

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПУТЕЙ СООБЩЕНИЯ (МИИТ)

ЮРИДИЧЕСКИЙ ИНСТИТУТ

А.И. ДМИТРИЕВ
С.Л. ЛОБАЧЁВ
О.А. МАЛЫГИН

КОМПЬЮТЕРНЫЕ СЕТИ,
МУЛЬТИМЕДИАТЕХНОЛОГИИ
И ПРОГРАММИРОВАНИЕ

*Методические указания
по выполнению практических
(лабораторных) работ*

МОСКВА
2013

УДК 681.3
ББК 32.97
Д—53

Дмитриев, А. И. Компьютерные сети, мультимедиа технологии и программирование : методические указания по выполнению практических (лабораторных работ) / А. И. Дмитриев, С. Л. Лобачёв, О. А. Малыгин. — М. : Юридический институт МИИТа, 2013. — 115 с.

Настоящий сборник содержит методические указания по выполнению практических (лабораторных) работ для студентов Юридического института МИИТа по следующим дисциплинам:

- Компьютерные системы, Интернет и мультимедиа технологии;
- Компьютерные системы и сети;
- Вычислительная техника и программирование.

Отдельные работы могут выполняться в рамках других учебных дисциплин по решению ведущего преподавателя, согласованному с заведующим кафедрой «Информационное обеспечение в правовой сфере» ЮИ МИИТа.

ОБЩИЕ СВЕДЕНИЯ

Цель выполнения практических работ – научить студентов применять полученные на лекциях знания как в процессе обучения, так и в будущей самостоятельной работе.

По ходу проведения работ студенты должны выполнить все задания по каждой конкретной работе. Полученные результаты должны быть отражены в отчёте по работе и в виде файла записаны в папку с номером учебной группы. Особенностью выполнения практических (лабораторных) работ по указанным дисциплинам является то, что студенты имеют доступ к специальному учебному серверу, и все компьютеры имеют выход в Интернет.

Выполнив все практические работы, студент должен освоить принципы работы с ОС и современным сетевым программным обеспечением персонального компьютера, получить навыки поиска информации в Интернет и использования облачных технологий, создания сайтов, программирования на языке VBA и создания мультимедиаресурсов.

Практические (лабораторные работы) охватывают общие и специальные темы перечисленных выше курсов, в том числе:

- систему адресации в ЛВС (локальные адреса компьютеров);
 - особенности передачи данных в сетях;
 - принципы маршрутизации в Интернете;
 - создание HTML-документов (сайтов)
 - создание мультимедиафайлов
- и ряд других.

Все работы рассчитаны на 2 академических часа. Для выполнения работ на компьютерах должна быть установлена ОС Windows 7 (XP) и браузер Google Chrome.

ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ (ЛАБОРАТОРНЫХ) РАБОТ

1. Изучение работы с интерпретатора командных строк ОС Windows.
2. Определение сетевых параметров рабочей станции.
3. Адресация и маршрутизация в сетях TCP/IP.
4. Веб-сервисы и облачные технологии Интернета (на примере Google).
5. Изучение методов работы с системой визуальной on-line маршрутизации.
6. Информационный поиск в Интернете.
7. Информационный поиск профильной информации в Интернете.
8. Изучение языка гипертекстовой разметки HTML-документов.
9. Изучение основ создания многостраничных сайтов.
10. Разработка таблиц в составе HTML документов.
- 11 Изучение рекордера макросов VBA в MS WORD.
12. Изучение среды программирования VBA в MS WORD.
13. Освоение среды программирования VBA в MS WORD.
14. Использование условного оператора.
15. Создание GIF-анимации в пакете Adobe Photoshop.
16. Создание видеофайла в формате MP4 из графических файлов в формате JPG.
17. Создание GIF-анимации из видеофайла в формате WMV.

Работа № 1

ИЗУЧЕНИЕ РАБОТЫ С ИНТЕРПРЕТАТОРА КОМАНДНЫХ СТРОК WINDOWS

Цель работы: Освоение основных способов работы с командной строкой в среде Windows.

Введение. В операционной системе Windows, как и в других операционных системах, интерактивные (набираемые с клавиатуры и сразу же выполняемые) команды выполняются с помощью командного интерпретатора, иначе называемого командным процессором или оболочкой командной строки (command shell). Командный интерпретатор или оболочка командной строки — это программа, которая, находясь в оперативной памяти, считывает набираемые на клавиатуре команды и обрабатывает их.

Для запуска командного интерпретатора (открытия нового сеанса командной строки) можно выбрать пункт **Пуск/Выполнить**, ввести имя файла **Cmd.exe** и нажать кнопку **ОК**. В результате откроется новое окно, в котором можно запускать команды и видеть результат их работы.

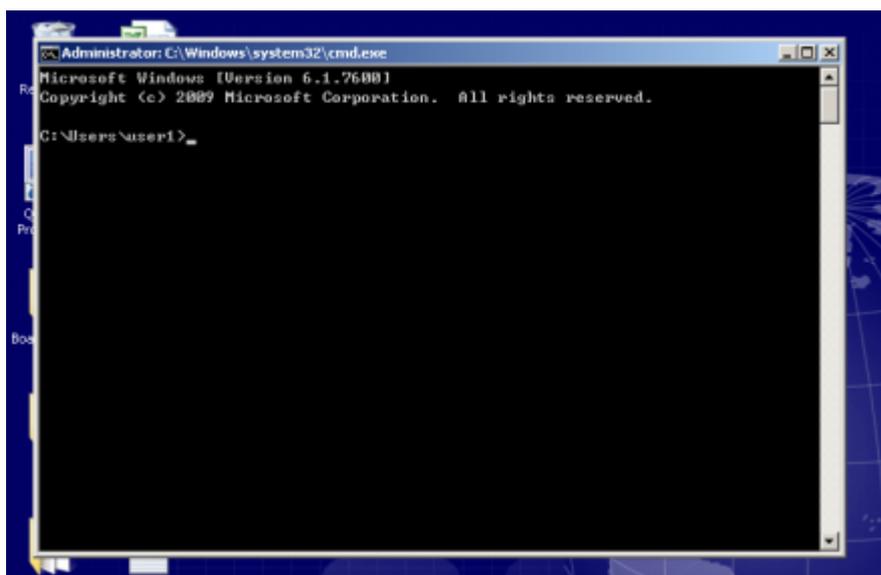


Рис. 1.1. Стартовое окно командного интерпретатора

Диалог компьютера с пользователем ведётся в режиме:

[Приглашение компьютера]>[команда пользователя] <Enter>

Стандартное приглашение командной строки Windows включает в себя текущий диск, каталог и знак «больше» (>), что удобно при работе с файлами и каталогами.

Некоторые команды распознаются и выполняются непосредственно самим командным интерпретатором. Такие команды называются внутренними (например, **copy** или **dir**). Другие команды операционной системы представляют собой

отдельные программы, расположенные по умолчанию в том же каталоге, что и **Cmd.exe**, которые Windows загружает и выполняет аналогично другим программам. Такие команды называются внешними (например, **more** или **xcopy**).

Многие команды Windows имеют большое количество дополнительных параметров и ключей, запомнить которые зачастую бывает трудно. Большинство команд снабжено встроенной справкой, в которой кратко описываются назначение и синтаксис данной команды. Получить доступ к такой справке можно путём ввода команды с ключом **/?** или **/help**.

Для перехода из одного каталога в другой служит команда **cd**. Если команда подаётся без параметров, то отображаются имена текущего диска и каталога (те, что в приглашении компьютера).

Для записи в файл информации, выводимой на экран при использовании команды пользователя, используется символ «>» с последующим именем файла (если файла нет, то он будет создан, если файл существует, то содержимое файла будет перезаписано) или «>>» (если файла нет, то он будет создан, если файл существует, то информация добавится в конец файла).

Например,

```
[Приглашение компьютера]>Help>c:\Temp\help-list.txt <Enter>
```

Команда создаст файл с именем **help-list.txt** в директории **c:\Temp** и запишет туда следующую информацию:

== начало файла ==

Для получения сведений об определённой команде наберите **HELP <имя команды>**

ASSOC – вывод либо изменение сопоставлений по расширениям имён файлов.

ATTRIB – отображение и изменение атрибутов файлов.

BREAK – включение и выключение режима обработки комбинации клавиш **CTRL+C**.

BCDEDIT – выбор свойств в базе данных загрузки для управления начальной загрузкой.

.....

TYPE Вывод на экран содержимого текстовых файлов.

VER Вывод сведений о версии Windows.

VERIFY Установка режима проверки правильности записи файлов на диск.

VOL Вывод метки и серийного номера тома для диска.

XCOPY Копирование файлов и деревьев каталогов.

WMIC Вывод сведений WMI в интерактивной среде.

Дополнительные сведения о программах приведены в описании программ командной строки в справке.

== конец файла ==

[Приглашение компьютера]>Help ver>c:\Temp\ver.txt<Enter>

Команда создаст файл с именем **ver.txt** в директории **c:\Temp** и запишет туда следующую информацию:

== начало файла ==

Вывод версии Windows.

VER

== конец файла ==

[Приглашение компьютера]>ver>>c:\Temp\ver.txt<Enter>

Команда допишет в файл с именем **ver.txt** в директории **c:\Temp** и содержимое файла будет следующим:

== начало файла ==

Вывод версии Windows.

VER

Microsoft Windows [Version 6.1.7601]

== конец файла ==

Указания по выполнению работы. Выполнение работы начинается с создания файла отчета по форме, приведенной в *приложении 1*, и заполнении ее личными данными студента (дисциплина, группа, ФИО). Именем файла с отчетом должна быть фамилия студента с его инициалами.

После создания файла отчета можно приступить к выполнению работы.

1. Получение сведений о командах режима командной строки с помощью команды **help**:

– получить общий список команд

– получить детальную информацию о трех-четырёх командах (по указанию преподавателя), для чего набрать

help <имя команды>.

2. Ознакомиться со списком и синтаксисом команд операционной системы Windows.

3. Добавить в файл отчёта текст справок указанных в п. 1 команд.

4. Добавить в файл отчёта результаты выполнения следующих команд: **ver, dir, chkdsk, tasklist**.

ПРИМЕЧАНИЕ. Для копирования текста справок нескольких команд из окна командной консоли в файл отчета – выполните следующие действия:

- щёлкнуть правой кнопкой мыши на заголовке окна командной консоли и в контекстном меню выбрать пункт **<Изменить>**
- пометить, а затем с помощью мыши выделить необходимый фрагмент текста и нажать **<Enter>** (операция копирования в буфер).
- перейти в файл отчета и вставить ранее скопированный фрагмент.

Другим способом копирования полученных результатов в отчет является использование комбинации **Alt-PrтSc**.

5. Добавить результаты выполнения следующих команд: **ver, driverquery, tasklist** из окна командной консоли в отчёт.

6. Записать файл отчёта в папку преподавателя.

Контрольные вопросы

1. Как называется программа, обеспечивающая выполнение команды, набираемой на клавиатуре?
2. Какой командой можно вывести на экран список папок и подпапок?
3. Какой командой можно вывести на экран справочную информацию о командах Windows?
4. Какие действия надо выполнить для запуска командного интерпретатора (открытия нового сеанса Командной строки)?
5. Какое назначение имеют дополнительные параметры и ключи в командах Windows?
6. Какой командой завершается работа с командным интерпретатором Windows?
7. Как можно узнать версию Windows средствами командного интерпретатора?
8. В чем состоит разница в действиях компьютера при записи в файл информации, выводимой на экран при использовании символов «>» или «>>»?
9. Как очистить экран при работе с командным интерпретатором?
10. Как создать новую папку средствами командного интерпретатора?

Литература

1. Справочник по параметрам командной строки, электронный ресурс: <http://technet.microsoft.com/ru-ru/library/cc778084.aspx>
2. Общие сведения о командной оболочке, электронный ресурс: [http://technet.microsoft.com/ru-ru/library/cc737438\(v=ws.10\)](http://technet.microsoft.com/ru-ru/library/cc737438(v=ws.10))
3. Никифоров, С. В. Введение в сетевые технологии / С. В. Никифоров. – М.: Финансы и статистика, 2007.

Работа № 2

ОПРЕДЕЛЕНИЕ СЕТЕВЫХ ПАРАМЕТРОВ РАБОЧЕЙ СТАНЦИИ

Цель работы: Освоение основных способов определения сетевых параметров и их настройки для подключения рабочей станции к сети передачи данных.

