сетям;
- сопровождения и контроля использования почтового сервера, SQL сервера;
- расчёта стоимости лицензионного программного обеспечения сетевой инфраструктуры;
- сбора данных для анализа использования и функционирования программно-технических средств компьютерных сетей;

уметь:

- администрировать локальные вычислительные сети;
- принимать меры по устранению возможных сбоев;
- устанавливать информационную систему;
- создавать и конфигурировать учетные записи отдельных пользователей и пользовательских групп;
- регистрировать подключения к домену, вести отчётную документацию;
- рассчитывать стоимость лицензионного программного обеспечения сетевой инфраструктуры;
- устанавливать и конфигурировать антивирусное программное обеспечение, программное обеспечение баз данных, программное обеспечение мониторинга;
- обеспечивать защиту при подключении к информационно-коммуникационной сети «Интернет» (далее – сеть

Интернет) средствами операционной системы;

знать:

- основные направления администрирования компьютерных сетей;
- типы серверов, технологию «клиент – сервер»;
- способы установки и управления сервером;
- утилиты, функции, удаленное управление сервером;
- технологии безопасности, протоколы авторизации, конфиденциальность и безопасность при работе в web;
- порядок использования кластеров;
- порядок взаимодействия различных операционных систем;
- алгоритм автоматизации задач обслуживания;
- порядок мониторинга и настройки производительности;
- технологию ведения отчётной документации;
- классификацию программного обеспечения сетевых технологий, и область его применения;

- порядок и основы лицензирования программного обеспечения;
- оценку стоимости программного обеспечения в зависимости от способа и места его использования.

В результате освоения МДК 02.01. Программное обеспечение компьютерных сетей происходит поэтапное формирование элементов общих и профессиональных компетенций:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности

ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности

ПК 2.1. Администрировать локальные вычислительные сети и принимать меры по устранению возможных сбоев.

ПК 2.2. Администрировать сетевые ресурсы в информационных системах.

ПК 2.3. Обеспечивать сбор данных для анализа использования и функционирования программно-технических средств компьютерных сетей

ПК 2.4. Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности

Рабочей программой предусмотрено выполнение обучающимися практических занятий, включая, как обязательный компонент практические задания с использованием персонального компьютера.

Распределение результатов освоения учебного материала в ходе выполнения лабораторных работ и практических заданий на практических занятиях происходит в соответствии с таблицей 1.

Таблица 1 – Распределение результатов освоения учебного материала

| Элемент модуля | Контрольно-оценочные мероприятия | Результаты | | | Поэтапно формируемые элементы общих и профессиональных |
|----------------|----------------------------------|------------------|------------------|-------------------|--|
| | | усвоенные знания | освоенные умения | практический опыт | |
| | | | | | ных |

| | | | | | компетенций |
|--|---|---|--|--|---|
| <p>Раздел 1. Разработка методов, средств и технологий применения и управления сетевыми ресурсами в информационных системах.</p> <p>МДК 02.01. «Программное обеспечение компьютерных сетей»</p> | <p>Практические занятия №1-20, тестирование</p> | <ul style="list-style-type: none"> -использование кластеров; -взаимодействие различных операционных систем; -классификацию программного обеспечения сетевых технологий и область его применения; -лицензирование программного обеспечения; | <ul style="list-style-type: none"> -устанавливать информационную систему; -рассчитывать стоимость лицензионного программного обеспечения сетевой инфраструктуры; - устанавливать и конфигурировать антивирусное программное обеспечение, программно-обеспечение баз данных, программно-обеспечение мониторинга; | <ul style="list-style-type: none"> -настройки сервера и рабочих станций для безопасной передачи информации; -установки web-сервера; -расчёта стоимости лицензионного программного обеспечения сетевой инфраструктуры; | <p>ПК 2.1- 2.4 ОК 1-9</p> |
| <p>Раздел 2. Администрирование локальных вычислительных сетей, сбор и анализ данных функционировании программно-технических средств компьютерных сетей и принятие мер по устранению возможных сбоев.</p> <p>МДК 02.02. «Организация администрирования компьютерных сетей»</p> | <p>Практические занятия 1-3 Лабораторные работы №1-11, тестирование</p> | <ul style="list-style-type: none"> -основные направления администрирования компьютерных сетей; -типы серверов, технологию «клиент – сервер»; -способы установки и управления сервером; -утилиты, функции, удаленное управление сервером; -мониторинг и настройку производительности; -технологию ведения отчетной документации; - оценку стоимости программного обеспечения в зависимости от способа и места его использования | <ul style="list-style-type: none"> администрировать локальные вычислительные сети; - создавать и конфигурировать учетные записи отдельных пользователей и пользовательских групп; - обеспечивать защиту Интернет средствами операционной системы; | <ul style="list-style-type: none"> -установки web-сервера; -организации доступа к локальным и глобальным сетям; -сбора данных для анализа использования и функционирования программно – технических средств компьютерных сетей; | <p>ПК 2.1- 2.4, ОК 1-9</p> |
| <p>Учебная практика П.02.01 «Администрирование сетей»</p> | <p>Практические занятия</p> | <ul style="list-style-type: none"> -основные направления администрирования компьютерных сетей; -типы серверов, технологию «клиент – сервер»; -способы установки и управления сервером; -утилиты, функции, удаленное управление сервером; -мониторинг и настройку производительности; -технологию ведения отчетной документации; - оценку стоимости программного обеспечения в зависимости от способа и места его использования | <ul style="list-style-type: none"> администрировать локальные вычислительные сети; - создавать и конфигурировать учетные записи отдельных пользователей и пользовательских групп; - обеспечивать защиту Интернет средствами операционной системы; | <ul style="list-style-type: none"> -установки web-сервера; -организации доступа к локальным и глобальным сетям; -сбора данных для анализа использования и функционирования программно – технических средств компьютерных сетей; | <p>ПК 2.1-2.4</p> |
| <p>Раздел 3. Администрирование адресного пространства</p> | <p>Практические занятия 1-15, тестирование,</p> | <ul style="list-style-type: none"> -технологии безопасности, протоколы авторизации, конфиденциальность и безопасность при работе в | <ul style="list-style-type: none"> - принимать меры по устранению возможных сбоев | <ul style="list-style-type: none"> -организации доступа к локальным и глобальным | <p>ПК 2.1-2.4 ОК 1-9</p> |

| | | | | | |
|--|----------------------|--|--|--|------------------------------------|
| <p>ства и организация межсетевое взаимодействие в сетях различного масштаба.</p> <p>МДК 02.03 Межсетевое взаимодействие в крупных глобальных сетях»</p> | | web; | | сетям; | |
| <p>Учебная практика П.02.02 Межсетевое взаимодействие крупных сетей»</p> | Практические занятия | -технологии безопасности, протоколы авторизации, конфиденциальность и безопасность при работе в web; | - принимать меры по устранению возможных сбоев | -организации доступа к локальным и глобальным сетям; | ПК 2.1-2.4 |
| <p>Производственная практика (по профилю специальности) итоговая по модулю</p> | Практические занятия | -технологии безопасности, протоколы авторизации, конфиденциальность и безопасность при работе в web; | - принимать меры по устранению возможных сбоев | сопровождения и контроля использования почтового сервера, SQL сервера; | ПК 2.1-2.4 ОК 1-9 |

Содержание практических занятий и лабораторных работ охватывает весь круг умений и компетенций, на формирование которых направлен **МДК 02.01**.

«Программное обеспечение компьютерных сетей»

Перечень практических занятий

1. Создание своего дистрибутива.
2. Автоматизация процедуры установки операционных систем для рабочих станций. Интеграция обновлений и программного обеспечения в дистрибутив.
3. Командная строка. Пакетные и командные файлы
4. Пакетные и командные файлы
5. Пакетные и командные файлы
6. Установка Windows server 2008.
7. Windows server 2008.Настройка FTP-сервера.
8. Создание документа с использованием языка HTML.
9. Создание документа с использованием языка HTML.
10. Создание документа с использованием языка HTML.
11. Создание документа с использованием языка HTML.
12. Создание документа с использованием таблиц CSS.
13. Создание документа с использованием языка Java.
14. Создание документа с использованием языка Java.
15. Создание документа с использованием языка Java.
16. Создание документа с использованием языка Java.
17. Создание документа с использованием языка PHP.
18. Создание документа с использованием языка PHP.

19. Создание документа с использованием языка PHP.
20. Создание документа с использованием языка PHP.

Усвоенные знания, усвоенные умения, практический опыт, формируемые элементы компетенций

Знания:

- основные направления администрирования компьютерных сетей;
- типы серверов, технологию «клиент – сервер»;
- использование кластеров;
- технологии безопасности, протоколы авторизации, конфиденциальность и безопасность при работе в web;
- взаимодействие различных операционных систем;
- автоматизацию задач обслуживания;
- классификацию программного обеспечения сетевых технологий и область его применения;
- лицензирование программного обеспечения;
- оценку стоимости программного обеспечения в зависимости от способа и места его использования.

Умения:

- принимать меры по устранению возможных сбоев
- устанавливать информационную систему
- рассчитывать стоимость лицензионного программного обеспечения сетевой инфраструктуры
- устанавливать и конфигурировать антивирусное программное обеспечение, программное обеспечение баз данных, программное обеспечение мониторинга
- обеспечивать защиту при подключении к Интернет средствами операционной системы

Практический опыт:

- настройки сервера и рабочих станций для безопасной передачи информации;
- расчёта стоимости лицензионного программного обеспечения сетевой инфраструктуры;

Практическое занятие №1.

Тема: Создание своего дистрибутива.

Цель: Научиться создавать свой дистрибутив операционной системы, на основе Windows 7.

Ход Работы:

1. Ваш рабочий компьютер (ПК1) с установленной системой Windows 7 любой разрядности.
1. Установочный диск Windows 7
2. Установочный диск WAIK (Windows Automated Installation Kit),
3. Виртуальная машина (VM),
4. Набор драйверов, которые будут установлены в системе.
5. Дистрибутивы приложений, которые будут установлены в системе и в последствии будут разворачиваться во время установки операционной системы, автоматически

План действий.

1. Установка операционной системы с установочного диска Windows 7
 2. Теперь применим твики реестра.
 3. Подготовка системы с помощью утилиты sysprep.
 4. Переходим к подготовке образа Windows PE
 5. Захват образа установленной системы.
 6. Создание дистрибутивного образа ISO.
 7. Интеграция драйверов в образ.
- Приступим.

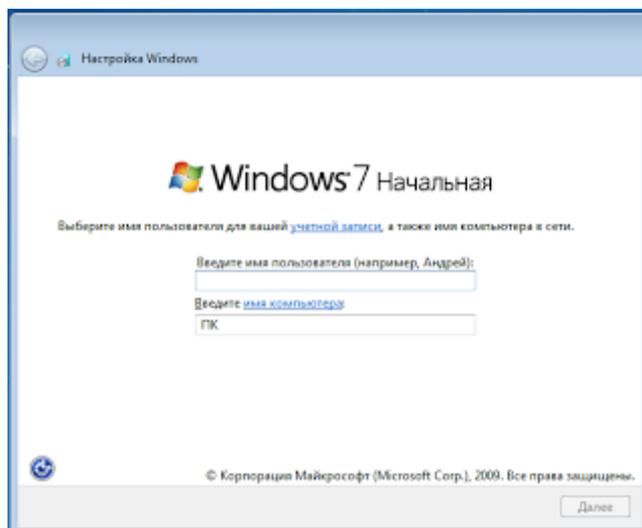
1. Установка операционной системы с установочного диска Windows 7

Устанавливаем Windows 7 Starter в VM или на ПК2.

Установка не таит в себе никаких хитростей. Если вы планируете сохранить настроенный образ на несистемном разделе диска, используйте программу установки для создания разделов и создайте два раздела в VM (или ПК2) при установке.

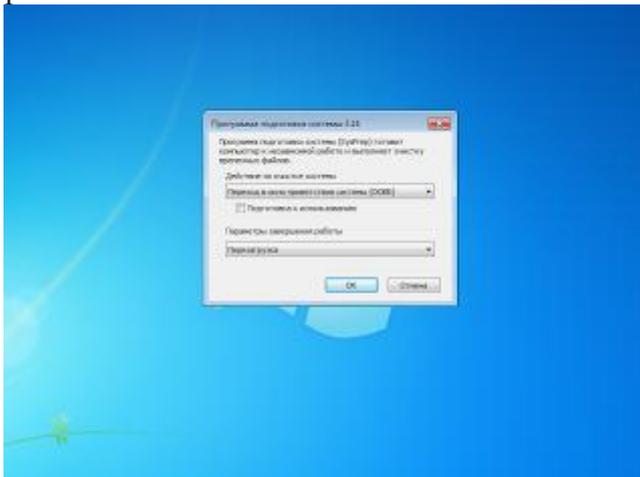
Совет. При установке системы на VMWare Workstation или Virtual Server 2005 можно подключить второй виртуальный диск и впоследствии использовать его для сохранения образа. Это упростит копирование образа в вашу рабочую среду, поскольку в этих виртуальных машинах есть возможность подключения виртуального диска к физической системе.

Продолжайте установку вплоть до этапа OOBE. Его легко узнать по предложению выбрать имя учетной записи и имя компьютера.



На этом этапе, не выбирая имя учетной записи, нажмите CTRL+SHIFT+F3. Эта комбинация клавиш переведет систему в режим аудита с правами встроенной учетной записи Администратор.

Система перезагрузится, появится надпись "Программа установки подготавливает компьютер к первому использованию". Затем вы увидите рабочий стол:



Не закрывайте данное окно. Если все-таки закрыли, что делать будет сказано ниже.

Теперь можно поставить нужные программы, так же установить обновления через Центр обновлений Windows. Если для установки приложения требуется перезагрузка системы, вы можете это сделать. После перезагрузки система вернется в режим аудита.

В "Панели управления - Программы и компоненты" через оснастку "Включение или отключение компонентов Windows" я отключаю игры и платформу гаджетов Windows, включаю клиент telnet (помогает иногда для диагностики).

2. Теперь применим твики реестра.

Особенность применения твиков на данном этапе в том, что мы не можем применить пользовательские настройки привычным способом, т.к. реального пользователя ещё не существует, а значит, не существует и его реестра. Решение задачи сводится к настройке пользователя по умолчанию (Default User), который станет донором для всех, создаваемых впоследствии пользователей.

Файл реестра пользователя по умолчанию находится по пути C:\Users\Default\NTUSER.DAT и чтобы внести в него изменения, необходимо подключить этот файл к активному реестру. Для этого выполняем команду:

```
reg load HKEY_USERS\Custom C:\Users\Default\NTUSER.DAT
```

Теперь вы можете открыть реестр (WIN+R >> regedit) и убедиться, что появилась новая ветка HKEY_USERS\Custom, в которую нам, как раз и предстоит внести настройки.

Применяем такие настройки:

Windows Registry Editor Version 5.00

;Использовать маленькие значки на панели задач

```
[HKEY_USERS\Custom\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced]
"TaskbarSmallIcons"=dword:00000001
```

;Отображать все значки и уведомления в области уведомлений

```
[HKEY_USERS\Custom\Software\Microsoft\Windows\CurrentVersion\Explorer]
"EnableAutoTray"=dword:00000000
```

;Запрет на использование USB-накопителей

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\USBSTOR]
"Start"=dword:00000004
```

После применения этого твика выгружаем файл реестра командой:

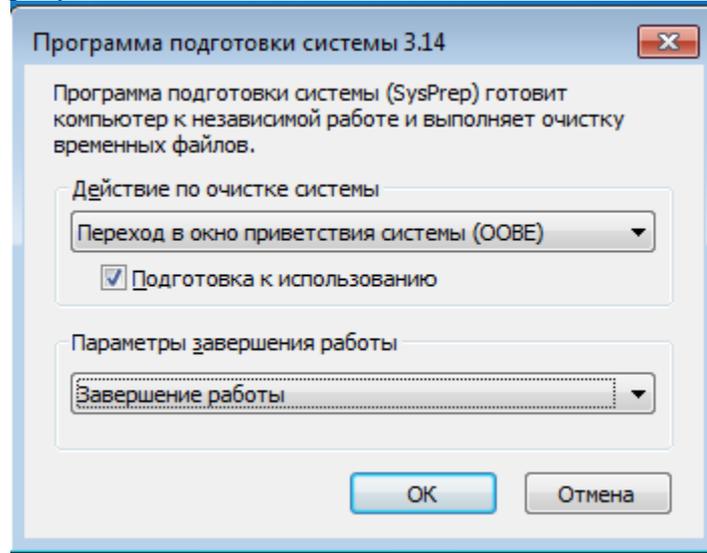
```
reg unload HKEY_USERS\Custom
```

Данный способ подсмотрен здесь.

Если установлены все программы, и сделаны все настройки, то выполняется подготовка с помощью утилиты sysprep.

3. Подготовка системы с помощью утилиты sysprep.

При ручной установке системы утилита sysprep запускается при входе в режим аудита. Для подготовки системы к дальнейшему использованию (в данном случае — к созданию настроенного образа) необходимо задать утилите параметры, как показано на рисунке ниже.



Или, если вы случайно закрыли это окно командой:

```
%SystemRoot%\system32\sysprep\sysprep.exe /generalize /oobe /shutdown /quiet
```

Первые три параметра командной строки указаны в том же порядке, что и описанные выше параметры графического интерфейса утилиты. Параметр */quiet* — обеспечивает работу в тихом режиме и необходим для автоматизации. Теперь рассмотрим два последних.

После того, как *sysprep* завершит работу, система будет выключена. Теперь она полностью настроена, и можно создать ее образ с помощью утилиты *ImageX*.

4. Переходим к подготовке образа Windows PE

Windows Preinstallation Environment, что это такое, можно прочитать в Википедии.

Если вкратце, то эта "штука" загружается при установке Windows и являет собой оболочку для инсталлятора.

Дальнейшие действия производим на рабочем компьютере ПК1.

Нам необходимо установить WAIK. После этого:

Мы создаем дистрибутив Windows 7 Starter, тогда и образ для захвата будет Windows PE x86. Для начала, открываем командную строку, WIN+R пишем в ней CMD и жмем Enter.

Переходим в папку с файлами WinPE:

```
cd c:\Program Files\Windows AIK\Tools\PETools
```

Теперь, нам необходимо скопировать файлы Windows PE. Назовем папку win_pex86 и скопируем в нее файлы архитектуры x86, папку заранее создавать не требуется, она создается автоматически, при копировании.

```
copye.cmd x86 d:\win_pex86
```

Также, нам необходимо, закинуть *ImageX* в папку *d:\win_pex86\ISO*

```
copy "c:\Program Files\Windows AIK\Tools\x86\imagex.exe" d:\win_pex86\ISO
```

На всякий случай, добавим в наш диск, средство диагностики памяти, по умолчанию оно не добавляется, но в меню присутствует, делается это на тот случай, когда при установке возникают непонятные глюки системы и тест памяти может сильно облегчить жизнь, когда причина сбоя непонятна, а скачать и нарезать на болванку нет возможности:

```
copy c:\Windows\Boot\PCAT\memtest.exe d:\win_pex86\ISO\boot
```

Теперь, нам необходимо, добавить образ Windows PE который и будет загружаться с нашего диска. В папке *d:\win_pex86* лежит файл *winpe.wim* именно он нам и нужен, его необходимо скопировать в папку *d:\win_pex86\ISO\sources* и переименовать в *boot.wim*, давайте сделаем это.

Копируем образ Windows PE

```
copy d:\win_pex86\winpe.wim d:\win_pex86\ISO\sources
```

Переходим в папку:

```
cd d:\win_pex86\ISO\sources
```

Переименовываем файл *winpe.wim* в *boot.wim*

```
rename winpe.wim boot.wim
```

С основной задачей все, нам необходимо создать загрузочный ISO образ, который можно записать на пустую болванку, для этого нам необходимо перейти:

Пуск->Все программы->Microsoft Windows AIK->Командная строка средств развертывания, копируем туда следующее:

```
oscdimg -n -bd:\win_pex86\etfsboot.com d:\win_pex86\iso d:\win_pex86\winpe.iso
```

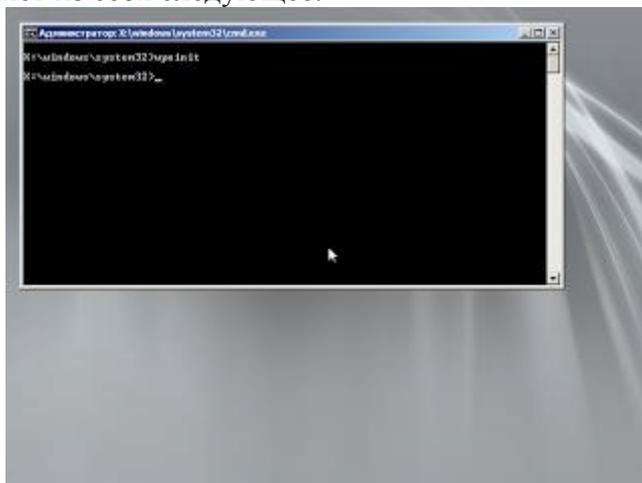
Внимание. Тут с пробелами все ок, ошибок в записи *-bd:...* нет.

В результате всех наших действий, в папке *d:\win_pex86* появится файл *winpe.iso*, это готовый для записи на болванку образ. Можете записать на чистый диск или сразу монтировать в VM.

5. Захват образа установленной системы.

Нам необходимо загрузиться с *winpe.iso*, который мы создали на ПК2 или VM.

Оболочка Windows PE представляет из себя следующее:



Не пугаемся :) все так и должно быть.

Вводим *notepad*, жмем *Enter*.

Через *Файл-открыть* рассматриваем наши диски, дабы определить кто из них кто.

Например, в моем случае диск C: с установленной Windows 7 Starter стал диском E:, а диск D: таковым и остался. А привод с образом winpe.iso имеет букву F:.

Запоминаем эти моменты, а лучше записываем :)

Закрываем блокнот и срезаем образ установленной Windows 7 Starter.

Команда:

```
f:\imagex.exe /capture E: d:\install.wim "Windows 7 Starter" "My Windows 7 Starter" /compress maximum /verify
```

В этой команде используются следующие параметры:

/capture E: — захват раздела E:

d:\install.wim — местоположение и имя сохраняемого WIM-файла. Укажите букву диска, соответствующую несистемному разделу или USB-диску.

"Windows 7 Starter" — имя образа. Имя должно быть заключено в кавычки.

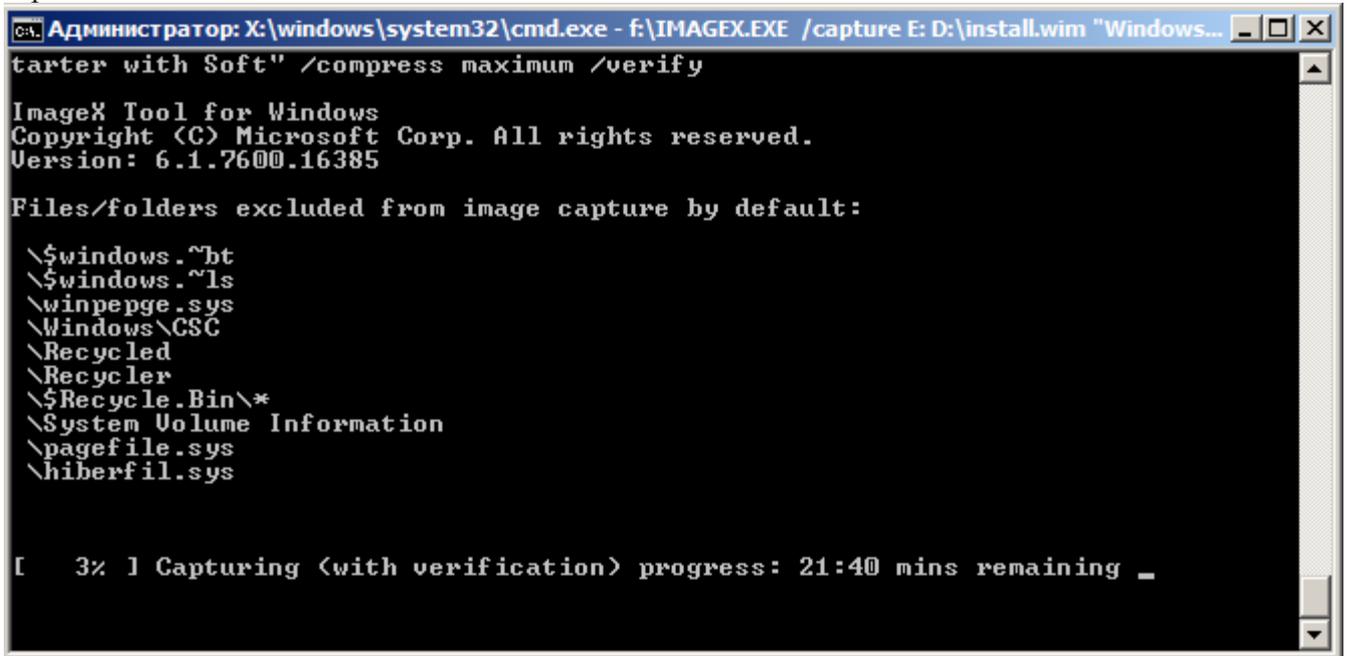
"My Windows 7 Starter" — (необязательно) описание образа. Описание должно быть заключено в кавычки.

/compress maximum — (необязательно) тип сжатия файлов в образе. Для параметра допустимы значения maximum (максимальное сжатие), fast (быстрое сжатие) и none (без сжатия). Если параметр не указан, используется быстрое сжатие. Максимальное сжатие позволяет уменьшить размер образа, но захват занимает больше времени, чем при быстром сжатии. Скорость распаковки образа практически не зависит от типа сжатия.

/verify - Обеспечивает проверку ресурсов файлов на наличие ошибок и дублирования при создании и применении. Данный параметр по умолчанию включен при действиях с сетью.

При максимальном сжатии захват образа может занять довольно длительное время.

Захват в процессе:



```
Администратор: X:\windows\system32\cmd.exe - f:\IMAGEX.EXE /capture E: D:\install.wim "Windows...
f:\imagex.exe /capture E: D:\install.wim "Windows 7 Starter with Soft" /compress maximum /verify

ImageX Tool for Windows
Copyright (C) Microsoft Corp. All rights reserved.
Version: 6.1.7600.16385

Files/folders excluded from image capture by default:
\$.windows.*bt
\$.windows.*ls
\winpepe.sys
\Windows\CSC
\Recycled
\Recycler
\$.Recycle.Bin\*
\System Volume Information
\pagefile.sys
\hiberfil.sys

[ 3% ] Capturing (with verification) progress: 21:40 mins remaining _
```

Пока образ создается идем гулять/пить кофе/кушать и т.д.

У нас ранее была создана папка, по пути d:\win_rex86\ISO, она нам понадобится еще раз, в ней мы соберем установочный образ, из папки ISO необходимо удалить все ее содержимое и скопировать в нее, содержимое установочного диска с Windows 7, перейдя в папку d:\win_rex86\ISO\sources скопировать туда файл install.wim (который у нас получился при захвате системы), т.к. эта папка уже содержит файл с таким именем, то соглашаемся на замену.

Практическое занятие №2.

Тема: Интеграция обновлений и программного обеспечения в дистрибутив.

Цель: Научиться интегрировать обновления ,драйверы и программное обеспечение в образ системы.

Ход работы:

Пуск->Все программы->Microsoft Windows AIK->Командная строка средств развертывания от имени Администратора.

Монтируем образ:

```
dism /mount-wim /wimfile:d:\win_pex86\ISO\sources\install.wim /index:1 /mountdir:d:\win_pex86\mount
```

где:

/mount-wim - команда монтирования WIM файла

/wimfile:d:\win_pex86\ISO\sources\install.wim - размещение WIM файла

/index:1 - индекс системы (в моем случае он отвечает индексу Starter, см. ниже)

/mountdir:d:\win_pex86\mount- в какую папку монтировать образ.

Как узнать индекс системы в которую мы интегрируем драйвера:

```
dism /get-wiminfo /wimfile:d:\win_pex86\ISO\sources\install.wim
```

Система DISMВерсия: 6.1.7600.16385

Сведения об образе : d:\win_pex86\ISO\sources\install.wim

Индекс: 1Имя : Windows 7 StarterОписание : My Windows 7 Starter with SoftРазмер (байт): 10 713 629 462

Операция успешно завершена.

Интегрируем драйвера для сетевых и wi-fi устройств и распаковываем в папку d:\drivers.Туда же помещаем программное обеспечение В папку распаковывали драйвера для Windows 7:

```
dism /image:d:\win_pex86\mount /add-driver /driver:d:\drivers /recurse /forceunsigned
```

/image:d:\win_pex86\mount - с каким образом работать.

/add-driver - добавить драйвера.

/driver:d:\drivers - папка с драйверами.

/recurse - позволяет брать драйвера еще и из подпапок.

/forceunsigned - если есть неподписанные драйверы.

Демонтируем образ:

```
dism /unmount-wim /mountdir:d:\win_pex86\mount /commit
```

Сборка финального ISO образа установочного диска Windows 7

Снова жмем Пуск->Все программы->Microsoft Windows AIK->Командная строка средств развертывания, копируем туда следующее:

```
oscdimg -n -m -o -bd:\win_pex86\etfsboot.com d:\win_pex86\ISO d:\win_pex86\win7-starter.iso
```

Система немного поработает и выдаст нам, готовый инсталляционный диск, который будет находиться в папке d:\win_pex86 с именем win7-starter.iso, если его размер не превысил 4,7 Gb, то могу вас поздравить, его можно нарезать на болванку.

Но прежде, чем что-то, куда-то записывать, нам необходимо протестировать работоспособность нашего ISOшника. Для этого нам можно воспользоваться виртуальной машиной, создаем на ней чистую систему и подключаем наш готовый образ, устанавливаем его, если все нормально, то можно нарезать на болванку и пользоваться, если загрузки с этого образа не происходит, то проверяем наличие файла etfsboot.com т.к. именно он и является той составляющей, которая делает обычную болванку, загрузочной.

Практическое занятие №3.

Тема: Командная строка. Пакетные и командные файлы

Цель: Научится работать с командами и создавать командные файлы.

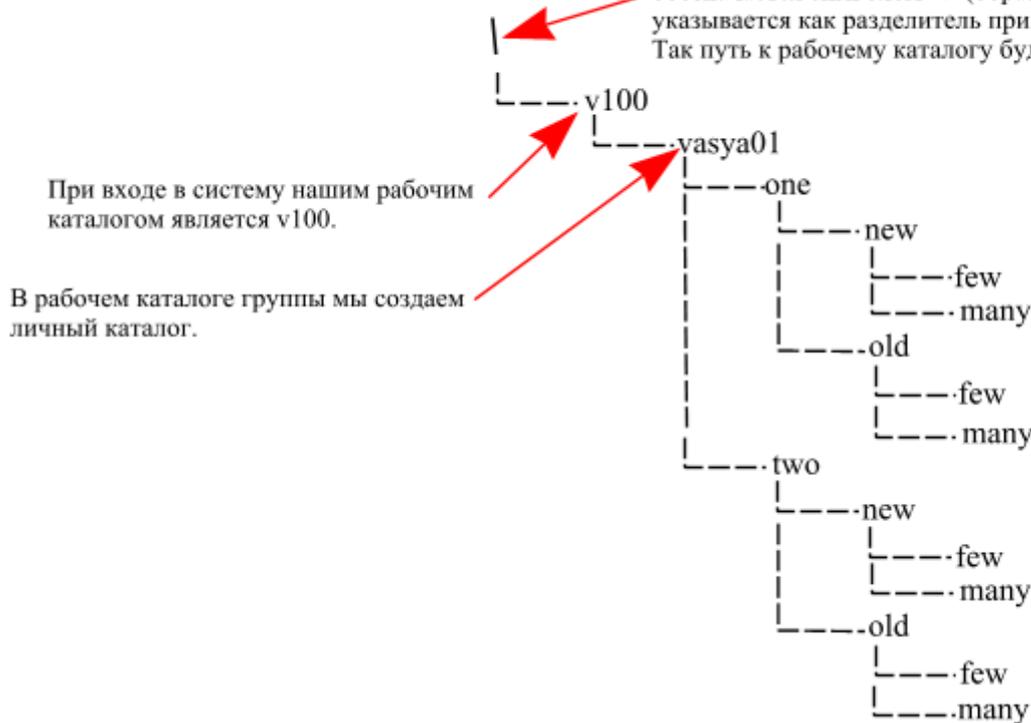
Ход работы:

Файлы, каталоги, файловая система - дерево каталогов. Практически вся информация на компьютерах представлена в виде файлов. Файл является основной единицей хранения данных и программ обрабатывающих эти данные. Файл - это именованная (т.е. снабженная именем) область внешней памяти. Обычно файлы временно или постоянно хранятся во внешней памяти компьютера - на дисках, магнитных лентах, "флэшках" и т.д. Кроме имени файлы характеризуются целым рядом атрибутов, таких как размер, время создания и т.п. Операционная система и прикладные программы (приложения) получают доступ к файлу по его имени. Максимальная длина имени файла или каталога в Windows 256 символов, включая расширение, имя и расширение разделяются точкой. Расширение указывает на вид информации или на приложение, которым может быть открыт этот файл, например `myfile.txt` - текстовый файл, `myfile.doc` - документ MS Word и т.д.

Дерево каталогов

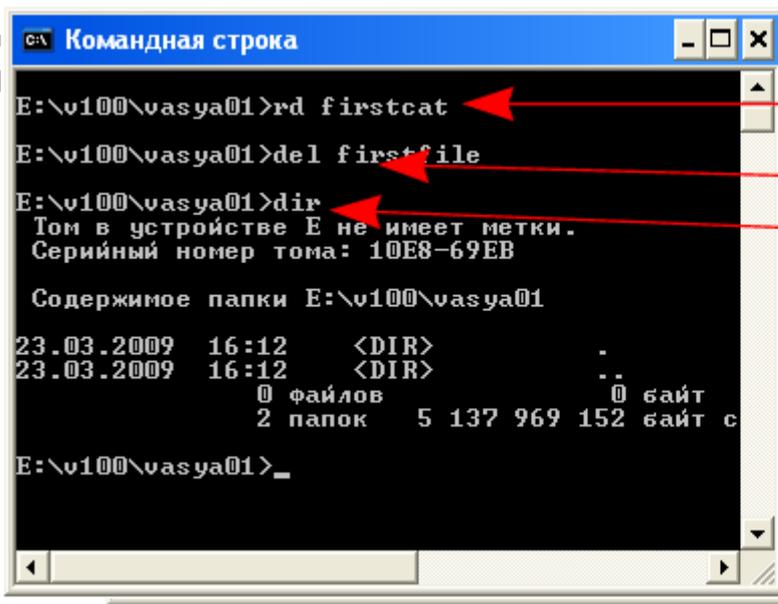
Файлы хранятся в системе вложенных каталогов (директорий) и организуются в файловую систему. Таким образом, файловой системой называется совокупность файлов и каталогов, организованных в древовидную структуру. Ее можно представить как перевернутое вверх корнем дерево (более точный образ - куст у которого от корня идет сразу множество ветвей, а не один ствол). Узлами, из которых расходятся "ветви", являются каталоги, восходящие, в конечном счете, к корневому каталогу. Узлом, из которого не происходит дальнейшего ветвления, как правило, соответствуют файлы, хотя это могут быть и пустые каталоги (см. дерево ниже). Обычно мы говорим: "Каталог (директория) содержит файлы" или "Файл находится в каталоге". Но при этом понимаем, что каталог не является областью памяти, собственно вмещающей сами файлы. Каталог лишь содержит список файлов, с указанием их имен и других атрибутов. По сути, каталог - это специализированный файл, назначением которого является хранение списка отнесенных к нему файлов (в том числе и подкаталогов, которые, как и все каталоги - тоже файлы). "Пустая" файловая система состоит только из корневого каталога.

В Windows каждый логический диск (в нашем случае диск E) имеет свою файловую систему. Корневой каталог диска обозначается символом "\" (обратный слэш). Этот символ также указывается как разделитель при записи пути к каталогу или файлу. Так путь к рабочему каталогу будет выглядеть так: `\v100\vasya01`



Рабочий каталог. Текущий каталог, абсолютный путь и относительный путь.

Когда мы входим в систему, то попадаем в свой рабочий каталог, он задан для нас системным администратором и сами мы его поменять не можем. В момент входа он является так же текущим каталогом. Текущим мы можем сделать любой существующий каталог, к которому у нас есть доступ. Зачем нужен текущий каталог? Дело в том, что операционная система осуществляет доступ к файлу или каталогу через его путевое имя или, попросту путь. Существуют два вида путей, абсолютные и относительные. Абсолютный путь - это последовательность имен каталогов, которая начинается от корневого каталога и, следуя по дереву файловой системы, заканчивается именем каталога или файла, с которым мы хотим работать. Относительный путь может быть указан от текущего или рабочего каталога, что может оказаться значительно короче и удобней, чем использование абсолютного пути. Когда мы находимся в рабочем каталоге, нам не нужно указывать пути к каталогам и



1. Удалим каталог firstcat посредством команды rd.
2. Удаление файла firstfile осуществим командой del.
3. Команда dir, как всегда, позволит удостовериться в наличии или отсутствии каталогов и файлов.

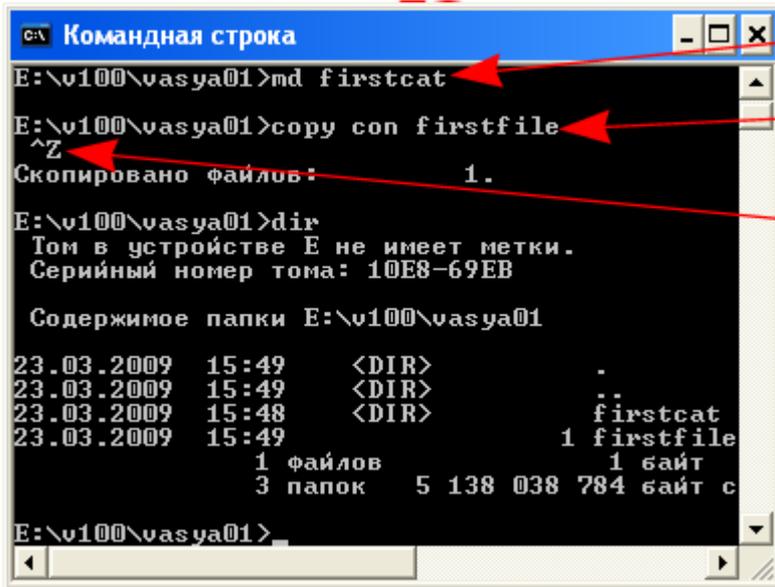
о).

Наш текущий каталог `\v100\vasya01`. Это полный путь от корневого каталога. Читается путь с права на лево, т.е текущий каталог `vasya01` является подкаталогом `v100`, расположенном в корневом каталоге, о чем свидетельствует начальный `"\"`.

Полезная функция. Попробуйте понажимать клавиши "ВВЕРХ" и "ВНИЗ" на клавиатуре, и Вы увидите, что командная строка запоминает команды, и этими самыми стрелочками их можно перелистывать и, при необходимости редактировать.

Как видно из примера, имена каталогов при написании пути разделяются символом `"\"`, который называется "обратный слэш". Если путь начинается с `"\"`, то это значит, что указан путь от корневого каталога или полный путь.

Создадим каталог и файл



1. Создадим каталог с именем firstcat при помощи команды md.
2. Для создания файла firstfile воспользуемся командой copy con, что означает копирование с консоли, т.е. с клавиатуры
3. После команды copy con нужно ввести любой текст и закрыть файл сочетанием клавиш "Ctrl"+"Z".
4. Команда dir демонстрирует нам список созданных объектов. перед каталогом указывается метка <DIR>. Первая метка <DIR> указывает на текущий каталог, обозначенный точкой, вторая метка <DIR> указывает на две точки - ими обозначена ссылка на предыдущий каталог, поэтому команда dir фиксирует в общей сложности 3 каталога (папки).

Удалим каталог и файл

Смена текущего каталога

Мы научились создавать и удалять каталоги, теперь научимся в них заходить и выходить обратно. Поскольку наша текущая директория пуста, то каталоги придется создавать снова.