В результате выполнения лабораторной работы студенты должны изучить основные сетевые параметры, усвоить способы их определения, познакомиться с методом задания этих параметров, если даже у них нет прав администратора.

В конце лабораторной работы студенты должны создать файл отчета, содержащий IP-адрес, маску подсети, адреса шлюза (маршрутизатора) и DNS-сервера своей рабочей станции, и передать отчет преподавателю через сетевое окружение (сетевую папку).

Указания по выполнению работы. Все выполняемые действия должны отражаться в отчете, форма которого приведена в приложении к настоящей работе. Наименование файла с отчетов должно соответствовать фамилии студента.

Перед началом работы следует определить версию ОС Вашего компьютера, для чего необходимо выбрать иконку «Мой компьютер» на рабочем столе и нажатием правой клавиши «мышки» в появившемся меню выберите пункт «Свойства» (рис. 2.1).

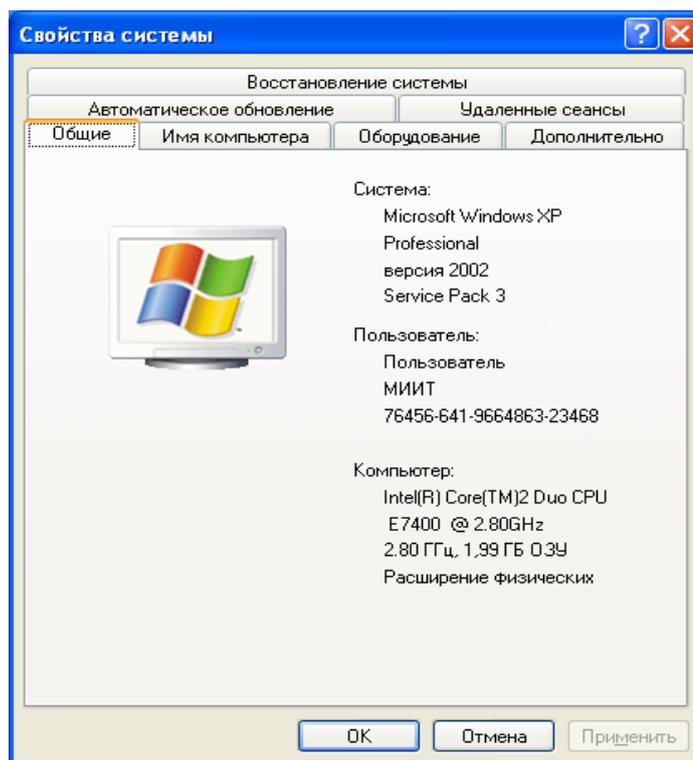


Рис. 2.1. Окно «Свойства системы»

Зафиксируйте версию ОС в отчете. При выполнении работ, используйте указания соответствующие определённой вами ОС.

Порядок выполнения работы

1. Определите имя NetBIOS вашего компьютера и компьютера преподавателя, для чего проделайте следующие действия:

меню **Пуск – Выполнить – набрать «cmd» – набрать «nbtstat –n»**

Занесите имя Вашего компьютера в отчёт.

2. Определите следующие параметры протокола TCP/IP вашего компьютера: IP-адрес сетевого адаптера; сетевую маску (маску подсети); адрес шлюза по умолчанию; адрес основного и вспомогательного сервера DNS, для чего проделайте следующие действия:

(для Windows XP) меню **Пуск - Панель управления – Сетевые подключения – Подключения по локальной сети – Поддержка – Подробности.**

(для Windows 7) меню **Пуск – Панель управления – Центр управления сетями... – Тип доступа/Подключения – Сведения.**

Зафиксируйте все параметры в таблице отчёта

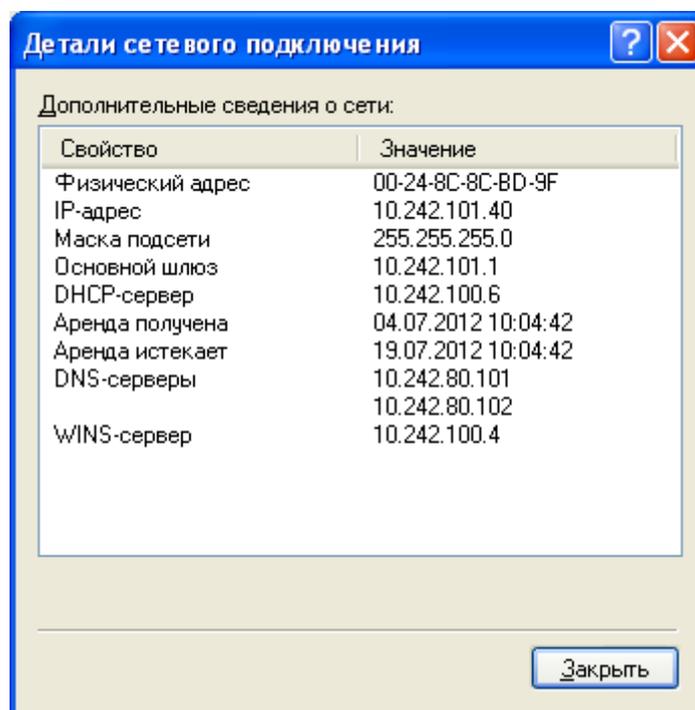


Рис. 2.2. Параметры сетевого подключения

3. Определите для Вашего компьютера физический адрес сетевого адаптера и доменное имя. Для этого в командной строке (меню **Пуск – Программы – Стандартные – Командная строка**) наберите **ipconfig /all** (рис. 2.3) и получите детальные параметры сетевого подключения.

Зафиксируйте результаты в отчёте.

4. Определите, открыт ли сетевой доступ к диску вашего компьютера. Для этого выберите иконку **«Мой компьютер»** и затем щёлкните правой клавишей на значке диска. В открывшемся контекстном меню выберите пункт **«Общий доступ и безопасность»** (Доступ). Определите также ёмкость всех дисков Вашего компьютера через меню **«Свойства»**.

Зафиксируйте результаты в отчёте.

```
Командная строка
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\Пользователь>ipconfig/all

Настройка протокола IP для Windows

Имя компьютера . . . . . : 6404Aws02
Основной DNS-суффикс . . . . . :
Тип узла . . . . . : гибридный
IP-маршрутизация включена . . . . . : нет
WINS-прокси включен . . . . . : нет
Порядок просмотра суффиксов DNS . . . . . : msk.miit.ru

Подключение по локальной сети - Ethernet адаптер:

Ethernet NIC
DNS-суффикс этого подключения . . . . . : msk.miit.ru
Описание . . . . . : Realtek RTL8102E Family PCI-E Fast E
Физический адрес . . . . . : 00-24-8C-8C-BD-9F
Dhcp включен . . . . . : да
Автонастройка включена . . . . . : да
IP-адрес . . . . . : 10.242.101.40
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . : 10.242.101.1
DNS-сервер . . . . . : 10.242.100.6
DNS-серверы . . . . . : 10.242.80.101
                        10.242.80.102
Основной WINS-сервер . . . . . : 10.242.100.4
Аренда получена . . . . . : 4 июля 2012 г. 10:04:42
Аренда истекает . . . . . : 19 июля 2012 г. 10:04:42

C:\Documents and Settings\Пользователь>
```

Рис. 2.3. Детальные параметры сетевого подключения

5. Определите быстродействие и объем памяти вашего компьютера. Для этого щёлкните правой кнопкой мыши на иконке «**Мой компьютер**» и в открывшемся контекстном меню выберите пункт «**Свойства**», а затем вкладку «**Общие**».

Зафиксируйте результаты в отчёте.

6. В сетевом окружении найдите общую папку с номером Вашей учебной группы на компьютере преподавателя. Положите свой файл с отчётом в эту папку. Файл должен иметь название, состоящее из фамилии студента и его инициалов, например:

ИвановИИ

7. По указанию преподавателя отчет может быть дополнен ответом на контрольные вопросы.

Контрольные вопросы

1. При определении сетевых параметров рабочей станции средствами Windows XP и Windows 7 будут ли они иметь одинаковые значения? Если нет – то в чем будет разница?
2. Может ли пользователь изменить физический адрес и, если да, то как?
3. Может ли изменить физический адрес компьютера пользователь администратор сети?
4. Может ли изменить доменное имя компьютера пользователь?
5. Может ли администратор сети изменить доменное имя компьютера пользователя?
6. Может ли пользователь сделать диск своего компьютера открытым?
7. Что означает «открыть сетевой доступ к диску вашего компьютера»?
8. Каковы функции сервера DNS?

Литература

1. Никифоров, С. В. Введение в сетевые технологии / С. В. Никифоров. – М.: Финансы и статистика, 2007.
2. Таненбаум, Э. Компьютерные сети / Э. Таненбаум. – СПб.: Питер, 2007.
3. Олифер, В.Г. Компьютерные сети. 4-е изд. / В. Г Олифер, Н. А. Олифер. – СПб.: Питер, 2010.

Работа № 3

АДРЕСАЦИЯ И МАРШРУТИЗАЦИЯ В СЕТЯХ ТСП/Р

Цель работы: Изучить на практике принципы маршрутизации и адресации в сетях, а также установить промежуточные узлы, обеспечивающие маршрутизацию между двумя хостами.

В результате выполнения данной работы студенты должны изучить принцип адресации в сетях, познакомиться со структурой файла, hosts.txt, являющегося прообразом DNS, выявить все промежуточные узлы, соединяющие заданные два узла.

В конце лабораторной работы студенты должны создать файл, содержащий полное описание команд ping и traceroute (pathing), а также доменные имена и IP-адреса всех узлов, соединяющих его рабочую станцию с сервером do.ui-miit.ru.

ПРИМЕЧАНИЕ. При технической невозможности выполнения ряда заданий, зафиксировать сообщение выводимое на экран компьютера в отчете.

Введение. Утилита traceroute осуществляет трассировку маршрута, посылая серию обычных IP-пакетов в конечную точку изучаемого маршрута. Идея метода состоит в следующем. Значение времени жизни (TTL) первого отправляемого пакета устанавливается равным 1. Когда протокол IP первого маршрутизатора принимает этот пакет, то он в соответствии со своим алгоритмом уменьшает значение TTL на 1 и получает 0. Маршрутизатор отбрасывает пакет с нулевым временем жизни и возвращает узлу-источнику ICMP-сообщение об ошибке истечения времени дейтаграммы (значение поля типа равно 11) вместе с заголовком IP и первыми 8 байтами потерянного пакета. Получив ICMP-сообщение о причине недоставки пакета, утилита traceroute запоминает адрес первого маршрутизатора (который извлекает из заголовка IP-пакета, несущего ICMP-сообщение).

ПРИМЕЧАНИЕ. ICMP (англ. Internet Control Message Protocol — протокол межсетевых управляющих сообщений) — сетевой протокол, входящий в стек протоколов ТСП/Р. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают. Также на ICMP возлагаются некоторые сервисные функции.