```

C:\ Командная строка
E:\v100\vasya01>md firstcat
E:\v100\vasya01>dir
Том в устройстве E не имеет метки.
Серийный номер тома: 10E8-69EB

Содержимое папки E:\v100\vasya01
23.03.2009 16:28 <DIR> .
23.03.2009 16:28 <DIR> ..
23.03.2009 16:28 <DIR> firstcat
0 файлов 0 байт
3 папок 5 137 932 288 байт с
E:\v100\vasya01>cd firstcat
E:\v100\vasya01\firstcat>cd ..
E:\v100\vasya01>

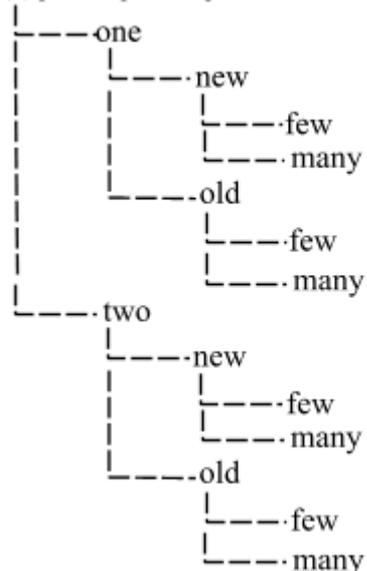
```

1. Создаем каталог firstcat, как обычно, командой md.
2. Воспользуемся dir и проверим, появился ли каталог.
3. Для того чтобы попасть в каталог firstcat используем команду cd.
4. Выйти из каталога firstcat на уровень выше нам поможет команда cd, в качестве параметра которой указаны две точки, которые и означают: "на уровень выше".

Вот, в принципе, мы и освоили набор команд, позволяющий построить нам дерево каталогов. Создав каталог firstcat, мы можем создать в нем столько каталогов, сколько потребуется, с любой степенью вложенности. Создаем - md, смотрим - dir, меняем текущий каталог - cd, удаляем, если ошиблись - rd. Давайте в последний раз удалим наш многострадальный firstcat (кто забыл, отмотайте назад) и приступим к построению дерева каталогов, параллельно изучая новые команды и возможности.

Начинаем строить дерево

Дерево строим в рабочем каталоге v100\vasya01



Чтобы было ясно какие имена каталогов использованы в этом дереве, запомните стишок:

```

one - two
old - new
many - few

```

Рекомендуем перерисовать построенное дерево на бумагу, чтобы постоянно видеть его перед собой

Несколько команд mkdir и дерево готово

Помимо команды md в Windows для создания каталогов можно воспользоваться командой mkdir, она удобна тем, что позволяет создавать цепочки вложенных каталогов:

```

C:\ Командная строка
E:\v100\vasya01>mkdir one\new\few
E:\v100\vasya01>mkdir one\new\many
E:\v100\vasya01>mkdir one\old\many
E:\v100\vasya01>mkdir one\old\few
E:\v100\vasya01>mkdir two\old\few
E:\v100\vasya01>mkdir two\old\many
E:\v100\vasya01>mkdir two\new\many
E:\v100\vasya01>mkdir two\new\few
E:\v100\vasya01>tree .
Структура папок
Серийный номер тома: 10E8-69EB
E:\V100\VASYA01
├── one
│   ├── new
│   │   ├── few
│   │   └── many
│   └── old
│       ├── few
│       └── many
└── two
    ├── new
    │   ├── few
    │   └── many
    └── old
        ├── few
        └── many
E:\v100\vasya01>

```

1. Команда `mkdir` позволяет строить ветви дерева каталогов, вместо того, чтобы строить их по отдельности.
2. При построении дерева мы пользуемся путями относительно нашего рабочего каталога.
3. Если Вы, создадите что-нибудь не то, у Вас есть команды: `cd`, `dir` и `rd`. Как ими пользоваться см. выше.
4. Рекомендуем использовать команду `tree` после каждого `mkdir`, чтобы процесс построения дерева был под контролем.

Работаем с относительными путями. Команда `cd`

Для удобства работы положите лист с нарисованным деревом перед собой.

При работе с относительными путями есть один весьма существенный момент. Посмотрите на дерево: если мы движемся от корня, то на каждой развилке обязательно должны указать имя очередного каталога, поскольку каждый каталог разветвляется на несколько, и у каждого есть свое имя. Когда мы движемся к корню мы не указываем имени каталога, поскольку каталог, в который мы можем попасть - единственный, вариантов нет. Поэтому вместо имени указывается "." (точка, точка). Имена каталогов в путях, даже если это "." разделяются \ (обратный слэш).

```

C:\ Командная строка
E:\v100\vasya01>cd one\new\few
E:\v100\vasya01\one\new\few>cd ..\..\..\two\old\many
E:\v100\vasya01\two\old\many>cd ..\few
E:\v100\vasya01\two\old\few>_

```

Создание и просмотр текстовых файлов.

```

C:\ Командная строка
E:\v100\vasya01\one\new\few>copy con paravas1
Папа у Васи силен в математике.^Z
Скопировано файлов: 1.
E:\v100\vasya01\one\new\few>type paravas1
Папа у Васи силен в математике.
E:\v100\vasya01\one\new\few>_

```

1. В директории `one\new\few` создадим текстовый файл `paravas1`.
2. Для его просмотра воспользуемся командой `type`.

А теперь самостоятельно создайте:

файл `paravas2` с текстом:

"Учится папа за Васю весь год" в каталоге `one\new\many`

файл paravas3 с текстом:

"Где это видано, где это слыхано" в каталоге one\old\few

файл paravas4 с текстом:

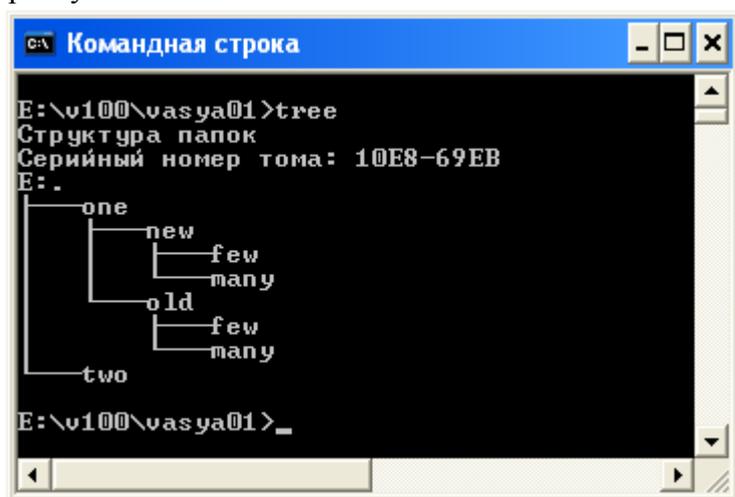
"Папа решает, а Вася сдает" в каталоге one\old\many

Построенное дерево каталогов и файлов предъявите преподавателю. Вас могут спросить: Что делает та или иная команда или какой командой выполняется то или иное действие. Что такое корневая, рабочая(домашняя) и текущая директория. Что такое абсолютный и относительный путь. Могут не только спросить, но и попросить продемонстрировать или на бумаге или на компьютере.

Копирование файлов.

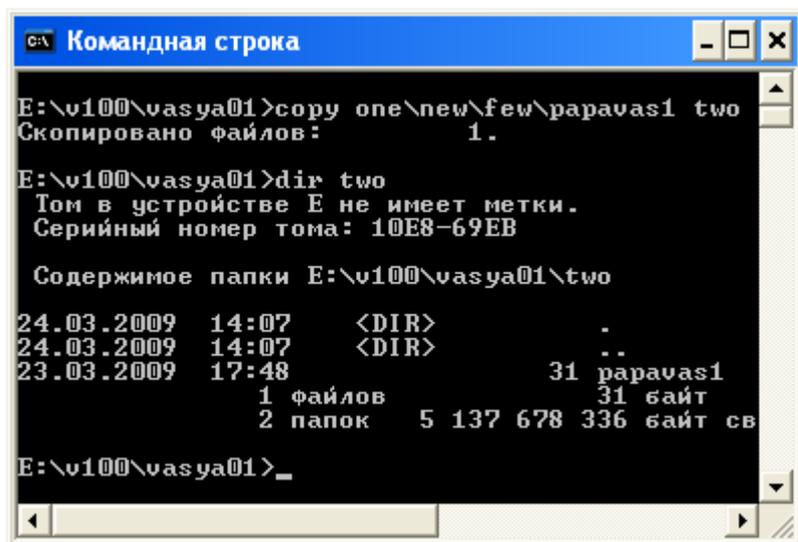
Копирование файлов осуществляется командой **copy что куда**. Где "что" и "куда" суть имена файлов, с указанием пути к ним. Для текущей директории путь, естественно, указывать не нужно.

Мы будем копировать файлы **paravas** в каталог **two**, он должен быть пустой, поэтому удалите из него все каталоги и приступим:



```
С:\> Командная строка
E:\v100\vasya01>tree
Структура папок
Серийный номер тома: 10E8-69EB
E:
├── one
│   ├── new
│   │   ├── few
│   │   └── many
│   └── old
│       ├── few
│       └── many
└── two
E:\v100\vasya01>_
```

Перед началом копирования каталог two должен быть пуст.



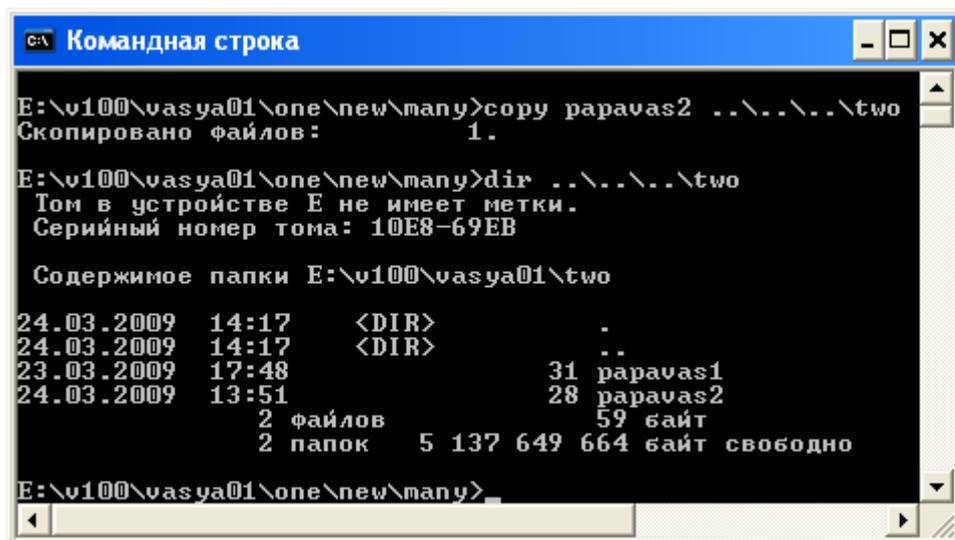
```
С:\> Командная строка
E:\v100\vasya01>copy one\new\few\paravas1 two
Скопировано файлов:      1.

E:\v100\vasya01>dir two
Том в устройстве E не имеет метки.
Серийный номер тома: 10E8-69EB

Содержимое папки E:\v100\vasya01\two

24.03.2009  14:07    <DIR>          .
24.03.2009  14:07    <DIR>          ..
23.03.2009  17:48                31 paravas1
                1 файлов      31 байт
                2 папок    5 137 678 336 байт св
E:\v100\vasya01>_
```

1. Текущая дериктория vasya01.
2. Копируем файл paravas1 в two не изменяя имя файла.



```
С:\> Командная строка
E:\v100\vasya01\one\new\many>copy paravas2 ..\..\..\two
Скопировано файлов:      1.

E:\v100\vasya01\one\new\many>dir ..\..\..\two
Том в устройстве E не имеет метки.
Серийный номер тома: 10E8-69EB

Содержимое папки E:\v100\vasya01\two

24.03.2009  14:17    <DIR>          .
24.03.2009  14:17    <DIR>          ..
23.03.2009  17:48                31 paravas1
24.03.2009  13:51                28 paravas2
                2 файлов      59 байт
                2 папок    5 137 649 664 байт свободно
E:\v100\vasya01\one\new\many>_
```

1. Текущая дериктория one\new\many.
2. Копируем файл paravas2 в two с тем же именем.

```

C:\ Командная строка
E:\v100\vasya01\two>copy ..\one\old\few\paravas3 .
Скопировано файлов:      1.

E:\v100\vasya01\two>dir
Том в устройстве E не имеет метки.
Серийный номер тома: 10E8-69EB

Содержимое папки E:\v100\vasya01\two

24.03.2009  14:25    <DIR>          .
24.03.2009  14:25    <DIR>          ..
23.03.2009  17:48                31 paravas1
24.03.2009  13:51                28 paravas2
24.03.2009  13:54                31 paravas3
                3 файлов             90 байт
                2 папок       5 137 612 800 байт свободно

E:\v100\vasya01\two>

```

1. Текущая дериктория two.
2. Копируем файл paravas3 в two с тем же именем.
3. Точка в качестве параметра (в конце), означает текущую директорию.

Файл paravas4 скопируйте в two самостоятельно.

Скопировав файлы в каталог two, предъявите результат преподавателю. После этого удалите все файлы и каталоги из каталога one, в него мы будем перемещать файлы из каталога two. у нас должно остаться дерево:

```

C:\ Командная строка
E:\v100\vasya01>tree
Структура папок
Серийный номер тома: 10E8-69EB
E:
├── one
└── two

E:\v100\vasya01>

```

Перемещение файлов.

Перемещение файлов осуществляется командой **move что куда**. Где "что" и "куда" суть имена файлов, с указанием пути к ним. В отличие от копирования файл "что" удаляется. в остальном действуют те же правила, что и для команды копирования.

```

C:\ Командная строка
E:\v100\vasya01>move two\paravas1 one\pv1
E:\v100\vasya01>dir one
Том в устройстве E не имеет метки.
Серийный номер тома: 10E8-69EB

Содержимое папки E:\v100\vasya01\one

24.03.2009  14:46    <DIR>          .
24.03.2009  14:46    <DIR>          ..
23.03.2009  17:48                31 pv1
                1 файлов             31 байт
                2 папок       5 137 534 976 байт с

E:\v100\vasya01>dir two
Том в устройстве E не имеет метки.
Серийный номер тома: 10E8-69EB

Содержимое папки E:\v100\vasya01\two

24.03.2009  14:46    <DIR>          .
24.03.2009  14:46    <DIR>          ..
24.03.2009  13:51                28 paravas2
24.03.2009  13:54                31 paravas3
24.03.2009  13:55                26 paravas4
                3 файлов             85 байт
                2 папок       5 137 534 976 байт с

E:\v100\vasya01>

```

1. Текущий каталог vasya01.
2. Перемещаем файл paravas1 из two в one с именем pv1.
3. Из каталога two файл paravas1 исчез.

```

C:\ Командная строка
E:\v100\vasya01\one>move ..\two\paravas2 .
E:\v100\vasya01\one>dir
Том в устройстве E не имеет метки.
Серийный номер тома: 10E8-69EB

Содержимое папки E:\v100\vasya01\one
24.03.2009  14:57    <DIR>          .
24.03.2009  14:57    <DIR>          ..
24.03.2009  13:51                28 paravas2
23.03.2009  17:48                31 pv1
                2 файлов          59 байт
                2 папок      5 137 494 016 байт с
E:\v100\vasya01\one>rename paravas2 pv2
E:\v100\vasya01\one>dir
Том в устройстве E не имеет метки.
Серийный номер тома: 10E8-69EB

Содержимое папки E:\v100\vasya01\one
24.03.2009  14:58    <DIR>          .
24.03.2009  14:58    <DIR>          ..
23.03.2009  17:48                31 pv1
24.03.2009  13:51                28 pv2
                2 файлов          59 байт
                2 папок      5 137 494 016 байт с
E:\v100\vasya01\one>

```

1. Текущий каталог one.
2. Перемещаем файл paravas2 из two в one с тем же именем.
3. Переименовываем paravas2 в pv2 при помощи команды rename.

Переместите файлы paravas3 и paravas4 из каталога two в каталог one с именами pv3 и pv4, предъявите результат преподавателю, после чего полностью очистите домашний каталог.

1. На имеющемся диске создать директорию по первым буквам своих фамилий, а затем в своей директории создаём каталог.

Где d – имя диска, md - создание директории, ud_MS-директория (имя).

D:

md ud_MS

md d:\ud_MS\ud1

md d:\ud_MS\ud1\ud11

md d:\ud_MS \ud1\ud12

md d:\ud_MS \ud1\ud13

md d:\ud_MS\ud2

md d:\ud_MS\ud2\ud21

md d:\ud_MS\ud2\ud22

md d:\ud_MS\ud2\ud22\ud221

md d:\ud_MS\ud2\ud22\ud222

md d:\ud_MS\ud2\ud22\ud223

md d:\ud_ MS \ud2\ud23

```
md d:\ud_ MS\ud3
```

```
md d:\ud_ MS\ud3\ud31
```

```
md d:\ud_ MS\ud3\ud32
```

```
md d:\ud_ MS\ud3\ud33
```

```
md d:\ud_ MS\ud3\ud33\ud331
```

2. Прodelать смену директории с помощью команды cd в директории ud MS.

```
cd ud_ MS
```

```
ud_ MS >cd ud3
```

```
ud_ MS \ud3>md ud332
```

```
ud_ MS \ud3>md ud333
```

```
ud_ MS \ud3>cd ..
```

```
ud_ MS >cd ..
```

```
cd ud_ MS
```

3. Для просмотра созданного каталога прописываем операцию > tree.

```
D:\ud_ MS >tree
```

4. Произвести удаление из каталога директории ud331, ud332 и ud333 при помощи команды rd-удаление директории:

```
D:\ud_ MS >
```

```
D:\ud_ MS >cd ud3
```

```
D:\ud_ MS \ud3>cd ud33
```

```
D:\ud_ MS \ud3\ud33>rd ud331
```

```
D:\ud_ MS \ud3\ud33>rd ud332
```

```
D:\ud_ MS \ud3\ud33>rd ud333
```

```
D:\ud_ MS \ud3\ud33>tree
```

1. Дать определение командной строки, команды.
2. Составить перечень команд, используемых в работе с их определением.
3. Прodelать пункты 1 – 4. Каждый шаг необходимо фиксировать принтскрином.
4. Вставить свои картинки (принтскрины) в отчет и дать обоснование происходящим действиям (т.е. объяснить – что делали?)

Вопросы к защите:

1. Что такое интерфейс и какой его вид используется в данной работе?
2. Составить перечень команд для решения подобных задач в различных ОС.

Практическое занятие №4.

Тема: Пакетные и командные файлы.

Цель: Научиться создавать командные файлы.

Ход работы:

Прежде чем мы приступим к созданию пакетных файлов для установки любимых приложений и изменений в реестре, вы, вероятно, захотите узнать, что же это за файлы и как ими пользоваться.

Пакетные файлы (*.cmd или *.bat) могут быть выполнены на различных стадиях установки Windows. Использование пакетных файлов помогает автоматизировать выполнение нескольких заданий. При этом никакого вмешательства пользователя не требуется.

Эти файлы могут быть выполнены из *cmdlines.txt*, *syspack.inf*, RunOnceEx раздела в реестре, или из секции [GuiRunOnce] в файле *winnt.sif*. Мы подробнее разберем различные варианты чуть позже.

Типичное содержимое пакетного файла

Пакетные файлы поддерживают все команды, которые могут быть выполнены из командной строки. Чтобы увидеть командную строку достаточно сделать следующее: Пуск->Выполнить->cmd->ОК

Давайте попробуем сделать простенький пакетный файл. Откройте Блокнот и скопируйте туда следующий текст:

```
title Batch File Testing
echo Hello World
echo.
echo Starting Notepad
start notepad
echo.
echo Starting Wordpad
start Wordpad
echo.
pause
exit
```

Сохраните файл с любым именем и в любом месте. Убедитесь, что вы назначили расширение *.cmd (а не .txt, которое Блокнот присвоит по умолчанию). Теперь запустите ваш файл. Вы увидите, что он назначит окну имя "Batch File Testing", запустит Блокнот и WordPad, попросит нажать любую клавишу для продолжения работы и закроет окно.

Что означают команды, использованные в данном файле?

| | |
|---------------------|--|
| @echo off | Скрывает строку C:\> и все команды, используемые в пакетном файле |
| echo. | Создает пробел между строками |
| echo message | Отображает сообщение в окне, исключительно с информационными целями |
| title | Название окна |
| start | Запускает приложение или установочный процесс |
| pause | Отображает сообщение "Press any key to continue..." ("Нажмите любую клавишу для продолжения...") |
| exit | Закрывает командную строку |

Когда вы начнете добавлять приложения для автоматической установки с инсталляционного диска XP, вы увидите, что в большинстве случаев команды будет выглядеть примерно так:

```
start /wait %systemdrive%\install\some_application\setup.exe /ключ /еще ключ
```

start запустит установку приложения, а **/wait** дождется ее окончания перед тем, как выполнить установку следующего приложения. Использование **/wait** **очень важно**, т.к. в противном случае одновременно запустится установка сразу нескольких приложений и конфликты неизбежны.

Вы всегда можете обратиться к [странице Примеров](#), где можно увидеть различные примеры применения пакетных файлов.

Изменение цветовой гаммы

Во время установки окно, с выполняющимися командами, будет выглядеть так. Несколько мрачно, но это можно разнообразить, сделав фон и шрифт цветными при помощи команды **color xx**. Атрибуты цветов задаются в виде двух шестнадцатеричных цифр - первая задает цвет фона, а вторая определяет цвет текста. Каждая цифра может иметь следующие значения:

| | |
|-------------|--------------------|
| 0 = Черный | 8 = Серый |
| 1 = Синий | 9 = Светло-синий |
| 2 = Зеленый | A = Светло-зеленый |
| 3 = Голубой | B = Светло-голубой |
| 4 = Красный | C = Светло-красный |
| 5 = Лиловый | D = Светло-лиловый |
| 6 = Желтый | E = Светло-желтый |
| 7 = Белый | F = Ярко-белый |

Таким образом, для ярко-белого шрифта и голубого фона вы используете **color 9F**.

Чтобы посмотреть, как это выглядит на практике, скопируйте текст в блокнот и сохраните файл как *colors.cmd*.