Затем traceroute посылает следующий IP-пакет, но теперь со значением TTL, равным 2. Этот пакет благополучно проходит первый маршрутизатор, но «умирает» на втором, о чем немедленно отправляется аналогичное ICMP-сообщение об ошибке истечения времени дейтаграммы. Утилита traceroute запоминает адрес второго маршрутизатора и т. д. Такие действия выполняются с каждым маршрутизатором вдоль маршрута вплоть до узла назначения или неисправного маршрутизатора. Такое рассмотрение работы утилиты traceroute весьма схематично, но и этого достаточно, чтобы оценить изящество идеи, лежащей в основе ее работы. Остальные ICMP-сообщения об ошибках имеют такой же формат и отличаются друг от друга только значениями полей типа и кода. Далее приведена копия

экранной формы, выведенной утилитой `tracert` (Windows) при трассировке хоста `ds.internic.net [198.49.45.29]`:

```
1 311 ms 290 ms 261 ms 144.206.192.100
2 281 ms 300 ms 271 ms 194.85.73.5
3 2023 ms 290 ms 311 ms moscow-m9-2-S5.relcom.eu.net [193.124.254.37]
4 290 ms 261 ms 280 ms MSK-M9-13.Relcom.EU.net [193.125.15.13]
5 270 ms 281 ms 290 ms MSK.RAIL-1-ATM0-155Mb.Relcom.EU.net [193.124.254.82]
6 300 ms 311 ms 290 ms SPB-RASCOM-1-E3-1-34Mb.Relcom.EU.net [193.124.254.78]
7 311 ms 300 ms 300 ms Hssi11-0.GW1.STK2.ALTER.NET [146.188.33.125]
8 311 ms 330 ms 291 ms 421.ATM6-0-0.CR2.STK2.Alter.Net [146.188.5.73]
9 360 ms 331 ms 330 ms 219.Hssi4-0.CR2.LND1.Alter.Net [146.188.2.213]
10 351 ms 330 ms 331 ms 412.Atm5-0.BR1.LND1.Alter.net [146.188.3.205]
11 420 ms 461 ms 420 ms 167.ATM8-0-0.CR1.ATL1.Alter.Net [137.39.69.182]
12 461 ms 441 ms 440 ms 311.ATM12-0-0.BR1.ATL1.Alter.Net [137.39.21.73]
13 451 ms 410 ms 431 ms atlantal-br1.bbnplanet.net [4.0.2.141]
14 420 ms 411 ms 410 ms viennal-br2.bbnplanet.net [4.0.3.154]
15 411 ms 430 ms 2514 ms viennal-nbr3.bbnplanet.net [4.0.3.150]
16 430 ms 421 ms 441 ms viennal-nbr2.bbnplanet.net [4.0.5.45]
17 431 ms 451 ms 420 ms cambridgel-br1.bbnplanet.net [4.0.5.42]
18 450 ms 461 ms 441 M C cambridgel-crl4.bbnplanet.net [4.0.3.94]
19 451 M C 461 M C 460 M C attbcsto11.bbnplanet.net [206.34.99.38]
20 501 M C 460 M C 481 M C shutdown.ds.internic.net [198.49.45.29]
```

Рис. 3.1. Результаты трассировки, выполненной с помощью `tracert`

Последовательность строк соответствует последовательности маршрутизаторов, образующих маршрут к заданному узлу. Первое число в строке — число хопов (шагов) до соответствующего маршрутизатора. Утилита `tracert` тестирует каждый маршрутизатор трижды, поэтому следующие три числа в строке — это значения RTT, вычисленные путём послыки трёх пакетов, время жизни которых истекло на этом маршрутизаторе. Если ответ от какого-либо маршрутизатора не приходит за заданное время, то вместо времени на экране печатается звёздочка (*).

Далее идут IP-адрес и доменное имя (если оно имеется) маршрутизатора. Видно, что почти все интерфейсы маршрутизаторов — поставщиков услуг Интернета зарегистрированы в службе DNS, а первые два, относящиеся к локальным маршрутизаторам, — нет. Обратите внимание на то, что время, указанное в каждой строке, это не время прохождения пакетов между двумя соседними маршрутизаторами, а время, за которое пакет проделывает путь от источника до соответствующего маршрутизатора и обратно. Так как ситуация в Интернете с загрузкой маршрутизаторов постоянно меняется, то время достижимости маршрутизаторов не всегда нарастает монотонно, а может изменяться достаточно произвольным образом.

Другим представителем группы ICMP-сообщений являются эхо-запросы и эхо-ответы, используемые в утилите `ping`.

Эхо-запрос и эхо-ответ, в совокупности называемые эхо-протоколом, представляют собой очень простое средство мониторинга сети. Компьютер или маршрутизатор посылает по сети ICMP-сообщение эхо-запроса, указывая в нем IP-адрес узла, достижимость которого нужно проверить. Узел, получивший эхо-

запрос, формирует и отправляет эхо-ответ отправителю запроса. Так как эхо-запрос и эхо-ответ передаются по сети внутри IP-пакетов, то их успешная доставка означает нормальное функционирование всей транспортной системы составной сети.

Утилита `ping` обычно посылает серию эхо-запросов к тестируемому узлу и предоставляет пользователю статистику об утерянных эхо-ответах и среднем времени реакции сети на запросы. Утилита `ping` выводит на экран сообщения следующего вида обо всех поступивших ответах:

```
# ping server1.citmgu.ru
Pinging server1.citmgu.ru [193.107.2.200] with 64 bytes of data:
Reply from 193.107.2.200: bytes=64 time=256ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=310ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=260ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=146ms TTL= 123
```

Рис. 3.2. Результаты выполнения утилиты `ping`

Из приведённой распечатки видно, что в ответ на тестирующие запросы, посланные узлу `server1.citmgu.ru`, было получено четыре эхо-ответа. Длина каждого сообщения составляет 64 байта.

В следующей колонке помещены значения времени оборота (RTT), т.е. времени от момента отправки запроса до получения ответа на этот запрос. Как видим, сеть работает достаточно нестабильно — время в последней строке отличается от времени во второй более чем в два раза. На экран выводится также оставшееся время жизни поступивших пакетов.

Из сказанного следует, что в основе работы популярных утилит мониторинга IP-сетей `ping` и `tracert` лежит протокол ICMP и формируемые им сообщения.

Содержание лабораторной работы

1. Познакомьтесь с предварительной информацией.
2. Изучите команды `ping` и `tracert`.
3. Познакомьтесь со всеми ключами этих команд.
4. Командой `ping` проверьте наличие связи с доменом `do.ui-miit.ru`.
5. Командой `tracert` определите IP-адреса и доменные имена узлов между вашим компьютером и сервером `do.ui-miit.ru`.
6. Сохраните отчёт о промежуточных узлах в файл.
7. Утилитой `nslookup` узнайте имя сервера `www.miit.ru`

В конце лабораторной работы создайте файл отчёта и поместите его в папку с номером группы на файл-сервере. Имя файла – фамилия студента.

Указания по выполнению работы

1. Командой «`tracert -?`» ознакомьтесь ключами команды `tracert`. Проверьте некоторые из них (например, «`-d`», «`-h`», «`-j`», «`-w`»).

ВНИМАНИЕ! Копии экранов каждого пункта работы необходимо занести в отчёт! (ПРИЛОЖЕНИЕ 1).

2. Командой «ping -?» ознакомьтесь ключами команды ping. Проверьте некоторые из них (например, «-t», «-a», «-n», «-l», «-w»).
3. Выполните команды: ping miit.ru;
Сохраните результат в отчете.
4. Выполните команду tracert miit.ru. Занесите адреса и имена промежуточных узлов, через которые осуществляется трассировка маршрута в отчет по работе.
5. Вызовите браузер Internet Explorer. В поле адреса наберите www.ip-ping.ru. Это сайт русского сервера, выполняющего некоторые функции сетевого операционного центра (Network Operation Center – NOC).

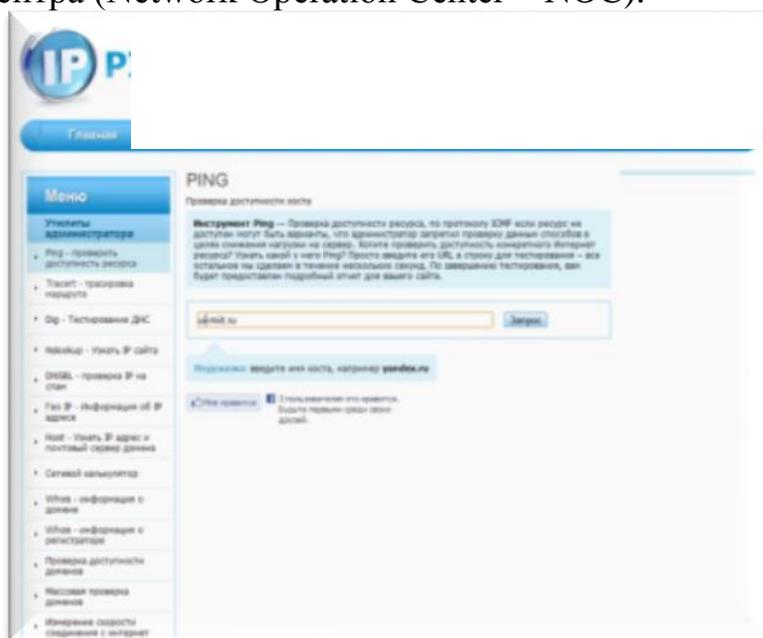


Рис. 3.3 Сайт IP- ping. Ru

6. Найдите на этом сайте операции Ping и Tracert. Выполните трассировку маршрута из Интернета к компьютеру DO.UI-MIIT.RU и к компьютеру EDU.UI-MIIT.RU – или иному по указанию преподавателя. Результат занесите в отчет.
7. Выполните команду tracert и ping до Мельбурнского, (www.mbs.unimelb.edu.au), Токийского (www.tufs.ac.jp) и Владивостокского (www.vvsu.ru) университетов. Сравните времена доступа и число «прыжков». Результаты занесите в отчет.
8. Проверьте работоспособность команды pathping для ОС, установленной на вашем хосте (компьютере).
9. В Интернете найдите справочный материал об утилите nslookup и занесите его в отчет.
10. С помощью nslookup определите имя DNS-сервера www.mii.ru и сохраните результат в отчете.

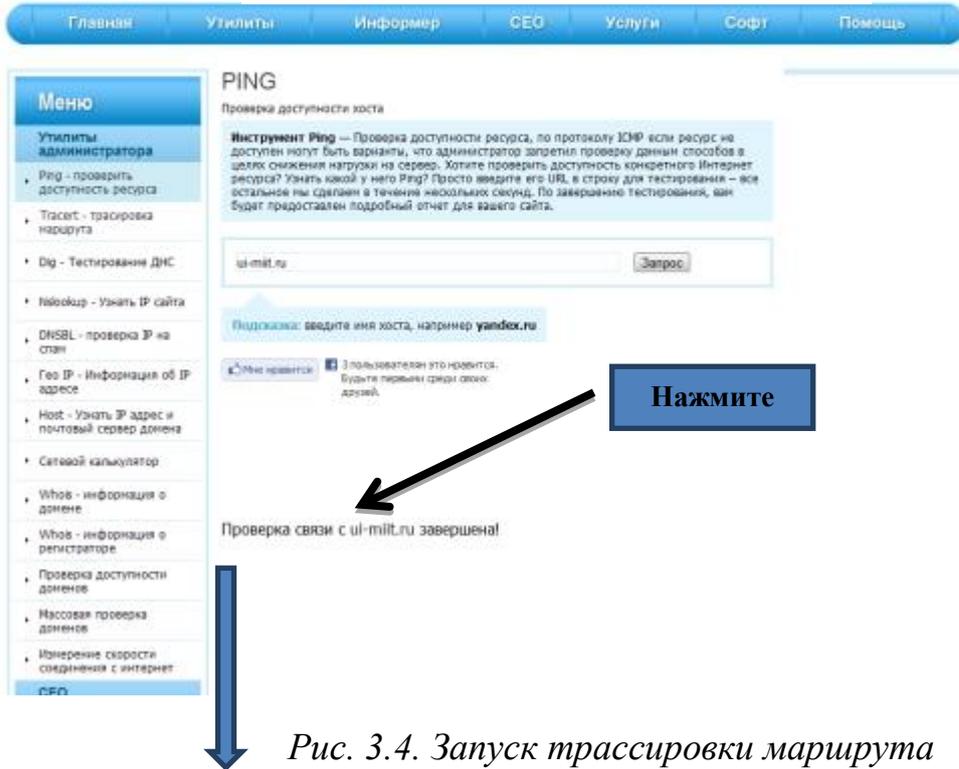


Рис. 3.4. Запуск трассировки маршрута

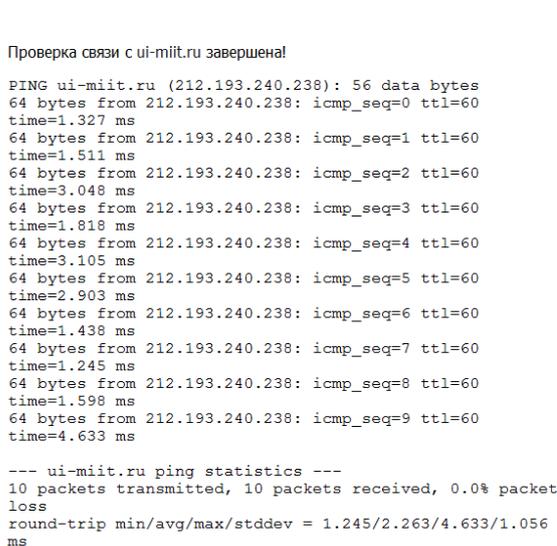


Рис. 3.5. Результаты трассировки маршрута

Контрольные вопросы

1. Какое представление имеет IP-адрес?
2. Опишите IP-адреса трёх нерегистрируемых сетей для классов А, В и С.
3. Может ли один хост иметь более одного IP-адреса? Обоснуйте свой ответ.
4. Для чего используется DNS?
5. Опишите принцип функционирования DNS.