```
cls
@echo off
Title Graphics group is being installed now
color 9F
ECHO Adobe Photoshop 7.0
ECHO ACDSSee 6.0.3
PAUSE
Title Internet group is being installed now
color 57
ECHO Stay Connected! 3.5
ECHO MSN Messenger 6.2
PAUSE
EXIT
```

Примечание: команда **PAUSE** дана исключительно для того, чтобы можно было рассмотреть все получше. В реальном файле установка пойдет без остановки. В данном примере подается идея группировки приложений по типу (Графика, Интернет) и назначения им различных цветовых групп. Чтобы очистить экран перед каждой группой можно вставлять **@CLS** перед строкой Title.

cmdlines.txt, svcpack.inf и [GuiRunOnce]

Пакетные файлы могут быть запущены из *cmdlines.txt* или *svcpack.inf*, которые будут исполнены на T-13/T-12 этапе установки Windows (T-13 означает 13 минут до окончания установки). *cmdlines.txt* удобен для таких задач как Добавление Пользователей, или для импортирования заранее подготовленных ключей реестра **HKKEY CURRENT USER** в профиль по умолчанию (Default Profile). Подробнее о *cmdlines.txt* можно почитать в соответствующей статье в разделе Справочник.

Метод *svcpack.inf*, как правило, используется для установки обновлений (хотфиксов). Однако ничто не мешает рассматривать пакетный файл как еще один хотфикс.

Пример пакетного файла, запущенного из [GuiRunOnce] (скриншот).

Как вы видите, здесь используется комбинация **ECHO** и **ECHO Message**. Делать это не обязательно, но рекомендуется, т.к. дает возможность видеть, как происходит процесс инсталляции приложений в реальном времени. [GuiRunOnce] запускается при первом входе в систему.

Пример содержимого пакетного файла:

```
@echo off
TITLE Windows XP SP2 - Unattended Installation
ECHO.
ECHO Over the next few minutes you will see automated installations
ECHO of various software applications, and registry tweaks being
ECHO implemented.
ECHO.
ECHO Removing Wallpapers and Screensavers...
DEL "%systemroot%\*.bmp"
```

```

DEL "%systemroot%\Web\Wallpaper\*.jpg"
DEL "%systemroot%\system32\dlcache\*.scr"
DEL "%systemroot%\system32\*.scr"
ECHO.
ECHO Removing useless shortcuts...
DEL "%systemdrive%\Documents and Settings\All Users\Start Menu\Windows
Update.lnk"
DEL "%systemdrive%\Documents and Settings\All Users\Start Menu\Set
Program Access and Defaults.lnk"
DEL "%systemdrive%\Documents and Settings\All Users\Start Menu\Windows
Catalog.lnk"
ECHO.
ECHO Installing TweakUI 2.10 Powertoy
ECHO Please wait...
start /wait %systemdrive%\Install\TweakUI.msi /qn
ECHO.
ECHO Applying Registry Tweaks...
REGEDIT /S %systemdrive%\Install\RegTweaks.reg
ECHO.
ECHO Deleting ASP.NET User Account created by .NET Framework 1.1...
net user aspnet /delete
ECHO.
EXIT

```

Он производит удаление обоев и скринсэйверов, устанавливаемых по умолчанию системой, а также некоторых ярлыков. Затем устанавливает TweakUI, импортирует ключи реестра и удаляет учетную запись ASP.NET, созданную при установке .NET Framework. Более содержательный файл вы можете посмотреть на странице примеров.

Любой командный пакетный файл имеет расширение bat и может вызываться с параметрами, указанными через пробел (например: class.bat Иванов Петров Сидоров), доступ к параметрам в командном файле осуществляется следующим образом: %номер_параметра (например, %1 - Иванов, %2 - Петров)

Основные команды пакетных командных файлов.

| Название команды | Синтаксис команды |
|-------------------------------|---------------------|
| Комментарии в командном файле | REM <любые символы> |
| Вывод сообщения на экран | ECHO <сообщение> |

| | |
|--|---------------------------------|
| Выключение вывода команд на экран | @ECHO OFF |
| Приостановка выполнения командного файла | PAUSE <сообщение> |
| Переходы в командном файле | GOTO <метка> ... :<метка> |
| Проверка условий в командном файле | IF <условие> <команда> |

Условие это одно из ниже приведенных выражений:

-строка1==строка2 (например, -%1==Петров)

EXIST имя файла (условие ИСТИНА, когда файл существует)

NOT EXIST имя файла (условие ИСТИНА, когда файл не существует)

Пример: командный файл lesson0.bat

@ECHO OFF

ECHO Характеристика класса

REM Проверяем введенный параметр - это Петров? если да, идем на метку P

IF -%1==Петров GOTO P

REM Проверяем введенный параметр - это Иванов? если да, идем на метку I

IF -%1==Иванов GOTO I

REM Если ни Иванов, ни Петров, то идем в конец файла

GOTO exit

:P

REM Выводим характеристику Петрова

ECHO Хулиган

GOTO exit

:I

REM Выводим характеристику Иванова

ECHO Отличник

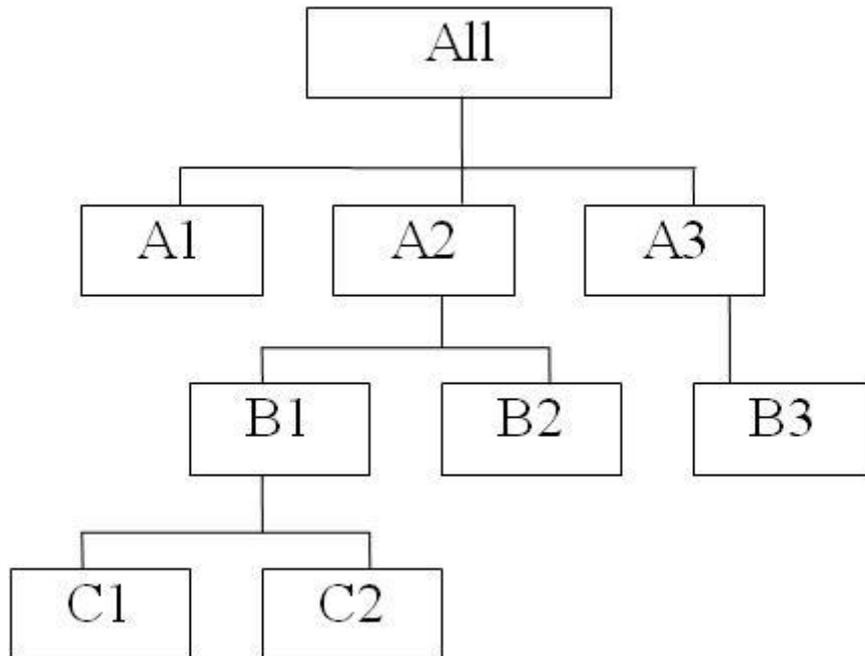
GOTO exit

REM Конец командного файла

:exit

Задание на лабораторную работу

1. В корневом диске с помощью командной строки создать дерево каталогов как показано на рисунке.



2. В каталоге All с помощью командной строки создать следующие файлы: 1.txt, 2.txt, 3.txt, lab1.pas, lab2.pas, face.bmp, foot.bmp, arm.bmp, beatles.mp3, rock.mp3.

3. Создать пакетный командный файл lesson1.bat в каталоге All, описать следующую последовательность команд и прокомментировать каждую из них.

3.1. Удалить каталог B3, в каталоге A1 создать подкаталоги B4 и B5 и удалить каталог B2.

3.2. Скопировать файл lab1.pas в каталог A2 и переименовать его в файл lab3.pas.

3.3. Очистить экран от служебных записей.

3.4. Вывести на экран поочередно информацию, хранящуюся во всех текстовых файлах в каталоге All.

3.5. Скопировать все графические и текстовые файлы из каталога All в каталог C2.

3.6. Объединить все текстовые файлы, хранящиеся в каталоге C2, в файл man.txt и вывести его содержимое на экран.

3.7. Скопировать файл man.txt в директорию A1.

3.8. Вывести на экран информацию о структуре каталога All.

4. Создать пакетный командный файл lesson2.bat в каталоге All, решающий определенную задачу:

- Создать командный файл при вызове, которого указать фамилию запускающего. Если запустил Иванов, то вывести строку: "Нет доступа", если запустил Петров, то вывести строку: "Не сегодня". Для всех других пользователей вывести информацию каталоге, из которого был запущен bat-файл.

- Создать командный файл, который бы выводил на экран все файлы заданного расширения. Если расширение при вызове командного файла не задано вывести: "Расширение не введено", если в текущем каталоге файлов с таким расширением нет, то вывести: "Файлов не найдено".

- Создать командный файл, который будет помогать в проведении метеосводок. При вызове командного файла указать кодовое слово: "снег", "дождь", "солнце". Если кодовое слово "снег", то вывести сообщение, что погода будет морозная от -15 до -20. Если кодовое слово "дождь", то погода будет слякотная температура около 0. Если кодовое слово солнце, на улице будет солнечно температура +20. Если параметр не указан, то вывести: "Метеорология бессильна".

Практическое занятие №5

Тема: Пакетные и командные файлы.

Цель: Научится создавать командные файлы.

Ход работы:

- Создать командный файл, который бы удалял все файлы заданного расширения. Если расширение при вызове командного файла не задано вывести: "Расширение не введено", если в текущем каталоге файлов с таким расширением нет, то вывести: "Файлов не найдено".

- Создать командный файл, при вызове которого указываются фамилии студентов, выполняющих лабораторную работу. Если такой студент один, вывести на экран "Работает один". Если их двое, то вывести "две головы лучше". Если больше двоих, то "Толпа". Если никто не работает, то "Где все?".

- Создать командный файл, при вызове которого указывается модель процессора семейства Intel Pentium (I, II, III, IV). Каждой такой модели сопоставить тактовую частоту и вывести её на экран.

- Создать командный файл, при вызове которого указывается марка автомобиля. Каждой такой марке сопоставить максимальную скорость и вывести её на экран. Если такой марки нет, то вывести "Не существует, Вы ошиблись".

- Создать командный файл, который выполнял следующие действия: при вызове с параметром 1 - создавал каталог NEW, с параметром 2 - выводил информацию о текущем каталоге, с параметром 3 - информацию о версии системы, с любым другим параметром - фразу "такая команда не определена".

- Создать командный файл, при вызове которого указываются фамилии двух студентов. Если первый Иванов, то вывести фразу "В классе будет шумно", если второй Петров, то вывести фразу "В аудитории будет гам". Если первый Иванов и второй Петров, то вывести фразу "Аудитории вообще не будет". Если их нет, то вывести фразу "Все спокойно".

- Создать командный файл, с помощью команды PAUSE реализовать следующие действия: вывести на экран "Отформатировать диск C:\? Да/Нет" и дождаться реакции пользователя, после любого действия пользователя вывести на экран "Вы точно хотите отформатировать диск C:\ ? Да/Нет" и дождаться реакции пользователя, далее вывести "Диск C:\ - отформатирован. До встречи."

Тема: Установка Windows Server 2008

1. Запустить VirtualBox
2. Установить полную версию операционной системы Windows Server 2008 Standard аналогично установке Windows 7 (Практическая работа II: Windows 7)
 - a) Образ диска для установки *SW_DVD5_Windows_Svr_DC_Ent_Std_2008_Russian_32bit_MLF_X14-26782.iso* находится в каталоге ISO
 - b) Настройки программы VirtualBox для операционной системы Windows Server 2008 использовать по умолчанию
 - c) Задать пароль Администратора: P@ssw0rd

Настройка внутренней сети

1. В окне VirtualBox выделить виртуальную машину Windows 7, нажать Настройка (Settings)
2. Настройка сети (Network), Адаптер 1 (Adapter 1), Присоединен к внутренней сети (Internal Network), Имя сети (Name): miigaik
3. Дополнительно (Advanced), Режим разрешения (Promiscuous Mode): Разрешить все (Allow All), Нажать ОК
4. Повторить пункт 2-4 для виртуальной машины Windows Server 2008

Настройка Windows 7

1. Запустить виртуальную машину Windows 7
2. Изменить имя компьютера на win7 и перезагрузить
3. Изменить настройки сетевого адаптера, использовать следующий IP-адрес (TCP/IPv4):
 - a) IP-адрес: 192.168.0.101
 - b) Маска подсети: 255.255.255.0
 - c) Основной шлюз: 192.168.0.1
 - d) Предпочитаемый DNS-сервер: 192.168.0.1
 - e) В окне VirtualBox поставить виртуальную машину Windows 7 на паузу: Machine - Pause

Настройка Windows Server 2008

1. Запустить виртуальную машину Windows Server 2008
2. Пароль Администратора: P@ssw0rd
3. Изменить имя компьютера на srv2008 и перезагрузить
4. Изменить настройки сетевого адаптера, использовать следующий IP-адрес (TCP/IPv4):
 - a) IP-адрес: 192.168.0.1
 - b) Маска подсети: 255.255.255.0
 - c) Изменить дополнительные параметры общего доступа:
 - Включить сетевое обнаружение
 - Включить общий доступ к файлам и папкам

Установка DNS и веб-сервера

1. Запустить «Диспетчер сервера» - Роли - Добавить роли:

- a) DNS сервер
- b) Веб-сервер (IIS)
2. Пройти мастер установки по умолчанию и завершить установку
3. Перейти в виртуальную машину Windows 7
4. Запустить Internet Explorer и перейти на сайт [http : //srv2008/](http://srv2008/)
5. Откроется приглашение «Добро пожаловать IIS»

Настройка веб-сервера

1. Перейти в виртуальную машину Windows Server 2008
2. Запустить «Блокнот»
3. Ввести содержимое простой веб-страницы:

```
<html>
<head>
<title>Простейшая веб-страница</title>
</head>
<body>
Hello World!
</body>
</html>
```
4. Сохранить файл как html документ hello.html в папке *C:\inetpub\wwwroot*
5. Перейти в виртуальную машину Windows 7
6. Запустить Internet Explorer и перейти на сайт [http : //srv2008/hello.html](http://srv2008/hello.html)
7. Откроется ваша веб-страница

Тема: **Windows server 2008.Настройка FTP-сервера.**

1. Запустить «Диспетчер сервера» - Роли - Веб-сервер (IIS) - Добавить службы ролей:
 - a) Выбор служб ролей: FTP-сервер
2. Пройти мастер установки по умолчанию и завершить установку
3. На диске C: создать папку dir
4. Запустить: Пуск - Диспетчер служб IIS
5. Развернуть «SRV2008», нажать правой кнопкой мыши «сайты», выбрать «Добавить FTP-сайт...»:
 - a) Имя FTP-сайта: ftp-miigaik
 - b) Физически путь: *C: \dir*
6. Параметры привязки и SSL:
 - a) IP-адрес: 192.168.0.1
 - b) Порт: 21
 - c) Запускать FTP-сайт автоматически
 - d) Без SSL
7. Сведения о проверке подлинности и авторизации:
 - a) Проверка подлинности: Анонимный
 - b) Разрешить анонимный доступ к: Анонимные пользователи
 - c) Разрешения: Чтение
8. Запустить командную строку, выполнить команду: *netsh advfirewall set global StatefulFtp enable*
9. Перезапустить операционную Windows Server 2008

Доступ к FTP-серверу

1. Перейти в виртуальную машину Windows 7
2. Запустить: Пуск - Компьютер, введите вручную и пройдите по адресу: ftp : //srv2008/
3. Прочитать содержимое Вашего FTP-сервера, попробуйте записать информацию на FTP-сервер
4. Перейти в виртуальную машину Windows Server 2008
5. Запустить «Диспетчер служб IIS», выделите сайт ftp-miigaik:
6. Проверка подлинности FTP:
 - a) Анонимный доступ: Включено
 - b) Обычная проверка подлинности: Включить
7. Правила авторизации FTP - Добавить разрешающее правило
 - a) Указанные пользователи: Администратор
 - b) Разрешения: Чтение и Запись

Настройка Active Directory

1. Запустить «Диспетчер сервера» - Роли - Добавить роли:

- a) Доменные службы Active Directory
- b) Возможности .NET Framework 3.5.1
2. Пройти мастер установки по умолчанию
3. Закрыть этот мастер и запустите мастер установки доменных служб Active Directory (dcpromo.exe)
 - a) Использовать простой режим установки
 - b) Создать новый домен в новом лесу
 - c) Полное доменное имя корневого домена леса: miigaik.local
 - d) Режим работы леса: Windows Server 2008
 - e) Продолжить после предупреждающего сообщения
 - f) Расположение для базы данных, файлов журнала и папки SYSVOL оставить по умолчанию
 - g) Пароль: P@ssw0rd
4. После завершения работы мастера перезагрузить операционную Windows Server 2008

Ввод компьютера в состав домена

1. Перейти в виртуальную машину Windows 7
2. Запустить: Пуск - Панель управления - Система
3. Найти в окне пункт «Имя компьютера, имя домена и параметры рабочей группы» нажать пункт «Изменить параметры»
4. На вкладке «Имя компьютера» нажмите кнопку «Изменить»
5. Компьютер является членом домена: miigaik.local
 - a) Пользователь: Администратор
 - b) Пароль: P@ssw0rd
6. Добро пожаловать в домен miigaik.local, перезагрузить компьютер

Новый пользователь домена

1. Перейти в виртуальную машину Windows Server 2008
2. Запустить: Пуск - Администрирование - Active Directory - пользователи и компьютеры
3. Развернуть домен miigaik.local - Users
4. Создать нового пользователя:
 - a) Имя: Ваше Имя
 - b) Фамилия: Ваша Фамилия
 - c) Имя для входа: familiya (ваша фамилия на английском языке)
 - d) Пароль: P@ssw0rd
 - e) Запретить смену пароля пользователем
 - f) Срок действия пароля не ограничен
5. Закончить создание нового пользователя и перейдите в виртуальную машину Windows 7
6. Авторизоваться в операционной системе Windows 7 под учетной записью нового пользователя

Контрольное задание

1. Настроить FTP доступ к папке oreg в операционной системе Windows Server 2008 для пользователя Operator с разрешением на чтение и запись

2. Создать фильтрацию запросов FTP для нового сайта в Диспетчере служб IIS
3. Запретить доступ к папке для файлов с расширением .exe

Практическое занятие №8.

Тема: Создание документа с использованием языка HTML.

Цель работы:

- Научиться создавать простейший гипертекстовый документ средствами текстового редактора Блокнот.
- Научиться использовать теги форматирования шрифта и абзаца.

•

• Задание на выполнение

1. Создать файл с гипертекстовым документом:

- Запустить редактор Блокнот, ввести текст:

Приветствую Вас на моей первой web-страничке!

- Сохранить файл в созданной папке. При сохранении, в окне диалога **Сохранить как...** в строке **Тип файла:** выбрать вариант **Все файлы (*.*)**, а в строке **Имя файла** задать имя с расширением **.htm**, например **1_name.htm** (где **name** – ваше имя)
- Закрыть документ, найти его пиктограмму в окне **Мой компьютер** или в окне программы **Проводник**.
- Открыть файл. Проанализировать, с помощью какого приложения отображается файл и как выглядит введенная фраза.

2. Ввести теги, определяющие структуру html-документа:

- С помощью контекстного меню открыть файл с помощью редактора Блокнот. Ввести приведенные ниже теги, в разделе заголовка документа (между тегами **<TITLE>** **</TITLE>**) указать свою фамилию.

```
<HTML>
```

```
    <HEAD> <TITLE> Фамилия </TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

Приветствую Вас на моей первой web-страничке!

```
  </BODY>
```

```
</HTML>
```

- **Сохранить** документ под тем же именем, обновить его отображение в браузере (выполнить **Вид/Обновить** или нажать кнопку **Обновить** на панели инструментов). Проанализировать произошедшие изменения в отображении документа.