6. Для чего используется команда ping?
7. В чем основное различие между протоколами TCP и UDP?
8. Для чего используется утилита nslookup?
9. Для чего используется команда tracer?
10. Как узнать свой IP-адрес, через который вы выходите в Интернет?
11. Кому адресовано ICMP-сообщение?

Литература

1. Никифоров, С. В. Введение в сетевые технологии / С. В. Никифоров. – М.: Финансы и статистика, 2007.
2. Таненбаум, Э. Компьютерные сети / Э. Таненбаум. – СПб.: Питер, 2007.
3. Леонтьев, В. Новейшая Энциклопедия ИНТЕРНЕТА 2009 / В. Леонтьев. – М. : ОЛМА Медиа Групп, 2009.
4. Олифер, В.Г. Компьютерные сети. 4-е изд. / В. Г Олифер, Н. А. Олифер. – СПб.: Питер, 2010.

Работа № 4

ВЕБ-СЕРВИСЫ И ОБЛАЧНЫЕ ТЕХНОЛОГИИ ИНТЕРНЕТА

Цель работы: Изучение на практике работы с веб-сервисами в сети Интернет на примере системы Google, создание и настройка собственного аккаунта, заведение почтового ящика, создание и работа с персональным онлайн контакт-листом, активизация, настройка и работа с облачным хранилищем, организация совместной работы с документами в облачных хранилищах.

В результате выполнения данной работы студенты должны изучить принципы организации веб-сервисов, методику создания и настройки аккаунтов, механизмы заведения и оптимальной настройки почтовых ящиков, особенности создания и работы с онлайн контакт-листами, принципы и методы работы с облачными хранилищами, возможности по организации совместной работы с документами в облачных хранилищах.

В конце лабораторной работы студенты должны создать файл отчёта и поместить его в папку с номером группы на файл-сервере. Имя файла – фамилия студента.

Введение. Учётная запись — запись, содержащая сведения, которые пользователь сообщает о себе некоторой компьютерной системе. Как синонимы в обиходе могут использоваться разговорные (учётка) и сленговые варваризмы (акк, аккаунт и эккаунт), от англ. account — учётная запись, личный счёт.

Учётная запись, как правило, содержит сведения, необходимые для идентификации пользователя при подключении к системе, информацию для авторизации и учёта. Это имя пользователя и пароль (или другое аналогичное средство аутентификации — например, биометрические характеристики). Пароль или его аналог, как правило, хранится в зашифрованном или хэшированном виде (в целях его безопасности).

Большинство пользователей Интернета воспринимают учётную запись, как личную страничку, кабинет, возможно даже место хранения личной и другой информации на определённом интернет-ресурсе (платформе).

Для повышения надёжности могут быть, наряду с паролем, предусмотрены иные средства аутентификации — например, специальный секретный вопрос (или несколько вопросов) такого содержания, что ответ может быть известен только пользователю. Такие вопросы и ответы также хранятся в учётной записи.

Учётная запись может содержать также дополнительные анкетные данные о пользователе — имя, фамилию, отчество, псевдоним, пол, национальность, расовую принадлежность, вероисповедание, группу крови, адрес e-mail, домашний адрес, рабочий адрес, номер домашнего телефона, номер рабочего телефона, номер сотового телефона, номер ICQ, идентификатор Skype и так далее. Конкретные категории данных, которые могут быть внесены в такую анкету, определяются создателями и (или) администраторами системы.

Учётная запись может также содержать одну или несколько фотографий или аватар пользователя.

Термин «облачные вычисления» (англ. Cloud Computing) произошёл от принятого графического обозначения выхода в Интернет в виде облачка. Идея облачных технологий такая: можно не иметь никаких программ на своём компьютере, а иметь только выход в Интернет. Всё основное располагается в Интернете, и то, что нужно, можно получить там. А вот платно или бесплатно — это будет зависеть от запросов.

Что дают нам облачные технологии? Для предприятий плюс облачных технологий состоит в том, что им не нужно покупать своё серверное оборудование, строить локальную сеть, заботиться о её работоспособности, тратиться на модернизацию и на зарплату сисадминам. Достаточно арендовать место на удалённом сервере с соответствующими параметрами: размером памяти, быстродействием, количеством клиентов. Потом, наполнить базы данных, раздавая имена и пароли пользователям до необходимого «куска» информации. И таким образом, можно получить доступ к ним из любой точки мира через обычный браузер.

За работоспособность и безопасность отвечают те, кто предоставляет услугу «облака», за соответствующую плату от клиента – провайдеры.

Основной аудиторией для таких сервисов являются корпоративные клиенты, заинтересованные в электронных системах документооборота, корпоративных почтовых ящиках и прочих виртуальных инструментах незаменимых, в современном офисе.

Индивидуальным пользователям, скорее всего, хватит бесплатных «облаков» — вроде того же Gmail от Google. Именно с этими системами и знакомит данная работа.

Создание аккаунта Google. Аккаунт Google позволяет работать с Gmail, Google+, YouTube и многими другими сервисами Google.

Аккаунт Google включает в себя адрес Gmail и профиль Google, который позволяет улучшить и персонализировать использование Google.

Чтобы создать аккаунт Google, необходимо перейти на страницу <http://accounts.google.com/signup>, для чего следует выйти на стартовую страницу <http://www.google.ru> и далее по ссылкам войти – зарегистрироваться.

Один аккаунт можно использовать для входа во все сервисы Google, вне зависимости от того, на странице какого продукта вы его создали.

Поскольку аккаунт Google предоставляет доступ к различным службам, таким как Gmail, YouTube и Google+, то в целях обеспечения безопасности аккаунта и удобства использования служб Google при регистрации необходимо указать определённую личную информацию (Ри.4.2.).

Имя – будет использоваться во всех службах Google. Например, оно будет отображаться рядом со всеми опубликованными материалами.

Имя пользователя – станет новым адресом Gmail для пользователя, и с его помощью пользователь сможет входить в Google. В имени пользователя можно использовать буквы, цифры и точки. Регистр не учитывается. Если пользователь затрудняется с выбором, система сама предложит ему несколько подходящих вариантов.

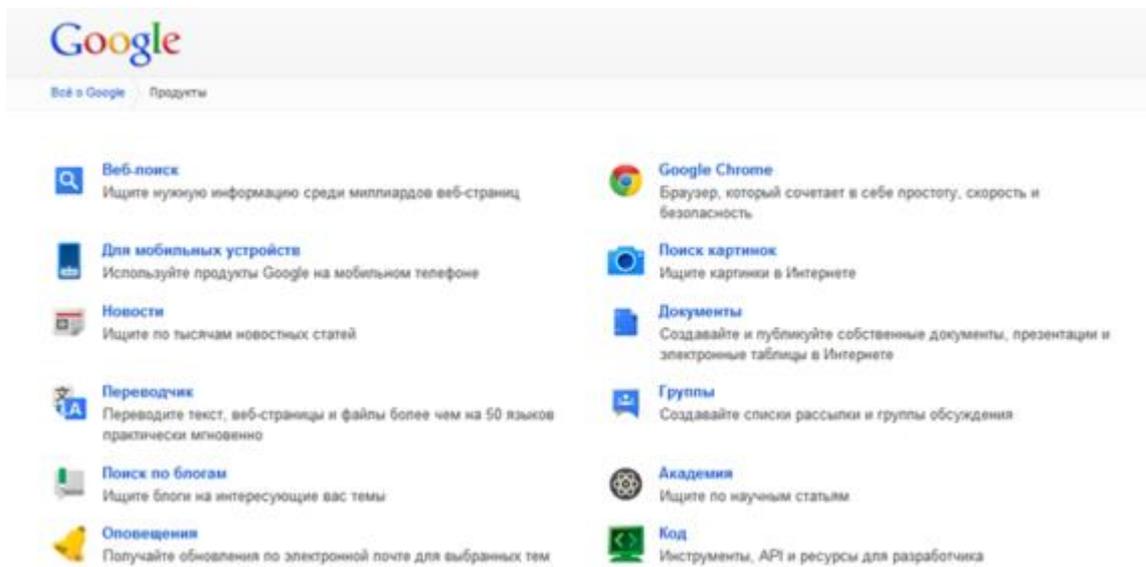


Рис. 4.1. Сервисы системы Google

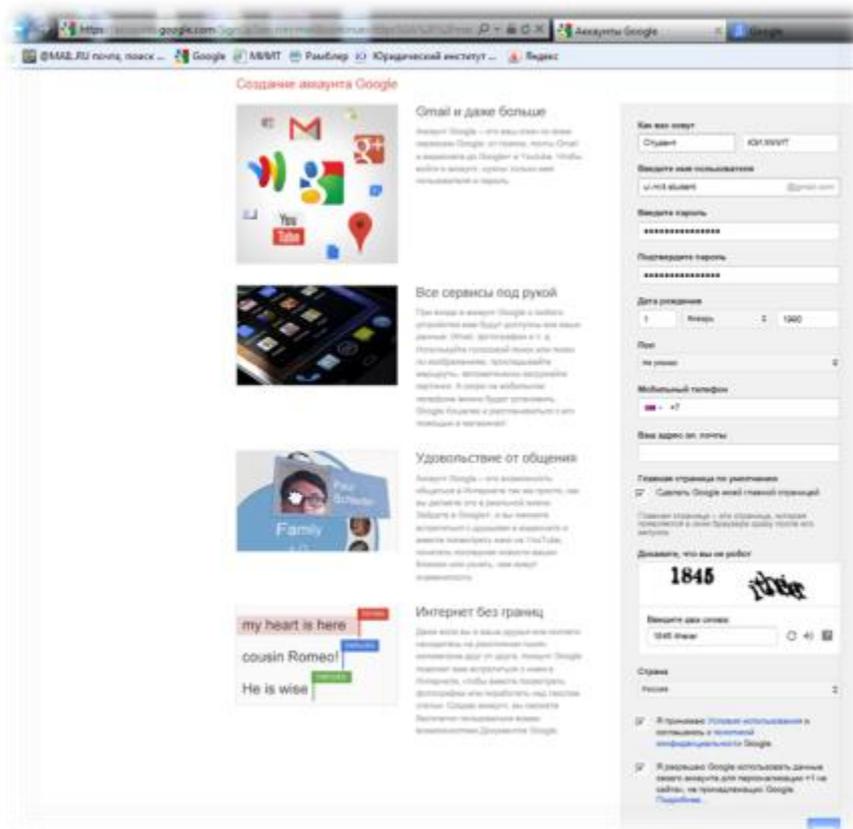


Рис. 4.2. Поля личных данных

Пароль позволит обеспечить безопасность аккаунта пользователя при условии выбора надёжного пароля.

Внимание! При создании пароля следуйте указанным ниже рекомендациям: Используйте надёжный пароль.

Не используйте один и тот же пароль на нескольких сайтах.

Проверьте пароль с помощью шкалы надёжности.

Пароль должен содержать не меньше 8 символов.

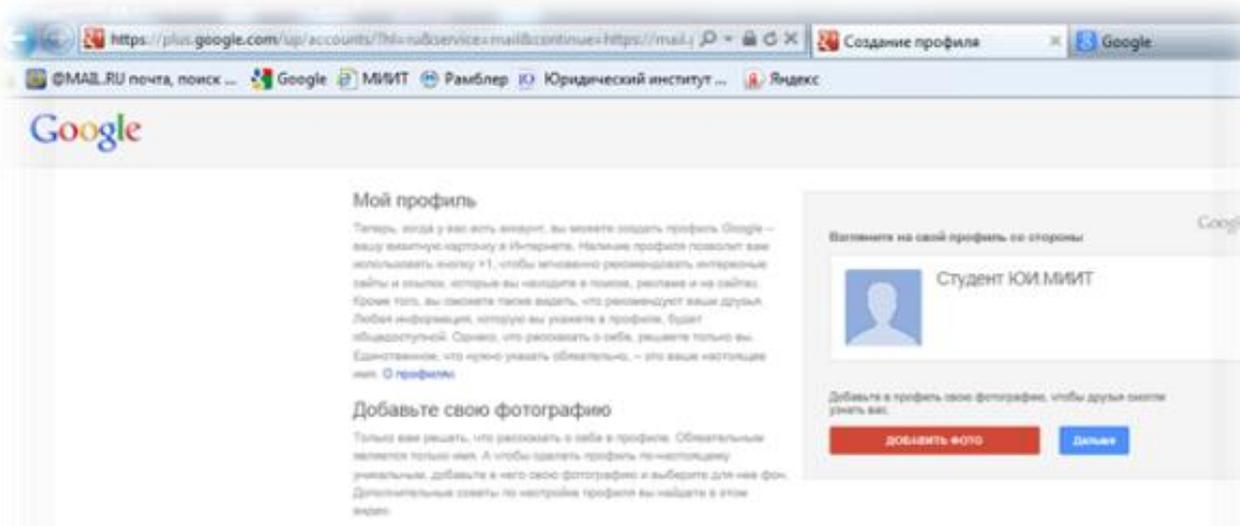


Рис. 4.3. Добавление фотографии в профиль

Дата рождения. Google задаёт настройки возрастных ограничений на основе указанной даты рождения. Отображаться она будет только с разрешения пользователя.

Пол. Эта информация помогает улучшить интерфейс служб Google. Например, в сообщениях о публикации фотографий будут использоваться местоимения «он» или «она». По умолчанию эта информация будет доступна всем пользователям, кто просматривает профиль пользователя или получает подобные сообщения. Изменить настройки отображения этих сведений можно сразу после создания аккаунта Google. Если вместо вариантов «Мужской» или «Женский» выбрать «Не указан» или скрыть эту информацию, личные местоимения использоваться не будут.

Мобильный телефон. Добавление номера мобильного телефона – это самый простой и надёжный способ обеспечения безопасности аккаунта. Поскольку телефон находится непосредственно у пользователя, он может легко его восстановить. Кроме того, номер телефона помогает предотвратить несанкционированный доступ к аккаунту. Это поле является необязательным, однако если у пользователя есть мобильный телефон, рекомендуется указать его номер.

Дополнительный адрес электронной почты, как и в случае с номером телефона, позволяет идентифицировать пользователя и защищать его аккаунт, а также отправлять ему важные оповещения. Необходимо использовать надёжный адрес электронной почты, который принадлежит лично пользователю. Это поле является необязательным, однако если у пользователя есть другой адрес электронной почты (не Gmail), настоятельно рекомендуется указать его в своём аккаунте.

Местоположение. В этом поле указывается страна или место жительства.

После создания аккаунта эту информацию можно проверить или изменить на странице настроек. (Обратите внимание: дату рождения изменить нельзя.)

Система позволяет «Добавить фото» (Рис. 4.3) для установления картинки профиля пользователя. В качестве картинки профиля Gmail можно выбрать любое изображение. Оно будет отображаться в общедоступном профиле и появляться,

когда другой пользователь наводит указатель мыши на имя пользователя в папке «Входящие», в контактах и в чате.

Для загрузки фотографии – нажмите кнопку «Добавить фото» и затем – «Создать фото профиля» и загрузите изображение с компьютера.

Перетащите рамку, чтобы обрезать фотографию или углы рамки, чтобы изменить размер.

Нажмите «Сделать главной фотографией профиля».

С помощью переключателей рядом со своей фотографией на странице настроек можно выбрать, кто может видеть фотографию: все пользователи Gmail или только те, с кем пользователь общается в чате.

Отображается для всех: картинку будут видеть все пользователи, с которыми пользователь ведет переписку. Если у пользователя есть общедоступный профиль Google без фотографии и выбран этот параметр, картинка будет использоваться в профиле Google и ее смогут видеть посетители.

Видна только тем пользователям, с которыми можно разговаривать в чате: картинку будут видеть только те, кто может общаться с пользователем в чате и видеть, что он находится в Сети.

Настройки почты

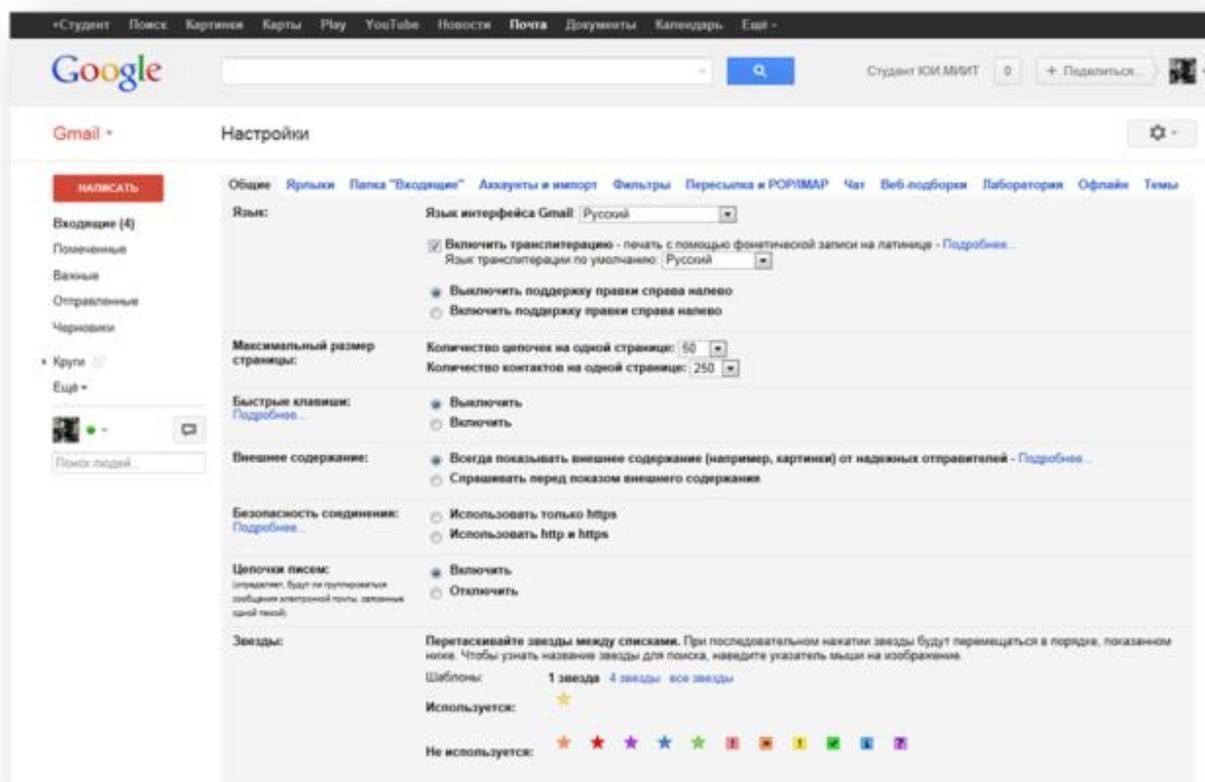


Рис. 4.4. Настройка параметров почтового ящика

Сервисы Google

Календарь

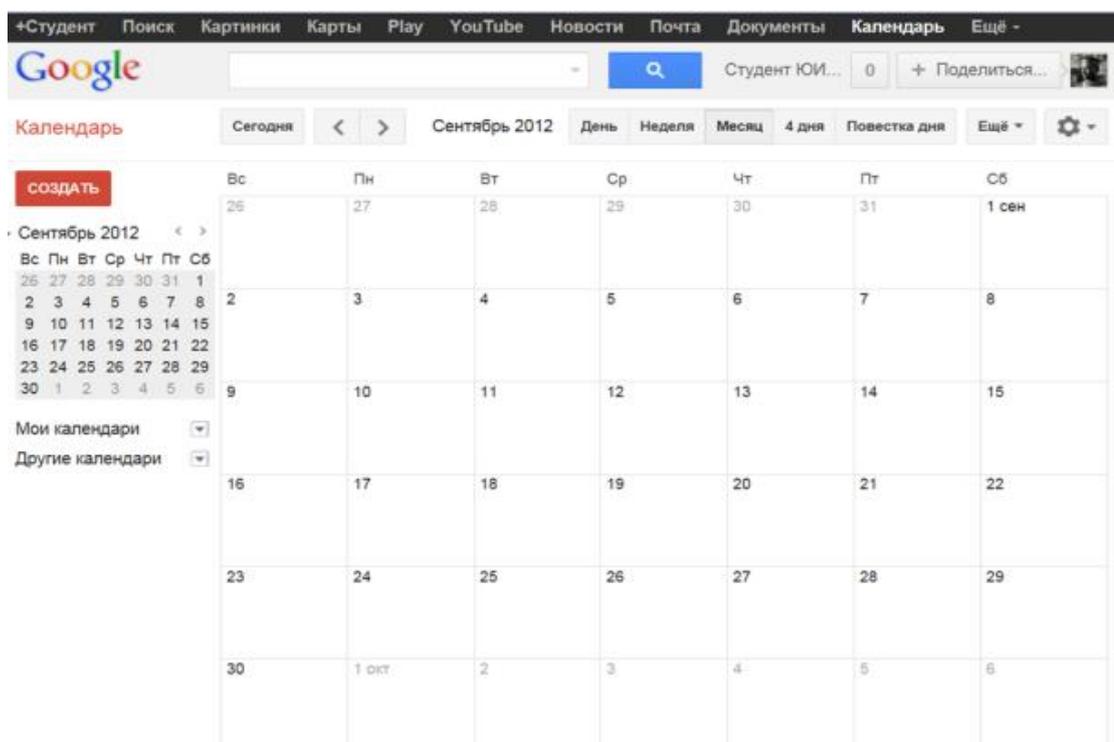


Рис. 4.5. Календарь системы Google

Документы (Google Диск)