3. Отредактировать документ:

- Вызвать меню браузера **Вид/Просмотр HTML-кода** и добавить после текста **«Приветствую Вас на моей первой web-страничке!»** текст подписи:

Студент группы NNN Фамилия Имя

Сохранить документ (но не закрывать) и обновить его просмотр в браузере.

- Используя одиночный тег **
, отредактировать документ так, чтобы подпись начиналась с новой строки, а **Фамилия Имя – в следующей строке. Просмотреть в браузере новый вариант.

Внимание! После каждого изменения документ нужно сохранять, а просмотр в браузере начинать с обновления загрузки документа с помощью кнопки **«Обновить»** на панели инструментов.

4. Оформить фрагменты текста с помощью стилей **Заголовков**:

- Первую строку документа оформить стилем **Заголовок 1-го уровня** с помощью парного тега **<H1> ...</H1>**. Вторую строку оформить как **Заголовок 6-го уровня**, а третью как **Заголовок 4-го уровня**.
- Просмотреть документ в браузере, изменяя настройку отображения шрифтов (меню **Вид / Размер шрифта / Самый крупный, Средний, Мелкий и Самый мелкий**).
- Поменять стиль оформления первой строки на **Заголовок 2 уровня**, второй строки - на **Заголовок 5 уровня**, последней строки - на **Заголовок 3-го уровня**.

5. Выполнить форматирование шрифта:

- После строки **Фамилия Имя** добавить еще одну строку текста
Нас утро встречает прохладой
- Оформить фразу по приведенному ниже образцу.

Нас утро встречает прохладой

В слове УТРО все буквы должны иметь **разные цвета**. В слове ПРОХЛАДОЙ оформить буквы **ПРО** – **красным** цветом, **ОЙ** – **синим**.

- Оформить строку с подписью (**Студент группы NNN Фамилия Имя**) курсивом, размер шрифта задать относительным изменением. Использовать теги **** и **<I>**
- Просмотреть полученный документ в браузере.

6. Выполнить форматирование абзацев:

- Создать новый документ **2_name.htm**, сохранить его в той же рабочей папке.
- Ввести текст (использовать копирование текста из документа **1_name.htm**):

<HTML>

<HEAD> <TITLE> Фамилия </TITLE>

</HEAD>

<BODY>

Приветствую Вас на моей второй web-страничке!
 Монолог Гамлета

</BODY>

</HTML>

- Выровнять текст **по центру**.
- Ввести текст:

Быть иль не быть - вот в чем вопрос. Что благороднее: сносить удары неистовой судьбы - иль против моря невзгод вооружиться, в бой вступить. И все покончить разом...

- Оформить выравнивание абзаца **по ширине**.
- Ограничить абзац горизонтальными разделительными линиями сверху и снизу, используя тег <HR>.
- Скопировать монолог и разбить его на абзацы. Выровнять **по центру**.

*Быть иль не быть - вот в чем вопрос.
Что благороднее: сносить удары
Неистойвой судьбы - иль против моря
Невзгод вооружиться, в бой вступить
И все покончить разом...*

- Сохранить документ.
- Просмотреть документ в окне броузера, изменяя размер окна.

7. Выполнить оформление списков:

- Создать новый документ **3_name.htm**, сохранить его в той же рабочей папке жесткого диска.
- Ввести текст:

<HTML>

<HEAD> <TITLE> **Фамилия** </TITLE>

</HEAD>

<BODY>

Приветствую Вас на моей третьей web-страничке!

</BODY>

</HTML>

- Дополнить текст документа (между тегами <BODY>...</BODY>) следующим текстом:
Я знаю как оформлять:
Шрифты,
Заголовки,
Абзацы
- Оформить три последние строки как **список нумерованный**. Для этого использовать следующую конструкцию тегов:

 Шрифты,

 Заголовки,

 Абзацы

- Поменять оформление списка на **список маркированный**. Использовать теги ,
- Создать «смешанный» список:

Я знаю как оформлять:

1. Шрифты

- Размер
- Цвет
- Гарнитуру
- Индексы

2. Заголовки

- От 1-го до 6-го уровня

3. Абзацы

- Выравнивание
- Разрыв строк внутри абзаца
- С использованием переформатирования.

8. Предъявить результаты работы преподавателю.

•

Таблица основных тегов HTML-документа. Теги форматирования шрифта и абзаца

| Назначение | Вид тегов | Примечание |
|---------------------------------------|-----------------|---|
| Общая структура документа HTML | | |
| Тип документа | <HTML></HTML> | Начало и конец документа |
| Имя документа | <HEAD></HEAD> | Не отображается браузером |
| Заголовок | <TITLE></TITLE> | Содержимое строки заголовка окна браузера |
| Тело документа | <BODY></BODY> | Содержимое WEB-страницы |
| Структура содержания документа | | |

| | | |
|--|--|---|
| Внутренние заголовки различного уровня | <H№> текст </H№> | Где № – номер уровня заголовка (от 1 до 6). Например, <H1>...</H1> - заголовок 1-го уровня. |
| Заголовок с выравниванием | <H№ ALIGN="LEFT CENTER RIGHT"> текст </H№> | LEFT - по левому краю, CENTER - по центру, RIGHT - по правому краю. |

Форматирование абзацев

| | | |
|--|---|---|
| Создание абзаца (параграфа) | <P> текст </P> | Абзацы отделяются двойным межстрочным интервалом |
| Перевод строки внутри абзаца | | Одиночный тег |
| Выравнивание абзаца | <P ALIGN="LEFT">текст </P> <P ALIGN="CENTER">текст </P> <P ALIGN="RIGHT"> текст</P> <P ALIGN="JUSTIFY"> текст </P> | LEFT - по левому краю CENTER - по центру RIGHT - по правому краю JUSTIFY – по ширине |
| Разделительная горизонтальная линия между абзацами | <HR SIZE=«?»> | Одиночный тег. «?» - толщина линии в пикселях. Толщину линии можно не указывать. |

Форматирование шрифта

| | | |
|--------------------|-------------------------------|--|
| Жирный | текст | Жирный |
| Курсив | <I> текст </I> | <I> <i>Курсив</i> </I> |
| Подчеркнутый | <U> текст </U> | <U> <u>Подчеркнутый</u> </U> |
| Перечеркнутый | <S> текст </S> | <S> Перечеркнутый </S> |
| Увеличенный размер | <BIG> текст </BIG > | |
| Уменьшенный размер | <SMALL> текст </SMALL> | ^{Верхний индекс} _{Нижний индекс} |
| Верхний индекс | ^{текст} | |
| Нижний индекс | _{текст} | |
| Размер шрифта | текст | ?- значения от 1 до 7 или относительное изменение (например, +2) |

| | | |
|-----------------------|---|---|
| Базовый размер шрифта | <BASEFONT SIZE=?> | Одиночный тег ? – размер от 1 до 7; по умолчанию равен 3 и задается для всего документа в целом |
| Гарнитура шрифта | • текст | Текст оформляется первым, установленным на компьютере шрифтом из списка названий |
| Цвет шрифта | • текст | Цвет задается либо ключевым словом, либо шестнадцатеричным кодом с символом # RED –красный, #FF0000 – шестнадцатеричный код – красного цвета |

Создание списков

| | | |
|----------------|----------------------------|---|
| Нумерованный | элементы списка | Элемент списка 1 Элемент списка 2 Элемент списка 3 |
| Маркированный | элементы списка | |
| Элемент списка | элементы списка | |

Таблица основных цветов

| Цвет | Color's name | Шестнадцатеричный код цвета | | |
|----------------|--------------|-----------------------------|-------|------|
| | | Red | Green | Blue |
| Черный | black | 00 | 00 | 00 |
| Темно-синий | navy | 00 | 00 | 80 |
| Голубой | blue | 00 | 00 | FF |
| Зеленый | green | 00 | 80 | 00 |
| Темно-зеленый | teal | 00 | 80 | 80 |
| Салатный | lime | 00 | FF | 00 |
| Бледно-голубой | aqua | 00 | FF | FF |
| Вишневый | maroon | 80 | 00 | 00 |
| Фиолетовый | purple | 80 | 00 | 80 |
| Оливковый | olive | 80 | 80 | 00 |

| Цвет | Color's name | Шестнадцатеричный код цвета | | |
|--------------|--------------|-----------------------------|-------|------|
| | | Red | Green | Blue |
| Серый | gray | 80 | 80 | 80 |
| Светло-серый | silver | C0 | C0 | C0 |
| Красный | red | FF | 00 | 00 |
| Лиловый | fushsia | FF | 00 | FF |
| Желтый | yellow | FF | FF | 00 |
| Белый | white | FF | FF | FF |

Практическое занятие №9.

Тема: Создание документа с использованием языка HTML.

Цель: Вставка в HTML-документ рисунков. Создание закладок и гиперссылок

1. Скопировать из Интернета или какой-либо папки в личную папку файлы три графических файла (например, **Arrows1.wmf**, **Arrows2.wmf**, **Arrows3.wmf**).

*Убедиться, что созданные ранее документы **1_name.htm**, **2_name.htm** и **3_name.htm** также находятся в вашей папке на жёстком диске.*

2. Вставка рисунков в документ.

- Открыть в Блокноте документ **2_name.htm**.
- Вставить рисунок **Arrows1.wmf** в начало документа **2_name.htm**. Для вставки использовать тег **IMG** с параметрами **WIDTH** и **HEIGHT** для установки размеров рисунка 50 пикселей по горизонтали и по вертикали.
- Сохранить документ под именем **4_name.htm**.
- Просмотреть в браузере полученный результат.
- Ввести в тег рисунка параметр **ALIGN** для выравнивания рисунка по правому краю. Просмотреть результат в браузере.
- Вставить рисунок **Arrows2.wmf** в конец документа **4_name.htm** перед, подобрать тип выравнивания рисунка на свое усмотрение. Установить размер рисунка 100 пикселей по горизонтали и по вертикали. С помощью параметра **ALT** создать всплывающую подсказку «**Рисунок 2**», появляющуюся при наведении курсора мыши на рисунок.
- Просмотреть в браузере полученный результат.

3. Создание гиперссылок и закладок.

- В документе **3_name.htm** закрепить гиперссылки за следующими элементами списка:

За словом **Шрифт** – гиперссылка на документ **1_name.htm**.

За словом **Заголовки** – на документ **1_name.htm**.

За словом **Абзацы** - на документ **2_name.htm**.

- Создать закладку в документе **1_name.htm** перед фразой «**Нас утро встречает прохладой**». Дать ей имя «**Morning**».
- Изменить первую гиперссылку (слово **Шрифт**) так, чтобы она указывала на закладку «**Morning**» в документе **1_name.htm**.
- Создать закладку в начале текущего документа **3_name.htm**. Присвоить ей имя «**Hello**».
- Изменить вторую гиперссылку (на слове **Заголовки**), определив для неё переход в начало текущего документа на установленную закладку «**Hello**».
- Создать закладку в документе **2_name.htm** перед фрагментом монолога. Присвоить ей имя «**Mono**».
- Установить на слово **переформатирования** гиперссылку на закладку «**Mono**».
- Проверить правильность переходов по всем гиперссылкам.

4. Закрепить гиперссылки за графическими файлами:

- Отредактировать тег вставки рисунка **Arrows1.wmf**, ввести в тег атрибут **ALT** для отображения текста подсказки «**Вернуться**». Просмотреть в браузере как реагирует рисунок на наведение курсора мыши.
- Закрепить за рисунком **Arrows1.wmf** в документе **4_name.htm** гиперссылку на документ **3_name.htm**. Выполнить переход между документами.

5. Предъявить результат преподавателю.

- **Основные теги вставки рисунков, закладок и гиперссылок**

Вставка изображений

| | | |
|--|--|---|
| Вставка графического файла | | <i>Пример:</i> |
| Выравнивание картинки относительно текста | | |
| Вывод текста всплывающей подсказки при наведении курсора мыши на рисунок | | |

Вставка ссылок

| | | |
|---------------------------------------|---|--|
| Ссылки на другую страницу | текст | Ссылка1 |
| Ссылка на закладку в другом документе | текст | На главную страницу |
| Ссылка на закладку в том же документе | текст | Ссылка2 |
| Определить закладку | текст | |

Цвет фона, текста и ссылок

| | | |
|------------------------|----------------------------------|--|
| Фоновая картинка | <BODY BACKGROUND="файл рисунка"> | <BODY BACKGROUND ="grafica.gif" TEXT="black" (черный) LINK="#FF0000" (красный) VLINK="#FFFF00" (желтый) ALINK="#FFFFFF" (белый) </BODY> |
| Цвет фона | <BODY BGCOLOR="#\$\$\$\$\$"> | |
| Цвет текста | <BODY TEXT="#\$\$\$\$\$"> | |
| Цвет ссылки | <BODY LINK="#\$\$\$\$\$"> | |
| Цвет пройденной ссылки | <BODY VLINK="#\$\$\$\$\$"> | |
| Цвет активной ссылки | <BODY ALINK="#\$\$\$\$\$"> | |



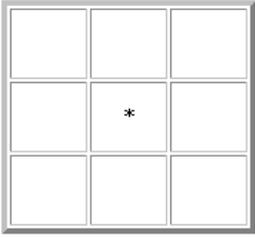
Практическая занятие № 10.

Тема: Создание документа с использованием языка HTML

Цель работы:

- Научиться использовать таблицы для оформления WEB-страниц

1. Создать таблицу по приведенному образцу, сохранить документ под именем **tabl_name.htm**.
Сверху над таблицей разместить заголовок **Таблица №1**



При отображении таблицы в браузере должны удовлетворяться следующие условия:

- таблица должна быть выравнена по центру и быть правильной (симметричной) формы;
- в центральной ячейке поместить символ * (звездочка), остальные ячейки должны быть пустыми.

Примечание. Для отображения пустых ячеек в них нужно поместить символный примитив пробела ** **;

2. В этом же документе создать копию таблицы №1, ввести заголовок **Таблица №2** и модифицировать ее:

- В центральной ячейке разместить рисунок **Arrows3.wmf**
- «Раскрасить» все остальные ячейки в различные цвета.

3. Создать еще одну копию таблицы – **Таблица №3** и отредактировать теги таблицы так, чтобы она соответствовала приведенному ниже образцу.



Примечание. Для объединения ячеек в тегах **<TD>** необходимо использовать параметры **colspan=** и **rowspan=**

4. Создать новый HTML-документ - **rasp_name.htm** с расписанием занятий.

- Документ должен начинаться заголовком **Расписание занятий гр. NNN на весенний семестр 2018 г.**
- Первая строка таблицы должна быть оформлена как заголовки полей (с использованием тегов **<TH>**).
- Таблица по ширине должна занимать полный размер окна. Ширину отдельных столбцов задать в относительных единицах (в %), с тем, чтобы при изменении ширины окна пропорции таблицы сохранялись.

| День недели | Время | Предмет | Преподаватель | Аудитория |
|-------------|-------------|------------------|--------------------|-----------|
| Понедельник | 8:30-10:05 | Математика (лек) | доц. Иванов А.А. | 320 |
| | 10:15-11:50 | Математика (пр) | преп. Петрова И.А. | 302 |

| | | | | |
|---------|-------------|---------------|--------------------|-----|
| | 12:30-14:05 | Физика (лаб) | доц. Сидоров О.И. | 307 |
| Вторник | 8:30-10:05 | История (лек) | проф. Громова О.А. | 310 |
| | 10:15-11:50 | История (сем) | преп. Попов М.А. | 302 |
| | 12:30-14:05 | Физика (лаб) | доц. Сидоров О.И. | 307 |
| ... | ... | ... | ... | ... |

- Просмотреть созданный документ в браузере при различных размерах окна и различных настройках размера шрифта.

5. Сохранить файл с расписанием под именем **rasp_menu_name.htm** и модифицировать его.

6. После заголовка создать таблицу, состоящую из одной строки меню с названиями дней недели.

Расписание

| | | | | | |
|------------------------|-------------------------|-----------------------|-------------------------|-----------------------|-----------------------|
| Понед. | Вторник | Среда | Четверг | Пятн. | Субб. |
|------------------------|-------------------------|-----------------------|-------------------------|-----------------------|-----------------------|

7. В таблице с расписанием установить закладки на названия дней недели.

8. В таблице меню создать гиперссылки на соответствующие дни недели.

9. Выполнить цветовое оформление каждой ячейки меню.

10. Проверить правильность выполнения переходов по гиперссылкам.

11. Создать группу web-страниц, объединенных меню:

- На рабочем диске создать папку **My_raspisanie** для размещения файлов расписания.
- Поместить расписание на каждый день недели и таблицу с меню в отдельные файлы. Имена файлов: **menu.htm** – для главной страницы, названия дней недели – для остальных. Все документы разместить в папке **My_raspisanie**.
- Отредактировать гиперссылки меню так, чтобы по ним выполнялись переходы на соответствующий документ.
- В конце каждого файла с расписанием на день организовать гиперссылку для возврата в главный документ с меню.
- Оформить фон каждого дня недели собственным цветом, совпадающим с цветом ячейки таблицы меню.

12. Предъявить результат преподавателю.

-
-
-

- Теги оформления таблиц

| Определить таблицу | <TABLE></TABLE> | Пример | | | | |
|--|---|---|-------|------|--------------|------|
| Окантовка таблицы | <TABLE BORDER =""?"" </TABLE> | <TABLE border=""1"" align =«CENTER» width=«50%» > | | | | |
| Строка таблицы | <TR> </TR> | <TR> | | | | |
| Выравнивание | <TR ALIGN=left right center middle bottom > | <TH >Товар</TH> <TH>Цена</TH> | | | | |
| Ячейка таблицы | <TD></TD> | </TR> | | | | |
| • Выравнивание по горизонтали | <TD ALIGN=LEFT RIGHT CENTER> | <TR> <TD>Радиотелефон</TD> | | | | |
| Выравнивание по вертикали | <TD VALIGN = TOP MIDDLE BOTTOM> | <TD>2000 </TD> </TR> | | | | |
| Установка ширины ячейки (в пикселях или %) | <TD WIDTH=«?»> | </TABLE> <table border="1"><thead><tr><th>Товар</th><th>Цена</th></tr></thead><tbody><tr><td>Радиотелефон</td><td>2000</td></tr></tbody></table> | Товар | Цена | Радиотелефон | 2000 |
| Товар | Цена | | | | | |
| Радиотелефон | 2000 | | | | | |
| Заливка цветом ячейки | <TD BGCOLOR = «# цвет»> </TD> | <TD BGCOLOR = «#FF0000»> </TD> красный цвет | | | | |
| Заголовок столбца или строки | <TH>текст </TH> | Текст в ячейке выравнивается по центру, устанавливается жирный шрифт | | | | |

Практическое занятие №11.

Тема: Создание документа с использованием языка HTML.

Запустить программу **Проводник**. В своей папке создать папку лабораторной работы **HTML-2**. 2. Запустить браузер **Internet Explorer**. 3. Установить пустую домашнюю веб-страницу (about-blank), если она не

установлена, на вкладке **Общие (Сервис/Свойства обозревателя)**, а затем обновить (кнопка **Домой**). 4. Открыть содержимое пустой веб-страницы в Блокноте (**Вид/В виде HTML**).

Вставить между тэгами <HTML></HTML> структурные тэги области заголовка и тела страницы. *Для справки открыть предварительно скопированный в свою папку файл .../Tutorial.htm*. 5. Сохранить модифицированную веб-страницу в папке **HTML-2** как файл с именем **Web-страница ФамилияИмя.htm**.

Внимание! Во избежание потери данных при сбоях формируемый файл время от времени сохранять в той же папке. 6. Перейти в окно браузера, открыть файл (**Файл/Открыть/Обзор**). 7. Перейти в Блокнот. Вставить между тэгами области заголовка тэги <TITLE></TITLE>, между ними написать название страницы. Сохранить файл. 8. Перейти в браузер, обновить изображение страницы (**Вид/Обновить** или кнопка **Обновить** или **F5**).

Посмотреть, где в окне браузера отобразилось название страницы. *Внимание! Все дальнейшие изменения просматривать таким же образом.* 9. В открывающемся тэге области основного текста страницы (тела веб-документа)

задать цвет фона и текста. Для задания цвета использовать текстовый формат: red, green, blue, yellow, black, white и т.д. 10.

Между тэгами области основного текста вставить заголовки 1-го уровня: **Фамилия Имя**; 2-го уровня: **Биографическая справка, Профессиональные интересы, Хобби**.

11. В разделе **Биографическая справка** вставить текст, задавая переход на новую строку: *Дата рождения: дд.мм.гг., Место Рождения: ..., Адрес: г. ... ул. ...д. ..., Телефон:000-00-00, E-mail: aaa@bbb.ccc.*

12. В разделе **Профессиональные интересы** написать текст, отформатировав его аналогичным образом, например: *Работаю: ...Учусь на факультете... Специальность: ... Специализация: ... Владею иностранным языком: ...Имею опыт: ... Особые умения: ... Хочу получить работу: ...* 13. В разделе **Хобби** создать несколько разделов в виде маркированного списка, например:

Спорт;

Искусство;

Рукоделия. 14. Запустить Word. Создать в своей папке файлы: *Спорт.html, Искусство.html,*

Рукоделия.html, кратко написав в них о своих увлечениях. 15. Слова раздела Хобби сделать гиперссылками на одноименные файлы. 16. Сохранить полученную страницу. Закрыть созданные файлы. Проверить работоспособность ссылок в браузере. 17. В конце раздела вставить картинку, предварительно выбрав или в папке Мои документы/Мои рисунки (или Пуск/Найти) и сохранив как файл в своей папке. 18. Запустить графический редактор. Нарисовать стилизованный автопортрет и

сохранить его в своей папке. Вставить это изображение после заголовка 1-го уровня, указав параметры размеров, выравнивание и др. 19. Сохранить последний раз полученную страницу. Предъявить работу преподавателю.

Создание простейшего мини-сайта (HTML-3)
С помощью инструмента, заданного преподавателем (редактор Блокнот) создать мини-сайт – совокупность связанных файлов (веб-страниц).

Структура сайта следующая:

Файл poet.htm
Заглавие документа
(гиперссылка на Заглавие

Текста2) ()

Таблица:

Портрет _____

Ф.И.Тютчева Текст1

(файл photo1.jpg) (файл article) (гиперссылка на

n-т Мерзлякова

Файл photo2.gif

Портрет А.Ф.Мерзлякова

(файл photo2)

Заглавие 2-го уровня

(гиперссылка на Текст3)

Текст2 (файл bio)

Горизонтальная черта

Текст4

(файл groza)

Файл about.htm

Текст3

(гиперссылка на Текст4)

Содержание работы: 1. В своей папке создать папку лабораторной работы **HTML-3**. Скопировать в нее **все файлы** из папки, указанной преподавателем (HTML_ТЮТЧЕВ). 2. Открыть файл **article.txt**. Задать кодировку “Кириллица Windows-1251”. Сохранить

его в как файл (*Web-страницу*) **poet.htm** в своей папке. Название страницы (заголовок – поле title) изменить на Great Russian Poet F.Tutchev.

Внимание! Во время работы во избежание потери данных при сбоях формируемые

файлы время от времени сохранять. 3. Запустить браузер Internet Explorer, загрузить (открыть) сохраненный файл

poet.htm, просмотреть его. Вернуться в окно редактора.

Далее, по мере внесения изменений, регулярно переходить в браузер, проверяя его

оформление и работоспособность ссылок. При необходимости исправлений вернуться в соответствующее окно, внести необходимые исправления, снова проверить их в браузере. И т.д. 4. Слова “Поэтическое наследие Тютчева” сделать заглавием, выделив в отдельную строку, оформить их как заголовок 1-го уровня. 5. Задать цвет фона (*серый*) и цвета ссылок (*непросмотренные – красный, просмотренные – синий, наведенные – фиолетовый*). 6. Под заголовком создать таблицу из двух столбцов и одной строки. Задать рамку вокруг таблицы толщиной 5 пикселей (4,5 пт.), цвет темно-синий (*Word: Таблица – Свойства таблицы – Границы и заливки – Рамка – Ширина., Цвет*). Установить промежуток между столбцами 10 пикселей (0,25 см) (*Word: Таблица – Свойства таблицы – Параметры – Интервал между ячейками 1 пт.*). Цвет ячеек таблицы сделать желтым. 7. Поместить в левую часть рисунок **photo1.jpg** (портрет Тютчева), изменив его размеры до 300 пикселей в высоту, выровнять его по центру ячейки. Задать альтернативное текстовое имя (*alt=“Ф.И.Тютчев”*). 8. В правый столбец переместить остальной текст статьи и отформатировать его (фонт Arial, размер 12). 9. К фрагменту текста “А.Ф.Мерзляков” вставить thumbnail (миниатюру) портрета А.Ф.Мерзлякова шириной в 35 пикселей (*Вставка-Гиперссылка” или тег <A>*) и от него сделать гиперссылку на полноформатный портрет (файл **photo2.gif**), используя для этого инструмент “*Вставка-Гиперссылка” или тег <A>*”. 10. Ниже таблицы вставить горизонтальную линию и текст биографической справки о Тютчеве (файл: bio.txt) и отформатировать его (Заголовок – стиль 2-го уровня, подчеркнуть, текст шрифтом Times Roman, размером 14). 11. Ниже текста биографической справки вставить горизонтальную линию размером 5 см и текст стихотворения “Гроза...” (файл: groza.txt). 12. Текст стихотворения отформатировать (Заголовок – стиль 3-го уровня, каждая строфа — отдельный абзац, выровненный влево, шрифт – Courier, текст – размером 12, начертание – курсив, дату под стихотворением выровнять вправо). 13. Вставить закладку № 1 (тег <A> с атрибутом name) на Заголовок биографической справки о Тютчеве (“Вставка-Закладка” или “Правка-Закладка”). 14. Вставить закладку № 2 (тег <A> с атрибутом name) на начало стихотворения “Гроза...” в файле poet.htm. 15. Сделать гиперссылку из заглавия от слова “Тютчев”, на закладку № 1 (начало биографической справки), используя для этой цели инструмент гиперссылки, указав в качестве адреса имя закладки. 16. Сделать гиперссылку из текста биографической справки от слов “символический параллелизм и космические мотивы” на файл about.htm. 17. Из файла about.htm от слов первой строфы стихотворения (“Люблю грозу в начале мая, // Когда весенний, первый гром, // Как бы резвяся и играя // Грохочет в небе голубом”) сделать гиперссылку на текст данного стихотворения в файле poet.htm (закладка № 2). 18. В последний раз сохранить веб-страницу как файл poet.htm. Проверить оформление веб-страниц и работоспособность ссылок в браузере. Предъявить работу преподавателю.

Формы (HTML-4)

С помощью инструмента, заданного преподавателем (редактор Блокнот) создать форму (веб-страницу) для отправки личных данных.

Содержание работы: 1. В своей папке создать папку лабораторной работы **HTML-4**. В ней создать файл form.htm. 2. Присвоить форме имя **pers_data**. Озаглавить веб-страницу как **Rësumë** (элемент title). Задать кодировку “Кириллица Windows-1251”. Цвет фона страницы – светлозеленый.

Форма должна быть переслана методом post по адресу электронной почты informatika21@yandex.ru (атрибут action тега form). Форма состоит из нескольких

разделов, разделенных отцентрированной горизонтальной линией шириной 80% и толщиной 2 пикселя с затенением. Разделы включают названия полей и собственно поля ввода данных формы. Названия разделов – заголовки уровня 2, по центру, шрифт Arial, размер шрифта 14. Названия подразделов – шрифт Times Roman, размер шрифта 12, начертание курсив. 3. Заголовок **Resume** повторить на самой веб-странице по-русски (**Резюме**), оформив его как заголовок уровня 1, по центру, с подчеркиванием, размер шрифта 18. 4. Создать текстовое однострочное поле типа пароль (тег input, type=password) (длина поля 10 знаков, значение по умолчанию guest, максимальная длина вводимого пароля 7 символов), присвоив ему название “Введите пароль:” (слева от поля ввода, красным

цветом). Название поля и поле для ввода пароля выровнять по центру. После этого вставить разделитель – горизонтальную линию толщиной 5 пикселя красного цвета. 5. Первый раздел формы озаглавить Личные данные. В первом разделе поместить следующие три текстовых однострочных поля (с названиями):

Имя, Фамилия, Отчество (текстовые поля, ширина 30, максимальная ширина 40);

Пол (кнопки-переключатели (поля выбора));

Семейное положение (открывающееся меню) (значения подбирают студенты);

Дата рождения:

число (текстовое поле ввода),

месяц (поле типа меню, варианты выбора: январь-декабрь (значения: 1-12)),

год (поле типа меню, варианты выбора и значения: 1975-1985).

Несмотря на разные типы полей, начальный вид полей оформить единообразно в виде

трех окон: **число месяц год** 6. Во втором разделе с заголовком **Профессиональные данные** поместить поля со следующими названиями:

Образование (меню из 4 пунктов: среднее, среднее специальное, незаконченное высшее, высшее; по умолчанию – высшее);

Опыт работы (текстовое многострочное поле ввода (окно) высотой четыре строки и шириной 80 колонок);

Языки программирования (6 вариантов: Fortran, VBasic, C++, Java, Pascal, Ada) (поля флажки);

Иностранные языки (4 пункта) (поля флажки).

Оформить поля иностранных языков и языков программирования как таблицы, в ячейки которых поместить названия языков и флажки выбора.

Хобби, дополнительная информация (текстовое многострочное поле ввода (окно) высотой три строки и шириной 80 колонок).

7. Третий раздел озаглавить **Отправка данных**. В разделе поместить две кнопки:

Переслать и *Ввести заново*. Расположить их по центру страницы на одном уровне на

расстоянии пять пробелов. 8. Сохранить последний раз полученную страницу. Предъявить работу преподавателю.

Фреймы (HTML-5.) 1. Создать в папке лабораторной работы **HTML-5** файлы веб-страницы `frame_lab#10`,

состоящей из 3 фреймов. Вначале разбить экран на два вертикальных столбца (20 и 80 процентов соответственно; боковые поля 4 пикселя). Затем, используя вложенную структуру `FRAMESET`, разделить второй столбец на две горизонтальных части, так чтобы

первая (верхняя) была в два с половиной раза больше второй. Задать атрибут, определяющий наличие полос прокрутки во фреймах только при их необходимости, и

разрешить изменение размеров фреймов. Фреймы поименуем слева направо и сверху вниз как **А**, **Б** и **В**. 2. Создать файл веб-страницы с заголовком (крупными цветными буквами) “*Великие*

русские поэты” (№ 1). 3. Создать страницу – маркированный список со ссылками на файлы, посвященные нескольким русским поэтам (Пушкин, Лермонтов, Тютчев, Блок, Маяковский и т.д.) (№ 2). 4. Создать страницу с содержанием *Page is under construction* и соответствующим

рисунком (№ 3). 5. Скопировать в папку данной работы файлы мини-сайта “Тютчев” из предыдущей лабораторной работы. Если нет, создать хотя бы одну страницу, посвященную какому-либо поэту с двумя ссылками с нее

на другие файлы (файл `poet.htm` и другие – обобщенно

№ 4). 6. Первоначально во фрейм А поместить страницу № 2, во фрейм Б поместить

страницу № 1, фрейм В оставить пустым. 7. Веб-страницы, посвященные поэтам (№ 4) и загружаемые по ссылкам из фрейма А,

отображать во фрейме Б. В случае, если страница какого-либо поэта отсутствует, сделать ссылку на страницу № 3. 8. Файлы по ссылкам со страницы № 4 загружать во фрейм В (на странице о Тютчеве

это ссылки на портрет А.Ф.Мерзлякова и на файл `bio.htm`). 9. Сохранить последний раз полученную страницу. Предъявить работу преподавателю.

Краткая информация.

Фреймы определяются тэгом FRAMESET, который для Web-страниц, использующих фреймы, употребляется

Практическая занятие №12

Тема: Создание документа с использованием таблиц CSS.

Цели:

1. научиться работать с каскадными таблицами стилей;
2. изучить основные методы работы с CSS, селекторы, атрибуты.

Задание

Оформить подготовленный структурированный гипертекст, представленный в файле index.html в соответствии со стилем представленным на рисунке index.bmp

При выполнении работы допускается использовать «песочницу» <http://cssdesk.com/> и справочник <http://htmlbook.ru/css>

Для подробного освоения темы по «margin» и «padding» рекомендуется прочитать:

<http://habrahabr.ru/blogs/css/121810/>

<http://habrahabr.ru/blogs/css/123250/>

Исходный HTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Template</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css" media="all">
@import "style.css";
</style>
</head>
<body>
<div class="content">
<div class="toph"></div>
<div class="right">
<div class="title">GREY</div>
<div class="nav">
<ul>
<li><a href=#>HOME</a></li>
<li><a href=#>ARTICLES</a></li>
<li><a href=#>GALLERY</a></li>
<li><a href=#>AFFILIATES</a></li>
<li><a href=#>SUPPORT</a></li>
<li><a href=#>CONTACT</a></li>
</ul>
</div>
<h2>Top Articles:</h2>
<ul>
<li><a href=#>NoHeader Template</a></li>
<li><a href=#>Consectetuer adipiscing elit</a></li>
<li><a href=#>Lorem ipsum dolor sit amet</a></li>
<li><a href=#>dolor sit amet consectetur</a></li>
</ul>
<hr />
<h2>Links</h2>
<ul>
<li><a href=#>any.com</a></li>
<li><a href="htmlbook.ru/samcss">htmlbook.ru/samcss</a></li>
</ul>
```

```

<hr />
</div>
<div class="center">
  <h2><a href=#>Try sNews 1.4!</a></h2>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer euismod ante non diam. Sed eleifend odio sed quam. Sed vulputate, <a href=#>turpis at tincidunt</a> porttitor, est elit consequat metus, non dignissim augue mauris quis arcu. Phasellus faucibus blandit eros. Curabitur porttitor ante non est. Maecenas dolor. Aenean egestas sem. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed suscipit, nisi sit amet pharetra malesuada, sem velit laoreet sem, vitae iaculis diam neque consequat est. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Pellentesque tincidunt eros non quam. Mauris a magna sit amet libero accumsan auctor. Aenean nec urna non dui lobortis viverra...
  <p class="date">Posted by Avenir 
  <a href=#>Read more</a> 
  <a href=#>Comments (2)</a>  21.02.</p>
  <br />
  <h2><a href=#>Heading Item</a></h2>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer euismod ante non diam. Sed eleifend odio sed quam. Sed vulputate, <a href=#>turpis at tincidunt</a> porttitor, est elit consequat metus, non dignissim augue mauris quis arcu. Phasellus faucibus blandit eros.
  <p class="date">Posted by James 
  <a href=#>Read more</a> 
  <a href=#>Comments (7)</a>  18.01.</p>
  <br />
  <div class="boxad"> Your Ads here...sNews is a completely free PHP and MySQL driven tool for publishing and maintaining news articles on a website.</div>
</div>
<div class="footer"></div>
</div>
</body>
</html>

```

Требуемый стиль:

The screenshot shows a website layout with a dark sidebar on the left and a light main content area on the right. The sidebar contains two news articles, each with a title, a short text snippet, and a metadata line showing the author, a 'Read more' link, a 'Comments' count, and a date. Below the articles is an advertisement placeholder. The main content area features a 'GREY' logo, a vertical navigation menu with links for HOME, ARTICLES, GALLERY, AFFILIATES, SUPPORT, and CONTACT. Below the menu is a 'Top Articles' section with a list of article titles and a 'Links' section with two links: any.com and htmlbook.ru/samcss.

Ход работы

1. Для селектора «body»:

- установить цвет фона тела страницы #7D8085 используя свойство «background» селектора «body»;
- установить шрифт тела страницы 74% Arial, Sans-Serif используя свойство «font»;
- 2. Для селектора (класса) «.toph»:
 - установить неповторяющийся фоновый рисунок «top.jpg», без полей, высотой 39px, с выравниванием по центру используя свойства «background», «height», «margin», «padding»;
- 3. Для селектора (класса) «.content»:
 - установить повторяющийся фоновый рисунок «bg.jpg», без полей, шириной 800px, с выравниванием по центру используя свойства «background», «width», «margin», «padding»;
- 4. Для селектора (класса) «.title»:
 - установить неповторяющийся фоновый рисунок «logo.jpg» с выравниванием по левой стороне используя свойство «background»;
 - с полями от верхнего края 10px, от левого 40px, высотой 28px используя свойства «height», «padding»;
 - установить размер шрифта 140%, полужирный, цвет #F29900 используя свойство «font»;
- 5. Для селектора (класса) «.right»:
 - установить обтекание слева используя свойство «float»;
 - с отступом от правого края 15px, и полем от правого края 1em используя свойства «margin», «padding»;
 - установить размер шрифта 95%, полужирный, используя свойство «font»;
 - установить ширину слоя 170px, используя свойство «width»;
- 6. Для селектора (класса) «.footer»:
 - установить запрет на обтекание одновременно с правого и левого края используя свойство «clear»;
 - установить неповторяющийся фоновый рисунок «bot.jpg» с выравниванием по центру стороне используя свойство «background»;
 - установить выравнивание текста по центру используя свойство «text-align»;
 - установить высоту слоя в 37px, используя свойство «height»;
 - установить автоматическую ширину слоя «auto», используя свойство «width»;
- 7. Для селектора (класса) «.center»:
 - установить обтекание справа, используя свойство «float»;
 - установить ширину слоя 530px, используя свойство «width»;
 - установить размер шрифта 95%, полужирный, используя свойство «font»;
 - установить цвет текста #FFF;
 - установить поля и отступы соответственно «margin: 0px 0 5px 35px; padding: 0;»;
- 8. Установить цвет ссылок для центрального блока:
 - цвет основной ссылки #F29900 используя селекторы «.center a»;
 - цвет ссылки под курсором #FFF используя селекторы «.center a:hover»;
- 9. Установить для блока «date»:
 - цвет основного текста #ccc
 - выравнивание текста по правому краю, используя свойство «text»
 - поля и отступы соответственно «margin: 4px 0 5px 0; padding: 0.4em 0 0 0;»
 - верхнюю границу блока толщиной в 1px цветом #555, используя свойство «border»;
- 10. Установить цвет ссылки в #ccc, используя свойство «color» селектора «.date a»;
- 11. Установить цвет ссылки в #7D8085, используя свойство «color» селектора «.right a»;
- 12. Установить цвет тегов параграфа и ссылок в #888, используя свойство «color» селекторов «p» и «a»;
- 13. Для селектора «a»:
 - установить наследование фона, используя свойство «background» с параметром «inherit»;
 - выключить стили текста, используя свойство «text-decoration»;
- 14. Для селектора «p» установить отступы и поля соответственно «margin: 0 0 5px 0; padding: 0;»
- 15. Для селектора «hr»:
 - установить высоту 1px;
 - установить основной и фоновый цвет в #eee;
 - убрать «border»;
- 16. Для селектора заголовка «h1»:
 - убрать отступы и поля;
 - установить цвет в #FFF;
 - установить жирный шрифт размера 1.8em, гарнитур Arial, Sans-Serif;
 - установить наследование фона, используя свойство «background» с параметром «inherit»;
 - установить свойство letter-spacing равным «-1px»;
- 17. Установить цвет для ссылок «a» находящихся в заголовке «h1» в #FFF, и установить для них наследование фона.

18. Для селектора заголовка «h2»:

- установить наследование цвета фона, используя свойство «background» с параметром «inherit»;
- установить отступы и поля согласно «margin: 10px 0 10px 0; padding:0;»
- установить основной цвет в #F29900;
- установить размер шрифта 140%, жирный;

19. Установить цвет для ссылок «а» находящихся в заголовке «h2» в #F29955. Для ссылок под курсором установить тот же цвет и убрать подчёркивание.

20. Применить стили элементов списка согласно инструкциям и вставить комментарии действия свойств:

```
ul { margin: 5px 0 20px 15px;
padding: 0;
list-style: none;
}
li { list-style-type: square;
color: #F29900;
margin: 0 0 0px 0;
padding: 0 0 0 0px;
}
li a { color: #7D8085; }
li a:hover { color: #F29900; }
```

21. Выполнить применение указанных стилей тремя методами: связанным, глобальным и внутренним

22. **Дополнительно:**

- рассмотреть работу базовых макетов построения сайта представленных в <http://htmlbook.ru/layout>
- ознакомиться с материалами статьи <http://habrahabr.ru/blogs/css/126207/>

«Экспорт стилей и валидация»

Цели:

1. *научиться экспортировать стили;*
2. *научиться структурировать документ.*

Ход работы

Используя материал предыдущей работы выполнить следующее задание:

1. Сменить заголовок с «GRAY» на «CSS», подобрать логотип.
2. Перенести из учебника материал, по числу пунктов меню начиная с введения.
3. Выполнить хотя бы одно абсолютное и относительное позиционирование (<http://ru.html.net/tutorials/css/lesson14.php>).
4. Сделать хотя бы один плавающий блок – картинку (см. <http://ru.html.net/tutorials/css/lesson13.php>)
5. Сделать наложение (<http://ru.html.net/tutorials/css/lesson15.php>)
6. Указать у всех используемых шрифтов родовые имена и объяснить зачем это нужно. (<http://ru.html.net/tutorials/css/lesson4.php>)
7. Выполнить экспорт стилей в файл и осуществить его подключение
8. Выполнить валидацию файла стиля (см. <http://ru.html.net/tutorials/css/lesson16.php>)

Тема: Создание документа с использованием языка Java.

Цели:

1. ознакомиться с синтаксисом языка;
2. иметь представление о методах ввода/вывода информации;
3. знать способы объявления переменных;
4. уметь применять полученные знания на практике.

Способы объявления переменных.

Объявление переменной делается оператором **var**:

```
var a;
```

Переменные бывают трех основных типов:

- **number** - любое число (12; -123; 12.3; 1.23e2)
- **boolean** - принимает всего два значения: правда и ложь ("2+2=4" - true; "2*5=-9" - false)
- **string** - текстовая строка ("Любая фраза")

JavaScript сам определит тип переменной при присвоении ей значения. А присвоить значение можно любым из следующих способов:

```
<script>
var a=6363;
var s;
s="Это текстовая строка";
YourName=prompt ("Ваше имя", "Вводить здесь");
</script>
```

Т.е. для объявления переменной не всегда необходимо пользоваться оператором var.

Названием переменной может быть не любая комбинация. Имя переменной должно состоять из латинских букв, цифр, лучше не использовать символы (кроме, символа нижней черты _), нельзя использовать пробелы, имя не должно начинаться с цифры и не должно совпадать с зарезервированным словом JScript.

Упражнение 1. Написать программу, которая спрашивала бы у пользователя его имя (например, Маша), а потом говорила «Привет, Маша».

Специальные символы

При выводе информации иногда бывает необходимо вывести и специальные символы. Вот некоторые из них:

\b - возврат на один символ

\f - переход на следующую страницу

\n - переход на новую строку

\r - возврат каретки

\t - табуляция

' - одинарные кавычки (апостроф)

'' - двойные кавычки

**** - обратная косая черта

\000 - вставка любого символа по восьмеричному коду (например: \123)

\x00 - то же самое только по шестнадцатеричному коду ASCII (например: \xAC)

Упражнение 2. Известна такая довольно старая шутка. Вы спрашиваете у вашего друга:

"Где ты больше всего любишь обедать?" (Например, он отвечает "В столовой")

"Твоя любимая еда?" ("Торт")

"Твой любимый музыкальный инструмент?" ("Барабан")

"Твоя любимая геометрическая фигура?" ("Круг")

"Твоя любимая скорость?" ("140 км/ч")

После этого вы рассказываете другу такую историю:

"Вы сидите в столовой. Играет романтическая музыка, горят свечи... Вы кладете в рот кусочек вкуснейшего торта, и вдруг на вас падает барабан, и со скоростью 140 км/ч вы превращаетесь в круг!.."

Осуществите эту шутку в JavaScript. Используйте методы `prompt()`, `alert()` и переменные.

Практическое занятие №14

Тема: Создание документа с использованием языка Java.

Цели:

1. ознакомиться с математическими операторами;
2. иметь представление об объектах языка (объект *Math*);
3. уметь применять полученные знания на практике.

Математические операторы

+ сложение

- вычитание

* умножение

/ деление

% остаток от деления

Реализуйте следующий пример:

```
<script>
alert (45+56);
alert (45-56);
alert (45*56);
alert (45/56);
alert (56%45);
</script>
```

Объекты

Объекты позволяют получить дополнительный набор операторов. Начнем с объекта **Math**. Этот объект предоставляет дополнительный набор математических операторов. Вот наиболее распространенные:

- **LN2; LN10; LOG2E; LOG10E** - различные логарифмы
- **E; PI** - число e и число π соответственно
- **SQRT2; SQRT1_2** - корень квадратный из двух и 0,5 соответственно

Реализуйте следующий пример:

```
<script>
alert (Math.Ln10);
a=Math.E;
alert (a);
</script>
```

- **sin(); cos(); tan(); asin(); acos(); atan()** - синус, косинус, тангенс, арксинус, арккосинус, арктангенс
- **abs()** - модуль числа
- **sqrt()** - квадратный корень числа
- **pow()** - степень числа (в скобках через запятую основание и показатель степени)
- **exp()** - экспонента числа
- **log()** - логарифм числа
- **random()** - псевдослучайное число
- **round()** - округление числа
- **ceil()** - округление по избытку
- **floor()** - округление по недостатку (целая часть от деления двух целых чисел)
- **max()** - максимальное значение из тех чисел, которые через запятую перечислены в скобках
- **min()** - минимальное значение из тех чисел, которые через запятую перечислены в скобках

Упражнение 1. Написать программу, вычисляющую площадь треугольника по формуле Герона.

Упражнение 2. Написать программу, вычисляющую объем усеченного конуса.

Упражнение 3. `Math.random()` дает псевдослучайное число в диапазоне от 0 до 1. Подумайте, как составить генератор псевдослучайных чисел в диапазоне от 1 до 100.

Практическое занятие №15

Тема: Создание документа с использованием языка Java.

Цели:

1. ознакомиться с синтаксисом условного оператора;
2. получить навыки работы с операторами *if* и *else*.

Существует такой тип переменной как **Boolean**. Операторы условия работают через него. Т.е. если некоторое выражение или переменная принимает значение **true**, то идет процесс выполнения определенного набора операторов. А если же **false**, то выполняется другой набор операторов (или не выполняется ничего).

Для получения булеановских выражений используются специальные операторы сравнения:

`==` равно (`5==5`)

`<` меньше (`-1<1`)

`>` больше (`1>-2`)

`<>` не равно (`1<>2`)

`>=` больше или равно (`10>5 5=5`)

`<=` меньше или равно (`5<10 10=10`)

Логические связки:

`!=` логическое НЕ (`1!=2`)

`&&` логическое И (`2>1 && 3>1`)

`||` логическое ИЛИ (`2>1 || 0>1`)

Если требуется проверить несколько условий, то каждое, помимо общих скобок, берется в свои, отдельные.

Задание 1. Разработать программу, которая говорила бы пользователю «Спасибо!» только в том случае, если он введет ноль, иначе она будет говорить «Упитанный, а не воспитанный».

Используйте оператор условия и операторы сравнения.

Задание 2. Разработать программу, которая бы проверяла, чтобы пользователь ввел число четное и не больше 10. Используйте оператор условия, операторы сравнения, логические связки.

Цели:

1. ознакомиться с основными подходами в разработке программ;
2. получить навыки разработки калькулятора.

Основы подхода к программированию на JavaScript.

Идея JavaScript очень проста. Все операции, которые можно исполнять в программе на JavaScript, описывают действия над хорошо известными и понятными объектами, которыми являются элементы рабочей области браузера и контейнеры языка HTML. Собственно объектная ориентированность JavaScript на этом и кончается. Никаких классов объектов, а тем более, наследования в JavaScript нет. Есть только объекты с набором свойств и набор функций над объектами, которые называются методами. Кроме методов существуют и другие функции, больше похожие на функции из традиционных языков программирования, которые позволяют работать со стандартными математическими типами или управлять процессом выполнения программы. Еще в JavaScript есть события - аналог программных прерываний. Эти события также ориентированы на работу в World Wide Web, например загрузка страницы в рабочую область Navigator или выбор гипертекстовой ссылки. Используя события, автор гипертекстовой страницы и программы, ее отображающей, может организовать просмотр динамических объектов, например бегущей строки, или управление многооконным интерфейсом.

Для встраивания скриптов в тело HTML-документа используется контейнер SCRIPT. Не все браузеры способны распознавать и исполнять скрипты, поэтому само тело скрипта помещается в контейнер комментария.

Пример

| | |
|---|--|
| <pre><html> <head> <!-- <script language="JavaScript"> Код сценария </script> --> </head> <body> Код HTML </body> </html></pre> | <p>Пример встраивания скрипта в тело HTML-документа.</p> |
|---|--|

В этом примере в заголовок документа (контейнер HEAD) включен контейнер SCRIPT. Далее, в тексте страницы определен комментарий, в который включен текст скрипта.

Необходимо запомнить следующие вещи:

1. Поместив сценарий на JavaScript в разделе <head> документа, вы делаете так, что весь сценарий будет загружен до того, как потребуется его выполнить.
2. Код сценария должен быть заключен в теги комментария HTML для того, чтобы старые браузеры, не понимающие JavaScript, не отображали его на экране.
3. Регистр, которым написаны буквы, в JavaScript имеют значение.

Отступление. Для того чтобы можно было обратиться к какому-то объекту, при его описании используется атрибут ID или name, **например**, <P ID=passage>, теперь к параграфу можно обратиться, используя passage.

Новые понятия

Прежде чем обратиться к объектам, составляющим объектную модель браузера, следует рассмотреть ряд новых понятий, которые мы будем использовать на протяжении ряда следующих занятий.

Объект — совокупность свойств, методов, событий и коллекций, предоставляемая браузером в рамках объектной модели. Все объекты создаются самим браузером, и доступ к ним осуществляется через экземпляр:

Объект.метод / событие / коллекция / свойство.

Свойство — переменная в рамках объекта, которая может использоваться для получения каких-то значений или установки новых. Ряд свойств может быть доступен только для чтения.

Метод — процедура или функция, предоставляемая объектом для выполнения каких-либо действий или управления свойствами объекта.

Событие — какое-либо действие пользователя или момент работы браузера. Для реакции на события создаются *обработчики событий*.

Коллекция — упорядоченный набор свойств, похожий на массив, доступ к которому осуществляется специальными средствами.

Запуск JavaScript. Что необходимо сделать, чтобы запускать скрипты, написанные на языке JavaScript? Вам понадобится браузер, способный работать с JavaScript - например, Netscape Navigator (начиная с версии 2.0) или Microsoft Internet Explorer (MSIE - начиная с версии 3.0).

События. События и обработчики событий являются очень важной частью для программирования на языке JavaScript. События, главным образом, инициируются теми или иными действиями пользователя. Если он щелкает по некоторой кнопке, происходит событие *onClick*. Если указатель мыши пересекает какую-либо ссылку гипертекста - происходит событие *MouseOver*. Существует несколько различных типов событий. Мы можем заставить нашу JavaScript-программу реагировать на некоторые из них. И это может быть выполнено с помощью специальных программ обработки событий. Так, в результате щелчка по кнопке может создаваться выпадающее окно. Это означает, что создание окна должно быть реакцией на событие щелчка - *Click*. Программа - обработчик событий, которую мы должны использовать в данном случае, называется *onClick*. И она сообщает компьютеру, что нужно делать, если произойдет данное событие. Приведенный ниже код представляет простой **пример** программы обработки события *onClick*:

| | |
|--|---|
| <pre><form> <input type="button" value="Щелкни" onClick = "alert('Yo')"> </form></pre> | Пример обработки события: при нажатии на кнопку выводится сообщение «Yo». |
|--|---|

Данный пример имеет несколько новых особенностей: рассмотрим их по порядку. Вы можете здесь видеть, что мы создаем некую форму с кнопкой. Первая новая особенность - `onClick="alert('Yo')"` в тэге `<input>`. Как мы уже говорили, этот атрибут определяет, что происходит, когда нажимают на кнопку. Таким образом, если имеет место событие *onClick*, компьютер должен выполнить вызов `alert('Yo')`. Обратите внимание, что в этом случае мы даже не пользуемся тэгом `<script>`. Функция `alert()` позволяет Вам создавать выпадающие окна. При ее вызове Вы должны в скобках задать некую строку. В нашем случае это 'Yo'. И это как раз будет тот текст, что появится в выпадающем окне. Таким образом, когда Вы щелкаете на кнопке, наш скрипт создает окно, содержащее текст 'Yo'. В этом примере мы написали `onClick="alert('Yo')"` - то есть мы использовали и двойные, и одинарные кавычки. Если бы мы написали `onClick="alert("Yo")"`, то компьютер не смог бы разобраться в нашем скрипте, поскольку становится неясно, к которой из частей конструкции имеет отношение функция обработки событий `onClick`, а к которой - нет. Поэтому Вы и вынуждены в данном случае перемежать оба типа кавычек. Не имеет значения, в каком порядке Вы использовали кавычки - сперва двойные, а затем одинарные или наоборот. То есть Вы можете точно так же написать и `onClick='alert("Yo")'`.

Вы можете использовать в скрипте множество различных типов функций обработки событий. Сведения о некоторых из них мы получим в данном описании, однако не о всех.

Функции. В большинстве наших программ на языке JavaScript мы будем пользоваться функциями. В большинстве случаев функции представляют собой лишь способ связать вместе нескольких команд. Давайте, к **примеру**, напишем скрипт, печатающий некий текст:

```
<html>
<script language="JavaScript">
function myFunction()
{
  document.write("Добро пожаловать на мою страницу!<br>");
  document.write("Это JavaScript!<br>");
}
</script>
<body>
<form>
<input type="button" value="Вызов функции" onClick="myFunction()">
</form>
</body>
</html>
```

В этом скрипте мы определили некую функцию, состоящую из следующих строк:

```
function myFunction() {
  document.write("Добро пожаловать на мою страницу!<br>");
```

```
document.write("Это JavaScript!<br>");
```

```
}
```

Все команды скрипта, что находятся внутри фигурных скобок - {} - принадлежат функции *myFunction()*. Это означает, что обе команды `document.write()` теперь связаны воедино и могут быть выполнены при вызове указанной функции. Можно увидеть, что здесь при нажатии на кнопку осуществляется вызов функции *myFunction()*.

Замечание:

1. Обратите внимание на строку `onClick="myFunction()"` при описании кнопки. Событие `onClick` возникает, когда пользователь щелкает левой кнопкой по объекту, и мы обрабатываем данное событие, вызывая функцию *myFunction*, описанную выше.
2. Обратите внимание, что даже если мы описываем функцию без параметров, например, *myFunction()*, как при описании, так и при вызове после имени функции мы должны указывать круглые скобки.

Практическая часть, написание калькулятора.

1. Используя программу Блокнот создать следующую форму. Замечания: используя атрибут `ID`, присвоить форме идентификатор `calc`, имена кнопок можно задать произвольно, чтобы все кнопки были одинакового размера (25x25), можно описать соответствующий класс CSS.

```
0
```

| | | | |
|---|-----|---|---|
| 7 | 8 | 9 | / |
| 4 | 5 | 6 | * |
| 1 | 2 | 3 | - |
| 0 | +/- | . | + |

2. Описать функцию с именем `zero()`, в теле которой написать следующий код: `идентификатор_формы.имя_текстовой_строки.value+= '0'`, это означает, что мы хотим добавить в строку цифру 0; ту же запись можно было записать так: `идентификатор_формы.имя_текстовой_строки.value= идентификатор_формы.имя_текстовой_строки.value+'0'`. Можно пользоваться любой из записей, первая из них – это особенность языка Си.

3. Для кнопки с цифрой нуль обработать событие `onClick`, вызвав описанную выше функцию следующим образом: `<input type="button" value="0" name="B10" onClick=zero()>`.

4. Опробовать в действии Вашу страничку.

5. Аналогичным образом описать все цифры, описывая соответствующие функции и обрабатывая событие `onClick` для соответствующей кнопки, можно пользоваться любой из записей, но первая предпочтительней.

6. Немножко уменьшим объем нашей странички. Все 10 описанных выше функций выполняют одну и ту же последовательность действий, а именно добавляет в строку один символ, значит все их можно объединить в одну функцию, в которую в качестве аргумента будем передавать символ. Для этого опишите новую функцию с именем `add(d)`, где `d` – это аргумент, в котором мы будем передавать символ. В теле функции написать оператор, который в строку добавляет значение переменной `d` (смотри выше).

7. Изменить обработчики событий для кнопок с цифрами следующим образом, например для кнопки с цифрой 1 нужно написать `onClick=add('1')`. Для кнопки с запятой обработать событие `onClick`, вызвав функцию `add`, передавая в качестве параметра `','`.

8. Опишем функцию с именем `znak(c)`, которая будет запоминать знак операции, первое число и очищать строку. Для этого выше всех функций описать три переменные с именами `oper` (знак операции), `perвое` (первое число), `второе` (второе число) следующим образом: `var oper, perвое, второе`, то есть мы описали переменные, значения которых будет известно во всех функциях. Теперь опишем функцию `znak(c)`, где `c` – параметр для хранения знака операции. Сначала в переменную `oper` помещаем значение переменной `c`, затем, используя функцию `eval` для преобразования строки в число, в переменную `perвое` помещаем значение текстовой строки и очищаем текстовую строку, помещая в нее значение, равное нулю. Обработать событие щелчок по кнопкам со знаками операций, вызвав описанную выше функцию.

9. Опишем функцию chet(), которая будет подсчитывать результат. Сначала переменной второе присвоим значение, хранящееся в текстовой строке (смотри предыдущий пример), затем используя условный оператор можем вычислить результат, например, если знак «+», то условный оператор выглядит следующим образом: if (op == '+') идентификатор_формы.имя_текстовой_строки.value=first+second;. Аналогично написать для всех знаков.
10. Сохранить документ и опробовать в действии.
11. Самостоятельно: добавить кнопку, которая бы очищала текстовую строку, кнопку, которая бы меняла знак числа.

Практическое занятие №17

Тема: Создание документа с использованием языка PHP.

Цели:

1. приобретение навыков обработки данных текстовых полей;
2. ознакомление со структурой стандартной формы.

Порядок выполнения работы:

1. Создайте каталог php2 и в нем файл forma.html. Создайте форму по следующему образцу:

Обработка формы. Результат

Введите Ваше имя:

Введите пароль:

Введите Ваше e-mail

Добавьте Ваш комментарий:

Создайте файл обработчик формы, который должен выполнять следующие действия:

- Проверку, все ли поля заполнены:

```
If ($_GET['nm'] == '' or ($_GET['pass'] == '' or ($_GET['email'] == '' ))
{echo "<font color='red'>Вы ввели не все данные</font>";
Exit; }
```

- Выводить приветствие с указанием имени;

- Формировать и отправлять письмо:

```
$komu='xxx.ru';
$tema="Вопрос от ".$_GET['nm']." ".$_GET['email'];
$text_p=$_GET['quest'];
mail($komu, $tema, $text_p);
echo "Ваш вопрос отправлен администратору";
```

Вопросы для самоконтроля

1. Какие типы переменных поддерживает язык PHP?
2. В чем отличие php-страницы и html-страницы?

«Создание счетчика посещений»

Цели:

1. приобретение навыков использования функций обработки файлов;
2. ознакомление со структурой стандартной формы.

fopen(имя файла, режим работы) – возвращает число – дескриптор открытого файла, по которому можно обращаться к открытому файлу.

Режим работы имеет 2 составляющие:

- способ работы с информацией (текстовый (t) и бинарный(b));
- способ работы с файлом:

| Способ | Описание |
|--------|---|
| R | Файл открывается для чтения, указатель текущей позиции в начале файла. Если файла не существует, возникает ошибка |
| R+ | Файл открывается для чтения и записи, указатель текущей позиции в начале файла. Если файла не существует, возникает ошибка |
| W | Создается пустой файл и открывается для записи, указатель текущей позиции в начале файла. Если файл существует, он перезаписывается |
| W+ | Создается пустой файл и открывается для чтения и записи, указатель текущей позиции в начале файла. Если файл <input type="checkbox"/> , он перезаписывается |
| A | Файл открывается для записи, указатель текущей позиции в конце файла. Если файла не существует, он создается |
| A+ | Файл открывается для записи и чтения, указатель текущей позиции в конце файла. Если файла не существует, он создается |

flock(дескриптор файла, режим блокировки) – блокирует файл для использования других пользователей.

Режимы блокировки:

- 1- устанавливает блокировку;
- 2- устанавливает блокировку;
- 3- снятие блокировки;

fgets(дескриптор файла) – считывает данные из файла.

ftruncate(дескриптор файла, размер) – обрезает данные из файла до заданного размера (указывается в байтах). Возвращает TRUE (при успешном выполнении) или FALSE.

fputs(дескриптор файла, данные) – осуществляет запись данных в файл.

fclose(дескриптор файла) – закрытие файла.

die(текст сообщения об ошибке) – выводит текст, переданный в качестве параметра, и осуществляет выход из программы.

Одновременное использование двух функций:

fopen() or die() – если результат выполнения первой функции FALSE, то в этом случае выполняется вторая функция.

Порядок выполнения работы

Количество посещений любой страницы хранится в текстовом файле с именем counter.txt.

1. Ввести код программы-счетчика посещений counter.php