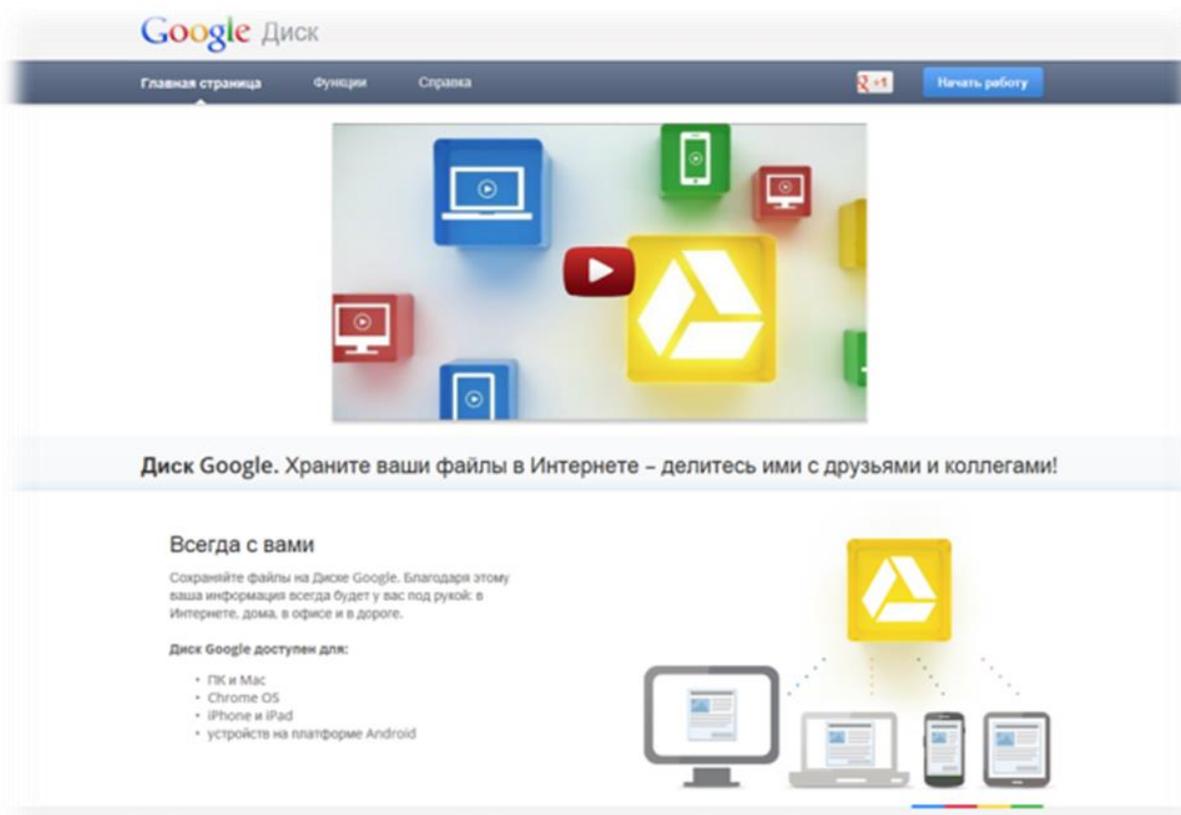


Рис. 4.6. Хранение файлов на диске

Карты Google

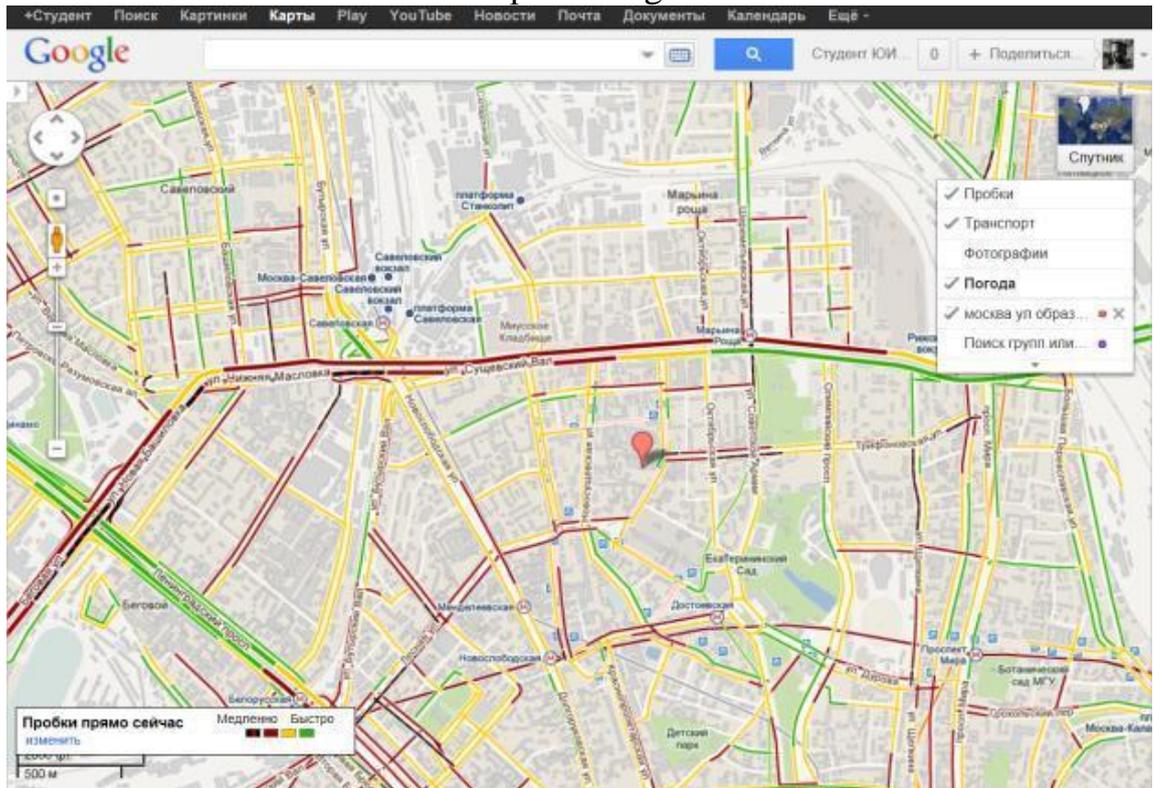


Рис. 4.7. Работа с картами

Переводчик

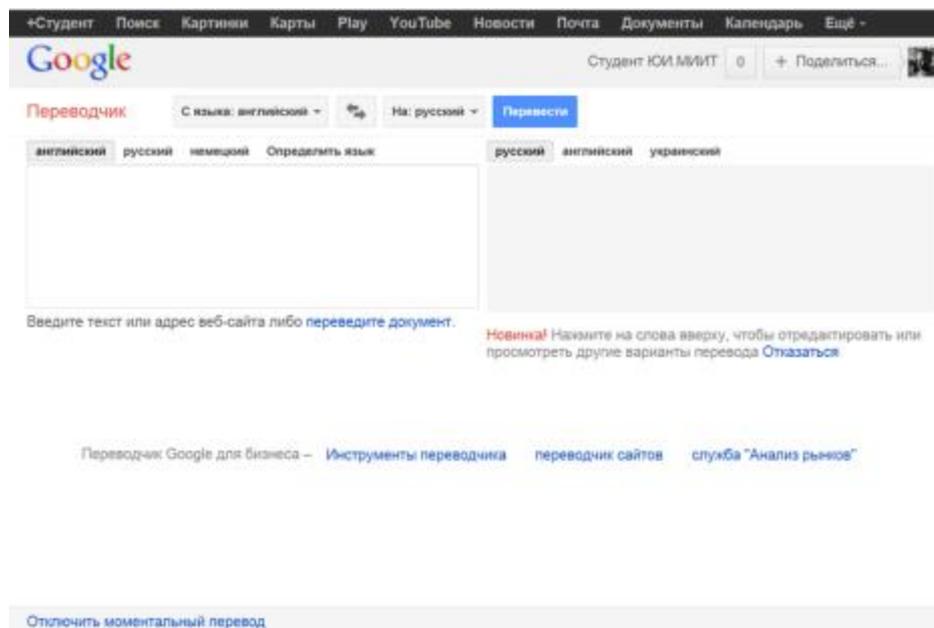


Рис. 4.8. Переводчик Google

После загрузки картинку можно будет просмотреть и изменить на странице настроек Gmail.

Система Google позволяет:

– настроить параметры почтового ящика (рис. 4.4);

- работать с календарем (рис. 4.5);
- хранить файлы (рис. 4.6);
- просматривать и работать с картами (рис. 4.7);
- использовать проводник (рис. 4.8) и др.

Указания по выполнению работы

1. Внимательно изучите всю предварительную информацию к данной работе.
2. Перейдите на страницу <http://accounts.google.com/signup> и создайте личный аккаунт.

ВНИМАНИЕ. Сохраните Ваши атрибуты для доступа в систему Google, они могут понадобиться при выполнении других работ.

3. С личного почтового ящика системы отправьте письмо преподавателю на указанный им адрес.

Содержание письма: Студент ПЕТРОВ И.И, гр. ЮТД-12 создал аккаунт в системе Google.

В теме письма укажите фамилию и инициалы: ПЕТРОВ И.И.

Приложите к письму скриншот папки «Входящие» Вашего почтового ящика.

4. Дополнительно настройте свой аккаунт, заполнив недостающие анкетные данные, прикрепите свою фотографию, сделайте скриншот вашего профиля.

5. Перейдите в Диск Google, создайте папку «My Labs», загрузите в неё свою фотографию, и для файла фотографии в контекстном меню выберите меню «Совместный доступ», настройте уровень доступа (Личный – доступ имеют только перечисленные ниже пользователи), добавьте пользователя (по адресу почты, предоставленному преподавателем) и откройте доступ с включённой опцией «Оповестить по электронной почте».

6. Зайдите в Карты Google, найдите точки месторасположения ЮИ МИИТ и точку «своего» проживания, сформируйте ссылки на эти точки и пошлите в теле письма на адрес, предоставленный преподавателем.

7. В Картах Google создайте маршрут от точки «своего» проживания до ЮИ МИИТ, отмасштабируйте карту, чтобы маршрут полностью умещался на экране, сделайте скриншот и пошлите в теле письма на адрес, предоставленный преподавателем.

Контрольные вопросы

1. Что такое веб-сервисы?
2. Что такое облачные технологии?
3. Что облачные технологии могут дать организации?
4. Зачем при регистрации рекомендуется указывать номер мобильного телефона?

5. Зачем при регистрации рекомендуется указывать ещё один адрес электронной почты?
6. Нужно ли знать свой MAC-адрес при работе с Google?
7. Что такое «Совместный доступ»?
8. В чем разница в работе с электронной почтой по протоколам POP и IMAP?
9. Что такое аккаунт?
10. Опишите принцип функционирования DNS?

Литература

1. Никифоров, С. В. Введение в сетевые технологии / С. В. Никифоров. – М.: Финансы и статистика, 2007.
2. Таненбаум, Э. Компьютерные сети / Э. Таненбаум. – СПб.: Питер, 2007.
3. Леонтьев, В. Новейшая Энциклопедия ИНТЕРНЕТА 2009 / В. Леонтьев. – М. : ОЛМА Медиа Групп, 2009.
4. Олифер, В.Г. Компьютерные сети. 4-е изд. / В. Г Олифер, Н. А. Олифер. – СПб.: Питер, 2010.

Работа № 5

ИЗУЧЕНИЕ МЕТОДОВ РАБОТЫ С СИСТЕМОЙ ВИЗУАЛЬНОЙ ONLINE МАРШРУТИЗАЦИИ И ДИАГНОСТИКИ

Цель работы: Изучить на практике принципы маршрутизации в сетях с использованием online-сервисов, а также установить промежуточные узлы, обеспечивающие маршрутизацию между заданными хостами, получение детальной информации о промежуточных узлах сетей.

В результате выполнения данной работы студенты должны изучить методы работы с системой визуальной online маршрутизации и диагностики, получить данные по конкретным маршрутам и провести их анализ.

В конце лабораторной работы студенты должны представить отчёт по работе преподавателю и ответить на его вопросы.

Введение

В ОС Windows имеется несколько утилит для диагностики состояния сети, но чаще всего используются Ping и Tracert. Напомним, программа Ping отправляет запрос указанному узлу сети и фиксирует время между отправкой запроса и получением ответа (RTT, от англ. Round Trip Time), иными словами, утилита позволяет определить время отклика интересующего сервера. Чем меньше это время, тем обмен данными с этим сервером производится быстрее. Программа Tracert выполняет отправку тестового пакета указанному узлу сети, отображая информацию обо всех промежуточных маршрутизаторах, через которые прошёл пакет на пути к запрошенному узлу, а также минимальное, максимальное и среднее время отклика каждого из них. Это позволяет оценить, насколько «длинный» путь прошёл пакет и на каком участке возникают наибольшие задержки, связанные с передачей данных.

По мере развития Интернета появились специализированные online-сервисы, которые способны проследить «путешествие» пакетов по сети и по IP-адресу сервера собрать и сообщить о нем различную дополнительную информацию. Подобные сервисы, наряду с расширенной визуализацией представления информации, могут оказаться весьма полезными для быстрого анализа и идентификации источника сетевых проблем.

Одной из таких систем является система Yougetsignal, доступная по адресу <http://Yougetsignal.com>.

Воспользовавшись функцией «Visual Trace Route Tool» на сервисе Yougetsignal.com можно провести трассировку, для чего достаточно ввести URL сервера или его IP-адрес и щёлкнуть на кнопке «Host Trace» (рис. 5.1). В итоге сервис отобразит путь следования пакетов на карте мира (в центре), а также в виде списка промежуточных серверов с указанием общего числа переходов и принадлежности каждого из них конкретной стране (справа). Используя «Proxu Trace» можно провести трассировку от Вашего компьютера через Yougetsignal.com до целевого сервера.