```
<?php
```

```
$f=fopen("counter.txt", "a+t") or die("Невозможно открыть файл");
```

```
flock( $f, 2);
```

```
$s = fgets($f);  
$s+=1; // $s=$s+1;  
ftruncate ($f, 0);  
fputs ($f, $s);  
flock ($f, 3);  
fclose($f);  
echo $s;  
?>
```

2. Открыть код страницы forma.html первой практической работы.
3. Добавить код для подключения счетчика в нижней части левой панели:

```
<?php  
echo "Количество посещений – "; require_once("counter.php");  
?>
```

Require_once(имя файла) – подключает модуль, имя которого указано в параметре. В качестве модуля используют программы PHP или HTML.

4. Заменить расширение файла: forma.php

Практическое занятие №18

Тема: Создание документа с использованием языка PHP.

Цели:

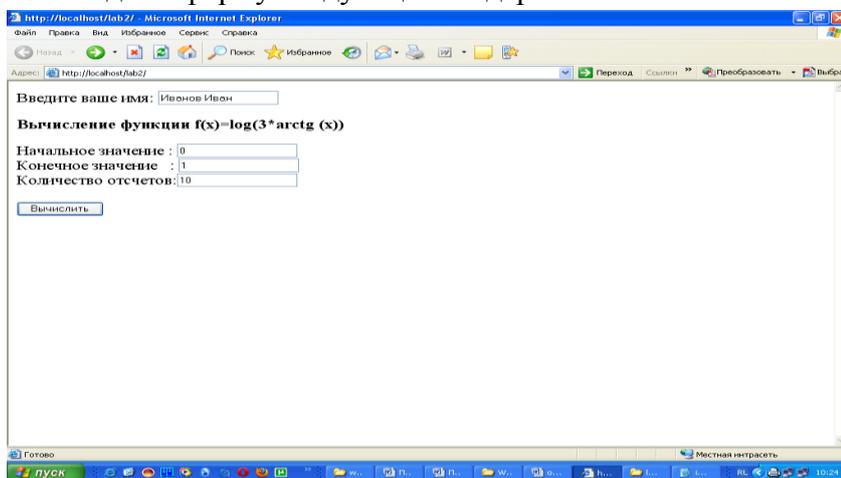
1. закрепление навыков отладки PHP-кода;
2. ознакомление со структурой стандартной формы.

Описание и вызов функций:

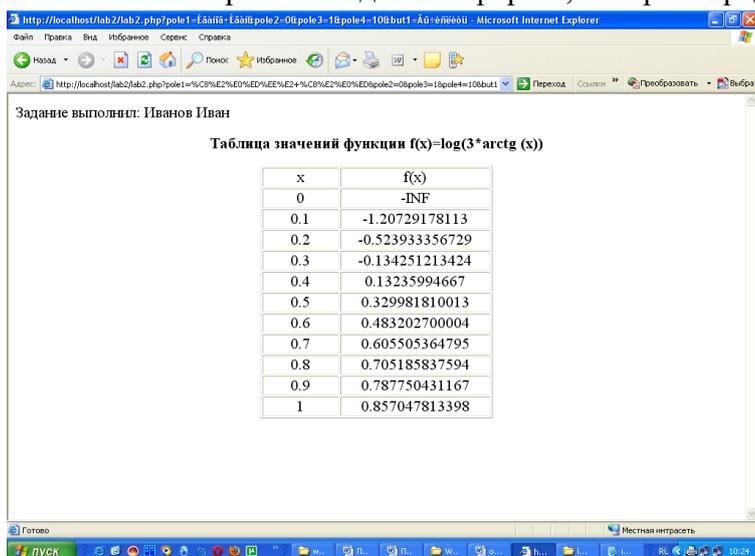
```
<?php
function first_function() {
    echo "<h4>Первая пользовательская функция</h4>";
    function second_function() {
        echo "<h5>Вторая пользовательская функция</h5>";
    }
    first_function();
    second_function();
?>
```

3. Порядок выполнения работы:

1. Создать форму следующего содержания:



2. Написать обработчик данных формы, который представляет результаты в виде следующей таблицы:



Практическое занятие №19

Тема: Создание документа с использованием языка PHP.

Цели:

1. закрепление навыков обработки данных формы;
2. закрепление навыков использования массива в программе.

В PHP существуют различные методы инициализации массивов:

1. простое присвоение значений

```
<?
$car[] = "passenger car";
$car[] = "land-rover";
echo($car[1]); // выводит "land-rover"
?>
```

2. явное указание индекса массива:

```
<?
$car[0] = "passenger car";
$car[1] = "land-rover";
echo($car[1]); // выводит "land-rover"
?>
```

3. использование конструкции array():

```
<?
$car = array("passenger car", "land-rover");
echo($car[1]); // выводит "land-rover"
?>
```

4. явное указание индексов (в этом случае применяется оператор ==>)

```
<?
$car = array("passenger car", 5 => "land-rover", "station-wagon", "victoria");
echo($car[0]); echo("<br>"); // выводит "passenger car"
echo($car[5]); echo("<br>"); // выводит "land-rover"
echo($car[6]); echo("<br>"); // выводит "station-wagon"
echo($car[7]); // выводит "victoria"
?>
```

5. индексами массива могут быть и строки:

```
<?
$car = array("pc" => "passenger car", "lr" => "land-rover");
echo($car["lr"]); echo("<br>"); // выводит "land-rover"
echo($car["pc"]); // выводит "passenger car"
?>
```

Для обработки элементов массива используют:

1. цикл FOREACH

```
foreach (array as [$key =>] $value)
{
    statements;
}
```

Пример:

```
<?
$car = array("passenger car", "land-rover", "station-wagon", "victoria");
foreach($car as $index => $val)
{
    echo("$index -> $val <br>");
}
?>
```

Как видно из синтаксиса, переменная \$key необязательна и может быть опущена:

```
<?
echo(
    "available cars: <br> <ul>"
);
```

```

$car = array("passenger car", "land-rover", "station-wagon", "victoria");
foreach($car as $val)
{
    echo("<li>$val</li>\n");
}
echo("</ul>");
?>

```

Порядок выполнения работы

1. Создать новую страницу с формой следующего вида:

<h2> Форма для регистрации студентов</h2>

<form action="1.php" method=POST>

Имя
 <input type=text name="first_name" value="Введите ваше имя">

Фамилия
 <input type=text name="last_name" >

E-mail
 <input type=text name="email" >

<p> Выберите курс, который будете посещать:

<input type=checkbox name='kurs[]' value='PHP'> PHP

<input type=checkbox name='kurs[]' value='LISP'> LISP

<input type=checkbox name='kurs[]' value='C++'> C++

<input type=checkbox name='kurs[]' value='UNIX'> UNIX

<p> Что Вы хотите, чтобы мы знали о Вас?

<textarea name="comment" cols=32 rows=5></textarea>

<input type=submit value="Отправить">

<input type=reset value="Отменить">

</form>

В файле action.php, обрабатывающем эту форму, можно написать следующее:

```

<?php
$str = "Здравствуйте, ".$_POST ["first_name"]." ".$_POST ["last_name"]."!  
>";
$str.= "Вы выбрали для изучения курс по ". $_POST["kurs"];
echo $str;
?>

```

2. Написать обработчик формы 1.php для регистрации участников заочной школы программирования и после регистрации отправить участнику сообщение. По полученным сведениям от зарегистрировавшегося человека, скрипт генерирует соответствующее сообщение. Если человек выбрал какие-то курсы, то ему выводится сообщение о времени их проведения и о лекторах, которые их читают. Если человек ничего не выбрал, то выводится сообщение о следующем собрании заочной школы программистов.

```

<?
// создадим массивы соответствий «курс-время» и «курс-лектор»
$time = array("PHP"=>"14.30", "LISP"=>"12.00", "C++"=>"15.00", "UNIX"=>"14.00");
$lector= array("PHP"=>"Васильев", "LISP"=>"Иванов", "C++"=>"Петров", "UNIX"=>"Сидоров");
define("SIGN", "С уважением, администрация");// определяем подпись как константу
define("MEETING_TIME", "18.00"); // задаем время собрания
$date="12 мая";
$str="Здравствуйте, уважаемый ".$_POST["first_name"]. " ". $_POST["last_name"]."!<br>";
$str .= "<br> Сообщаем Вам, что ";
$select ="";
$skurses=$_POST["kurs"];
if (!isset($skurses)) {
    $sevent= "следующее собрание студентов";
    $str .= "$sevent состоится $date ". MEETING_TIME . "<br>";
} else {
    $sevent= "выбранная Вами лекция состоится $date <ul>";
    for ($i=0; $i < count($skurses); $i++) {
        $k=$skurses[$i];
        $select = $select . "<li> лекция по $k в $time[$k]";
        $select .= " ваш лектор, $lector[$k]"
    }
    $sevent = $sevent . $select . "</ul>";
    $str .= "$sevent";
}
$str .= "<br>" . SIGN;
echo $str
?>

```

3. Реализуйте тест: ЗНАЕТЕ ЛИ ВЫ СЕБЯ.

Ученые установили, что если из двух полушарий ведущим является правое полушарие мозга, то у человека преобладает эмоциональная сфера. Если ведущим оказывается левое полушарие, то у человека аналитический склад ума преобладает над эмоциональностью. Предлагаемый тест как раз и позволяет выявить ведущее полушарие. Это является признаком врожденным и, как правило, не меняется до конца жизни.

Вопрос N 1. Переплетите пальцы рук и вы заметите, что сверху всегда оказывается один и тот же палец, если левый - вы человек эмоциональный, правый - у вас преобладает аналитический склад ума.

Вопрос N 2. Попробуйте "прицелиться", выбрав себе мишень и глядя на нее через своеобразную мушку - карандаш или ручку. Правый ведущий глаз говорит о твердом, настойчивом, более агрессивном характере, левый - о мягком и уступчивом.

Вопрос N 3. Если при переплетении рук на груди наверху оказывается левая рука, то вы способны к кокетству, правая - склонны к простоте и простодушию.

Вопрос N 4. Если удобнее хлопать правой рукой, можно говорить о решительном характере, левой - вы часто колеблетесь, прежде чем принять решение.

ПППП - для вас характерен консерватизм, ориентация на общепринятое мнение (на стереотип). Вы не любите конфликтовать, спорить и ссориться.

ПППЛ - определяющая черта вашего характера - нерешительность.

ППЛП - для вас характерны кокетство, решительность, чувство юмора, артистизм. При общении с вами необходимы юмор и решительность. Очень контактный тип характера. Этот тип у женщин встречается наиболее часто.

ПЛЛЛ - редкий тип характера. Мягкий. Наблюдается некоторое противоречие между нерешительностью (левое аплодирование) и твердостью характера (правый ведущий глаз).

ПЛПП - тип характера, сочетающий аналитический склад ума и мягкость. Чаше встречается у женщин - тип "деловой женщины". Медленное привыкание, осторожность, в отношениях терпимость и некоторая холодность.

ПЛПЛ - слабый и самый редкий тип характера. Обладатели такого характера беззащитны, подвержены различному влиянию. Встречается, как правило, у женщин.

ЛППП - такое сочетание встречается очень часто. Основная черта - эмоциональность, сочетающаяся с недостаточной настойчивостью.

ЛППЛ - для вас характерны мягкость, наивность. Требуете особого, внимательного отношения к себе - тип "маленькой королевы".

ЛЛПП - вам присущи дружелюбие и простота, некоторая разбросанность интересов, склонность к самоанализу.

ЛЛПЛ - в вашем характере преобладают простодушие, мягкость, доверчивость. Очень редкий тип характера, практически не встречается у мужчин.

ЛЛПП - вы эмоциональный, энергичный и решительный человек, но часто наспех принимаете решения, которые приносят серьезные осложнения. Очень важен дополнительный тормозной механизм. Мужчины с таким характером менее эмоциональны.

ЛЛЛЛ - у вас антиконсервативный тип характера. Способны на старые вещи взглянуть по-новому. Характерны эмоциональность, эгоизм, упрямство, иногда переходящее в замкнутость.

ЛППП - самый сильный тип характера. Вас трудно в чем-либо убедить. Вы с трудом меняете свою точку зрения. Но в то же время вы энергичны, упорно добиваетесь поставленных целей.

ЛПЛЛ - вы настойчивы в достижении поставленных целей. Обладатели такого характера - люди неуступчивые, переубедить их порой оказывается невозможно. Они склонны к самоанализу, с трудом находят себе новых друзей.

ПЛПП - у вас легкий характер. Вы счастливо умеете избегать конфликтов, любите путешествовать. Легко находите друзей. Однако вы часто меняете свои увлечения.

ПЛЛЛ - вам присущи непостоянство и независимость, желание все сделать самому. Способность анализировать помогает вам успешно решать сложные задачи. Обычно вы кажетесь мягким, но становитесь требовательным и даже жестоким, когда доходит до дела.

Практическое занятие №20

Тема: Создание документа с использованием языка PHP.

Цели:

1. приобретение навыков создания и управления базой данных с помощью программы *phpMyAdmin*;
2. закрепление навыков использования *MySQL*.

Структура базы данных TOVARS:

1. Таблица *tovar*, содержит учетные записи товаров

| № | Название поля | Описание | Тип |
|---|---------------|-------------------|--------------|
| 1 | id | Поле-счетчик | INT |
| 2 | name | Название товара | VARCHAR (20) |
| 3 | cost | Стоимость | INT |
| 4 | kol | Количество товара | INT |
| 5 | date | Дата реализации | DATE |

Пример записей:

| id | name | cost | kol | date |
|----|---------------|------|-----|----------|
| 1 | Хлеб столовый | 24 | 100 | 25.03.10 |
| 2 | Хлеб ржаной | 20 | 50 | 27.03.10 |

Выбор данных:

```
SELECT column1,... FROM table WHERE definition ORDER BY col_name
```

Добавление данных:

```
INSERT INTO table VALUES (value1, ...)
```

Удаление данных:

```
DELETE FROM table WHERE definition
```

Основные функции для работы с *MySQL*:

int mysql_connect(string hostname, string username, string password) - создать соединение с *MySQL*. Функция возвращает параметр типа *int*, который больше 0, если соединение прошло успешно, и равен 0 в противном случае.

hostname – имя хоста, на котором находится база данных.

Username – имя пользователя.

Password – пароль пользователя.

int mysql_select_db(string database_name, int link_identifier) - выбрать базу данных для работы. Функция возвращает значение *true* или *false*

Database_name – имя базы данных.

link_identifier – *ID* соединения, которое получено в функции *mysql_connect*. (параметр необязательный, если он не указывается, то используется *ID* от последнего вызова *mysql_connect*)

int mysql_query(string query, int link_identifier) - функция выполняет запрос к базе данных. Функция возвращает *ID* результата или 0, если произошла ошибка.

query – строка, содержащая запрос *link_identifier* – см. предыдущую функцию.

int mysql_result(int result, int i, column) - функция возвращает значение поля в столбце *column* и в строке *i*.

int mysql_close(int link_identifier) - функция закрывает соединение с *MySQL*. Функция возвращает значение *true* или *false*.

link_identifier – см. выше.

3. Порядок выполнения работы

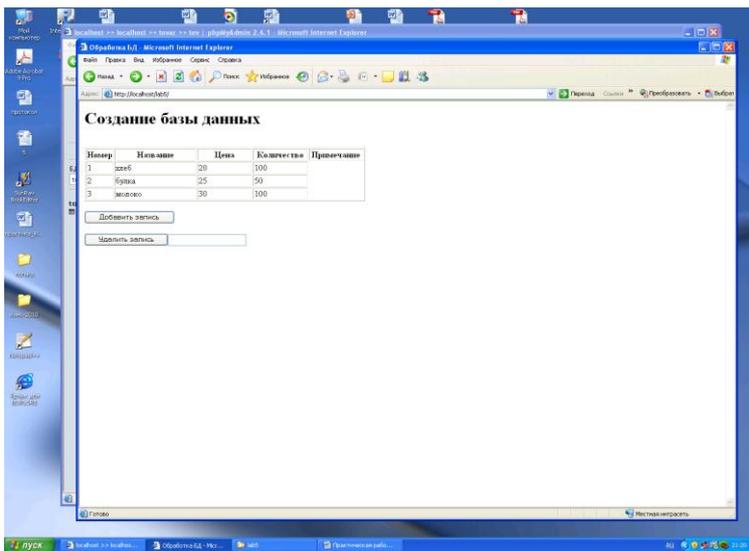
- Открыть программу phpMyAdmin: набрать в строке браузера localhost и выбрать пункт phpMyAdmin.
- В разделе Привилегии добавить нового пользователя homeuser со всеми правами.
- В поле Создать новую БД ввести имя базы TOVAR – Создать (Create Database)
- Создать таблицу tovar:
 - В поле Имя ввести tovar;
 - В поле Поля – число 5;
 - Кнопка Пошел.
- Указать название полей таблицы и их тип. В поле Дополнительно указать auto_increment (автоматическое присвоение). Установить переключатель primary key(первичный ключ). Сохранить результат.

Сервер: localhost ▶ БД: tovar ▶ таблица : tovar

Структура Обзор SQL Искать Вставить Экспорт Операции Очистить

| | Поле | Тип | Сравнение | Атрибуты | Ноль | По умолчанию | Дополнительно | Действие |
|--------------------------|------|-------------------------------|-----------|----------|------|--------------|----------------|----------|
| <input type="checkbox"/> | id | int(11) | | | Да | NULL | auto_increment | |
| <input type="checkbox"/> | name | varchar(20) cp1251_general_ci | | | Да | | | |
| <input type="checkbox"/> | cost | int(11) | | | Да | | | |
| <input type="checkbox"/> | kol | int(11) | | | Да | | | |
| <input type="checkbox"/> | prim | varchar(20) cp1251_general_ci | | | Да | | | |

- Ввести 4 записи для данных базы.
- Создать папку lab4. В ней файл index.php, в котором описать код вывода данных таблицы на экран.



Пример кода:

```
<?php
echo "<h1>Создание базы данных</h1><br>";
echo "<table border=1><tr align=center><td width=10%><b>Идентификатор</b></td><td width=30%><b>Название</b></td><td width=20%><b>Цена</b></td><td width=20%><b>Количество</b></td><td width=20%><b>Примечание</b></td></tr>";
$mysqlhost="localhost"; $mysqluser="homeuser"; $mysqlpass=""; $db="TOVARS";
// соединение с базой данных
mysql_connect($mysqlhost,$mysqluser,$mysqlpass) or die ("нет доступа!".mysql_error());
// выбирает базу для последующей работы
mysql_select_db($db) or die ("нет соединения".mysql_error());
$zap="SELECT * FROM tovar ORDER BY id";
```

```
// выполнение SQL-запроса выбора данных из БД
$zap_res=mysql_query($zap);
while (list($id, $name, $scena, $kol, $prim)=mysql_fetch_row($zap_res))
{
    echo "<tr>
    <td>$id</td> <td>$name</td> <td> $scena</td> <td>$kol</td> <td>$prim</td> </tr>";
}
echo "</table>"; ?>
```

7. Разместить 2 кнопки Добавить запись и Удалить запись № и текстовое поле для указания № удаляемой записи.
9. Создать файл insert.php, в котором разместить форму для ввода данных в таблицу.

Пример кода:

```
<?php
if (isset($_REQUEST))
{
    foreach($_REQUEST as $key=>$val)
    { $key=$val; }
}
$sqlhost="localhost"; $sqluser="homeuser"; $sqlpass=""; $bd="TOVARS";
mysql_connect($sqlhost,$sqluser,$sqlpass) or die ("нет доступа!".mysql_error());
mysql_select_db($bd) or die ("нет соединения".mysql_error());

$zap="INSERT INTO tovar( name, cost, kol, prim) VALUES ($name, $scena, $kol, $prim)";
$zap_res=mysql_query($zap);
if ($zap_res==true)
    echo "Запись успешно добавлена"; else echo "Ошибка при записи данных";
?>
```

10. Создать файл delete.php, в котором описать код для удаления записи по заданному номеру.

Пример кода:

```
<?php
if (isset($_REQUEST))
{
    $num=$_REQUEST[num];
}

$sqlhost="localhost"; $sqluser="homeuser"; $sqlpass=""; $bd="TOVARS";
mysql_connect($sqlhost,$sqluser,$sqlpass) or die ("нет доступа!".mysql_error());
mysql_select_db($bd) or die ("нет соединения".mysql_error());

$zap="DELETE FROM 'tovar' WHERE id = $num ";
$zap_res=mysql_query($zap);
if ($zap_res==true)
    echo "Запись успешно удалена";
else echo "Ошибка при удалении данных";
```

?>

Описание функций работы с БД – http://rusphp.chat.ru/34_MySQLFunctions.html или на сайте PHP.SU