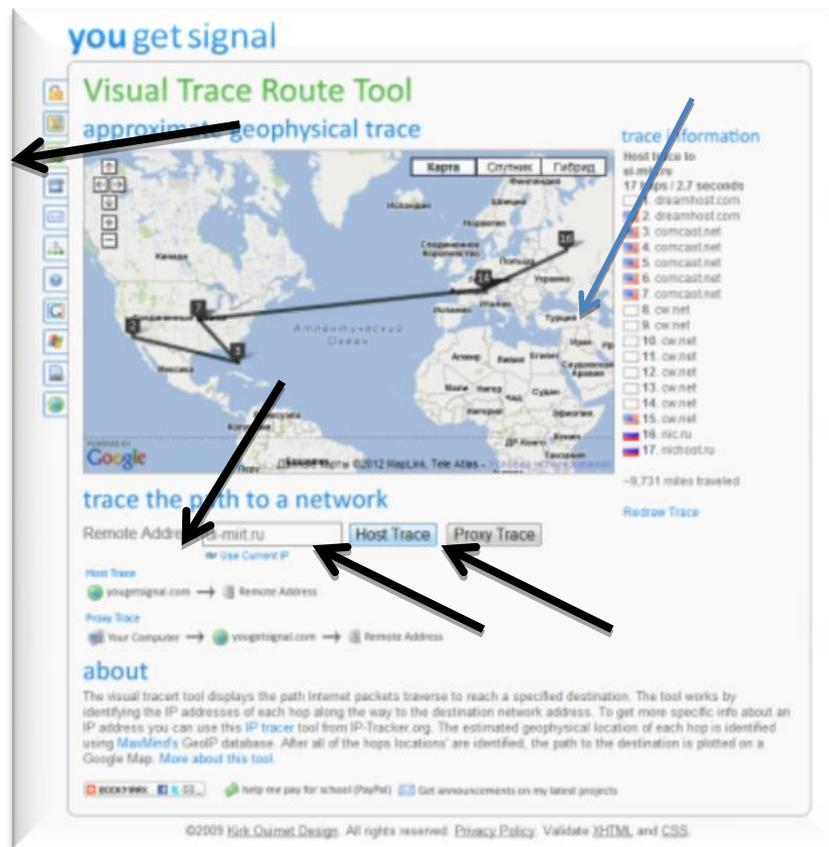


Рис. 5.1. Функция «Visual Trace Route Tool» на сервере Yougetsignal.com

Активировав функцию «Network Location Tool», можно выяснить географическое положение любого сервера по его IP-адресу (рис. 5.2).

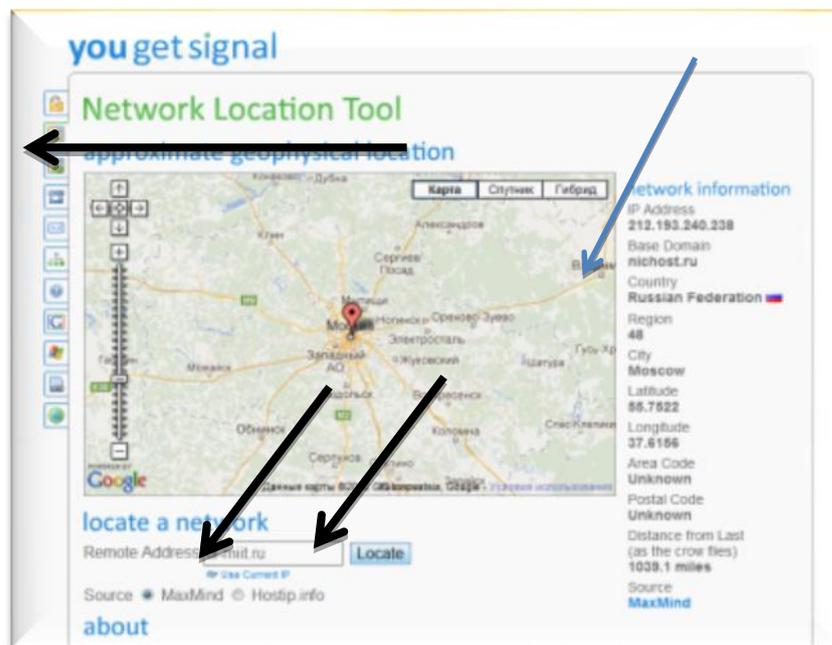


Рис. 5.2. Функция «Network Location Tool»

А воспользовавшись функцией «WHOIS lookup Tool» можно получить информацию о сервере с информационного сервиса WHOIS.

Задания

1. Познакомьтесь с основными видами маршрутизации в сетях.
2. Ознакомьтесь с системой Yougetsignal,
3. Ознакомьтесь с сервисами online данной системы.
4. Сравните предлагаемые сервисы с командами ping и tracer.
5. Проведите маршрутизацию с сервера системы да заданных Вам серверов.
7. Проведите маршрутизацию с Вашего компьютера до заданных Вам серверов.
8. Определите географическое положение серверов..
9. Получите информацию о серверах с информационного сервиса WHOIS.
10. Сохраните отчёт о промежуточных узлах в файл.

Указания по выполнению работы

1. Обратитесь к серверу системы Yougetsignal и ознакомьтесь со следующими ее функциями:
 - Visual Trace Route Tool
 - Network Location Tool
 - WHOIS lookup Tool
2. Проведите трассировку от сервера системы и от Вашего компьютера до серверов, указанных в вариантах выполнения настоящей работы. (*Приложение 2*). Зафиксируйте в файле отчёта:
 - Скриншот трассы
 - Время доступа
 - Число промежуточных маршрутизаторов (хопов)
3. Повторите п. 2 для сервера МПТ.RU
4. Определите географическое положение серверов Вашего варианта по их IP-адресу. Результат зафиксируйте в отчёте
5. Получите информацию о серверах Вашего варианта с информационного сервиса WHOIS. Результат зафиксируйте в отчёте.
6. Сформируйте отчёт по работе в форме *Приложения 1*. Проанализируйте материалы отчёта. Сформулируйте достоинства и недостатки данной системы по сравнению с использованием команд ping и tracer. Вывод занесите в отчёт.

Контрольные вопросы

1. Что такое доменный адрес и как он связан с IP-адресом?
3. Может ли один хост иметь более одного IP-адреса? Обоснуйте свой ответ.

4. Какие знаете домены первого уровня вы знаете, назовите примерное количество таких доменов.
5. Какие серверы Интернета вы знаете?
6. Для чего используется команда `tracert`?
7. Как узнать свой IP-адрес, через который вы выходите в Интернет?
8. Для чего используется команда `ping`?
9. Какие возможности предоставляет функция Visual Trace Route Tool?
10. Какие возможности предоставляет функция WHOIS lookup Tool?

Литература

1. Олифер, В. Г. Компьютерные сети (4-е изд.) / В. Г Олифер, Н. А. Олифер. – СПб. : Питер, 2010.
2. Никифоров, С. В. Введение в сетевые технологии / С. В. Никифоров. – М. : Финансы и статистика, 2007.
3. Леонтьев, В. Новейшая Энциклопедия ИНТЕРНЕТА 2009 / В. Леонтьев. – М. : ОЛМА Медиа Групп, 2009.
4. Глушаков, С. В. Новейшая Энциклопедия работы в Интернете / С. В. Глушаков и др.. – М. : АСТ, 2008.

Работа № 6

ИНФОРМАЦИОННЫЙ ПОИСК В ИНТЕРНЕТЕ

Цель работы: Получить навыки по поиску информации в Интернет с помощью поисковых систем.

В результате выполнения данной работы студенты должны изучить методы поиска и обработки информации, полученной через поисковые системы Интернет.

В конце лабораторной работы студенты должны представить отчёт по работе преподавателю и ответить на его вопросы.

Введение. Поиск информации или информационный поиск представляет один из основных информационных процессов. Интернет-технологии и программно-технические средства, доступные большинству людей, позволяют осуществлять данный процесс в любое время, практически в любом месте по любым запросам.

В процессе поиска в той или иной последовательности производится сопоставление отыскиваемого с каждым объектом, хранящимся в массиве. Цель любого поиска заключается в потребности, необходимости или желании находить различную информацию, способствующую получению нужных сведений, знаний и т.д. для повышения собственного профессионального, культурного и любого иного уровня; создания новой информации и формирования новых знаний; принятия управленческих решений и т.п.

Термин «информационный поиск» (англ. «information retrieval») ввёл американский математик К. Муэрс. Он заметил, что побудительной причиной такого поиска является информационная потребность, выраженная в форме информационного запроса. К объектам информационного поиска К. Муэрс отнес документы, сведения об их наличии и (или) местонахождении, фактографическую информацию. С точки зрения использования компьютерной техники «информационный поиск» – совокупность логических и технических операций, имеющих конечной целью нахождение документов, сведений о них, фактов, данных, релевантных запросу потребителя.

«Релевантность» – устанавливаемое при информационном поиске соответствие содержания документа информационному запросу или поискового образа документа поисковому предписанию. Системы, обеспечивающие реализацию подобного поиска информации, называются поисковыми системами (ПС). В традиционных технологиях ПС представляют картотеки и каталоги, адресные и иные справочники, указатели, энциклопедии, справочный аппарат к изданиям и другие материалы. «Поисковые системы» осуществляют поиск среди документов базы или иных массивов машиночитаемых данных, содержащих заданные слова.

Терминологически «информационно-поисковая система» (англ. «information retrieval system», IRS) – представляет систему, предназначенную для поиска и хранения информации; пакет программного обеспечения, реализующий процессы

создания, актуализации, хранения и поиска в информационных базах и банках данных.

Информационно-поисковая система трактуется и как система, обеспечивающая поиск и отбор необходимых данных на основе информационно-поискового языка и соответствующих правил поиска, а база данных – как совокупность средств и методов описания, хранения и манипулирования данными, облегчающих сбор, накопление и обработку больших информационных массивов. Организация различных БД отличается видом объектов данных и отношений между ними.

Функционирование современных ИПС основано на двух предположениях:

- 1) документы, необходимые пользователю, объединены наличием некоторого признака или комбинации признаков;
- 2) пользователь способен указать этот признак.

Оба эти предположения на практике не выполняются, и можно говорить только о вероятности их выполнения. Поэтому процесс поиска информации обычно представляет собой последовательность шагов, приводящих при посредстве системы к некоторому результату и позволяющих оценить его полноту.

Оценить адекватность выражения запроса, как и полноту получаемого результата, пользователь может, отыскав дополнительные сведения или так организовав процесс, чтобы часть результатов поиска могла использоваться для подтверждения или отрицания адекватности другой части. В то же время, для пользователей-профессионалов характерна устойчивость тематического профиля. Когда они являются «информационно-ориентированными», то им свойственно желание и способность организовать информационное пространство проблемы. Это означает, что пользователь создаёт по существу новый, «самостоятельный» проблемно ориентированный, индивидуально обновляемый и пополняемый ИР, включающий помимо подборок документов также и метаинформацию, например, словари специальной терминологии, классификаторы предметных областей, описания ресурсов и т.д.

Информационный поиск подразумевает использование определённых стратегий, методов, механизмов и средств. Поведение пользователя, осуществляющего управление процессом поиска, определяется не только информационной потребностью, но и инструментальным разнообразием системы – технологиями и средствами, предоставляемыми системой.

Стратегия поиска - общий план (концепция, предпочтение, установка) поведения системы или пользователя для выражения и удовлетворения информационной потребности пользователя, обусловленный как характером цели и видом поиска, так и системными «стратегическими» решениями – архитектурой БД, методами и средствами поиска.

Методы поиска, т.е. выделение подмножества документов, потенциально содержащих описание решения задачи отбора документов (ОД), являются отражением процесса нахождения решения и зависят от характера задачи и предметной области.

Процесс поиска можно представить в виде следующих шагов:

- 1) формулирование запроса на естественном языке, выбор поисковых системы и сервисов, формализация запроса;
- 2) проведение поиска в одной или нескольких поисковых системах;
- 3) обзор полученных результатов (ссылок);
- 4) предварительная обработка полученных результатов: просмотр содержания ссылок, извлечение и сохранение релевантных данных;
- 5) при необходимости модификация запроса и проведение повторного (уточняющего) поиска с последующей обработкой полученных результатов.

Для уменьшения объёма отобранных материалов осуществляют фильтрацию результатов поиска по типу источников (сайтов, порталов), тематике и другим основаниям.

Технологии поиска информации

Поисковые средства и технологии, используемые для реализации информационных потребностей, определяются типом и состоянием решаемой пользователем задачи основной деятельности: соотношением его знания и незнания об исследуемом объекте. Кроме того, процесс взаимодействия пользователя с системой определяется уровнем знания пользователем содержания ресурса (полноты представления, достоверности источника и т.д.) и функциональных возможностей системы как инструмента. В целом эти факторы обычно сводятся к понятию «профессионализма» – информационного (подготовленный/неподготовленный пользователь) и предметного (профессионал/непрофессионал) «профессионализма».

Процесс поиска информации обычно носит эмпирический характер. Он представляет последовательность шагов, приводящих при посредстве системы к некоторому результату, позволяющих оценить его полноту.

Реализация поиска

Что обычно ищут в Интернете: персональные данные об индивидуумах и организациях; различные адресные данные; конкретные материалы (статьи, книги, фотографии, справочные данные, программное обеспечение и др.) в том числе место их хранения; где и сколько стоят те или иные материалы, услуги, продукты и т.п.; информационные сайты и порталы и др.

Общепринята организация поиска по начальным фрагментам слова (поиск с усечением справа), например, вместо слова *библиотечный* можно ввести его фрагмент *библиоте**. При этом будут найдены документы, в которых содержится не только слово *библиотечный*, но и *библиотека*, *библиотекарь*, *библиотековедение* и др. В каждом случае пользователь должен представлять, что именно он хочет найти, так как в предложенном ему варианте будет найдено гораздо большее количество документов, чем при задании данного слова полностью (без усечения). В подобном случае возможно в полученном массиве информации провести уточняющий поиск и в результате получить более релевантные и пертинентные данные.

Критерии оценки поиска

Критерием результата поиска является получение пользователем списка документов, одного документа или их частей, максимально удовлетворяющего его потребностям, сформулированным в поисковом запросе. В ИПС принято форми-

ровать список полученных в результате поиска документов по их релевантности. Различают критерии смыслового и формального соответствия между поисковым предписанием и выдаваемым документом.

Полнота и точность поиска являются взаимосвязанными показателями. Увеличение одного из них ведёт к снижению другого. В современных ИПС при сбалансированном поиске их значения составляют примерно 70%. Следует учитывать ситуацию, при которой список выданных поисковой системой ссылок содержит несколько, а порой и десятки разных адресов с одним и тем же текстом. Подобные ссылки характеризуются как дубликаты. Из них при подсчёте коэффициентов учитывается только один документ.

Оценка и обработка результатов поиска

Учитывая, что идеальный результат поиска должен удовлетворять требованиям единственности, полноты и непротиворечивости, получаем, что различные виды поиска определяют различные требования к функциональным возможностям системы в части оценивания результата. Однако для случая предметного поиска доказательство полноты является тривиальным: непустой результат поиска подтверждает факт существования (или отсутствия) объекта, обладающего искомыми свойствами. При этом результат тематического поиска множественен и требует последующей систематизации – ещё одного процедурного шага для упорядочения полученного множества объектов по значениям не определённого явно основания. В свою очередь проблемный поиск предполагает уже двухуровневую систематизацию.

Интернет-поисковые системы

Для получения информации в среде Интернета создаются специальные поисковые системы. Как правило, они общедоступны и обслуживают пользователей в любой точке планеты, где имеется возможность работы с Интернетом. Непосредственно для поиска используются поисковые машины, число которых в мире исчисляется несколькими сотнями. Они ориентируются на определенные типы запросов или их сочетание (библиографический, адресный, фактографический, тематический и др.). Кроме того, бывают полнотекстовые, смешанные и другие поисковые машины.

Технология поиска (англ. Search Technology) означает совокупность правил и процедур, в результате выполнения которых пользователь получает искомый результат. При поиске в Интернете рекомендуется обращать внимание на две составляющие: полноту (ничего не потеряно) и точность (не найдено ничего лишнего). Обычно соответствие найденных материалам этим критериям называют релевантностью, т.е. соответствием ответа вопросу (запросу).

Каждая поисковая машина имеет свои алгоритмы сортировки результатов поиска. Чем ближе к началу списка, полученного в результате проведения поиска, оказывается нужный документ, тем выше релевантность и лучше работает поисковая машина.

Поисковые машины используют общие принципы работы, ориентированные на выполнение двух основных функций. Первая функция реализуется программой-роботом, автоматически просматривающей различные серверы в Интер-

нете. Находя новые или изменившиеся документы, она осуществляет их индексацию и передаёт на базовый компьютер поисковой машины. *Робот* – автоматизированный браузер, загружающий веб-страницу, изучающий её и, при необходимости, переходящий к одной из её гиперсвязей. Когда ему попадается страница, не содержащая связей, робот возвращается на одну-две ступени назад и переходит по адресу, указанному в одной из обнаруженных ранее связей. Запущенный робот проходит огромные расстояния в среде Интернета (киберпространстве), ориентируясь на развитие веб-сети и изменяя в соответствии с этим свои маршруты. Индексирующие роботы обрабатывают лишь HTML-файлы, игнорируя изображения и другие мультимедийные файлы. Они могут: обнаруживать связи с уже несуществующими страницами; устанавливать связь с наиболее популярными узлами, подсчитывая количество ссылок на них в других веб-страницах; регистрировать веб-страницы для оценки увеличения системы и др. Чаще всего роботы просматривают серверы самостоятельно, находя новые внешние ссылки в уже обследованных документах. Вторая функция заключается в обработке выявленных документов. При этом учитывается все содержание страниц (не только полный текст, но и наличие иллюстраций, аудио- и видеофайлов, Java-приложений). Индексации подвергаются все слова в документе, что дает возможность использовать поисковые системы для детального поиска по самой узкой тематике. Образующие гигантские индексные файлы, хранящие информацию о том, какое слово, сколько раз, в каком документе и на каком сервере употребляется, составляют БД, к которой собственно и обращаются пользователи, вводя в поисковую строку ПОЗ (сочетания ключевых слов). Выдача результатов осуществляется с помощью специальной подсистемы, производящей интеллектуальное ранжирование результатов. В своих расчетах она опирается на местоположение термина, частоту его повторения в тексте, процентное соотношение данного термина с остальным текстом на данной странице и другие параметры, характеризующие возможности конкретной поисковой машины.

Роботы имеют ряд разновидностей, одной из которых является *паук* (англ. *spider*). Он непрерывно *ползает по сети*, переходя с одной веб-страницы к другой с целью сбора статистических данных о самой *паутине* (Web) и (или) формирования некоторой БД с индексами содержимого веб.

Автоматизированные агенты *спайдеры* регулярно сканируют веб-страницы и актуализируют БД адресов (гиперссылки), средства индексирования информации, расположенные по указанным адресам. Полученные индексы используются для быстрого и эффективного поиска по набору терминов, задаваемых пользователем.

Все чаще применяются *интеллектуальные агенты* - небольшие программы, обладающие способностью самообучаться, и действовать самостоятельно от имени своего владельца. Имея связь с компьютером пользователя, они выступают в роли персональных помощников, выполняющих ряд задач с применением знаний о потребностях и интересах пользователя. Интеллектуальные роботы-агенты ведут самостоятельный поиск в сети по собственным уникальным алгоритмам. Некоторые из них не только просматривают ключевые слова, но и осуществляют в

Интернете семантический анализ информации, выявляя степень ее смыслового соответствия поставленной задаче.

Эффективный доступ к информации в Интернете обеспечивают такие зарубежные поисковые системы (машины), как Альта-Виста (AltaVista), Lycos, Yahoo, Google, OpenText, Wais, WebCrawler и др. Их адреса в Интернете: www.altavista.com; www.yahoo.com, www.google.com; www.opentext.com.

К отечественным поисковым машинам относятся: Апорт (Aport АО Агама), Rambler (фирма Stack Ltd.), Яндекс (Yandex фирма CompTek Int), Русская машина поиска, Новый русский поиск, и др. Их адреса в Интернете: www.aport.ru; www.rambler.ru; www.yandex.ru; search.interrussia.com; www.openweb.ru соответственно и др.

Все эти поисковые машины позволяют по ключевым словам, тематическим рубрикам и даже отдельным буквам оперативно находить в сети, например, все или почти все тексты, где эти слова присутствуют. При этом пользователю сообщаются адреса сайтов, где найденные IP постоянно присутствуют. Однако ни одна из них не имеет подавляющих преимуществ перед другими. Для проведения надёжного поиска по сложным запросам специалисты рекомендуют использовать последовательно или параллельно (одновременно) различные ИПС.

Полнотекстовая поисковая машина индексирует все слова видимого пользователю текста. Наличие морфологии дает возможность находить искомые слова во всех склонениях или спряжениях. Кроме этого, в языке HTML существуют теги, которые также могут обрабатываться поисковой машиной (заголовки, ссылки, подписи к картинкам и т.д.). Некоторые машины умеют искать словосочетания или слова на заданном расстоянии, что часто бывает важно для получения разумного результата.

Полноту и точность ответа пользователь получает в зависимости от точности сформулированного им запроса. В результате поиска ему обычно предоставляется гораздо больше информации, чем ему необходимо, часть которой может вообще не иметь отношения к сформированному запросу. Много зависит не только от грамотно сформулированного запроса, но и от возможностей поисковых систем, которые весьма различны. При этом достаточно ярко проявляется *лесной синдром* (из-за леса не видно дров), заключающийся в том, что в полученных данных можно пропустить главные, необходимые сведения. Это наглядно проявляется при формировании простых запросов. Простые запросы в виде отдельных достаточно распространённых терминов приводят к извлечению тысяч (сотен тысяч) документов, абсолютное большинство которых пользователю не требуется (информационный шум).

Порядок оформления работы

1. Откройте редактор Word (если работа в Windows) и создайте новый документ с именем файла – «Лабораторная работа №...» (*Приложение 1*) – файл отчёта по данной лабораторной работе. Создайте в папке Вашей группы новый каталог с именем – «Ваша фамилия». Сохраните файл отчёта с именем «Лабораторная работа №.....» в этой папке.

2. В начале документа создайте титульный лист для отчёта по лабораторной работе в стандартной форме. Сверните (но не закрывайте окно редактора Word!) (*Приложение 1*)

3. Запустите программу Internet Explorer в Windows.

Таким образом, у Вас одновременно должны быть запущены три программы (три окна) – эта методичка, редактор с отчётом и браузер для поиска информации по теме лабораторной работы.

РЕКОМЕНДАЦИЯ: При выполнении лабораторной работы не задерживайтесь долго на одном задании, если Вы не можете ответить на все вопросы задания, внесите в отчёт полученные результаты и переходите к следующему. Потом вернётесь и дополните ответ.

Указания по выполнению работы

1. Найдите и зафиксируйте в отчёте названия и адреса трёх русскоязычных поисковых систем.

2. Сформулируйте пять терминов по вашей специальности и с помощью трёх поисковых систем (п.1) найдите, как часто они встречаются в поисковых запросах каждой из пяти поисковых систем.

Сравните, как на один и тот же запрос на поиск реагируют разные поисковые системы. Какая из них, по вашему мнению, эффективнее. Анализ эффективности поместите в отчёт (текст не более 0,5 страницы).

3. Зайдите в поисковую систему Yandex или другую систему. Найдите перечень сайтов организаций по вашей специальности (архивы, таможенные посты, суды и т.д.) Проведите анализ сайта и его характеристику поместите в отчёт (текст не более 1 страницы).

4. Зайдите в поисковую систему Yandex или другую систему. Найдите сайт, посвящённый электронным библиотекам, и копию сайта зафиксируйте в отчете. Проанализируйте сайт одной конкретной библиотеки. Расположение библиотеки, объем ее фондов контактные реквизиты поместите в отчёт.

5. Зайдите в поисковую систему и найдите материал, который поможет ответить вам на следующие вопросы:

– Где и когда родился известный деятель XX в. (на ваш выбор)? В каком учебном заведении и на каком факультете он обучался?

– Перечислите основные этапы жизни и деятельности. Почему его имя вошло в историю?

– Найдите и сохраните в отчёте его фотографию.

Всю эту информацию (текст не более 1 страницы) поместите в отчёт.

6. Зайдите в поисковую систему. Найдите официальный сайт высшего учебного заведения одного из следующих городов: Пенза, Томск, Омск, Пермь, Владивосток, Белгород, Екатеринбург, Уфа и т.д.. Ответьте на следующие вопросы:

– Кто сегодня является ректором университета?

– Найдите страничку какого-либо факультета.

– Найдите список кафедр факультета.

– Найдите информацию для абитуриентов?

Всю эту информацию (текст не более 1 страницы) поместите в отчёт

7. Зайдите в поисковую систему.

Найдите ответ на вопрос:

–Какие ближайшие три рейса аэрофлота в Лондон, Париж и Владивосток?

– Какие ближайшие три поезда в Калининград, Петрозаводск и Владивосток?

Всю эту информацию (текст не более 1 страницы) поместите в отчёт.

8. Найдите в Интернете информацию о вузах, обучающих по вашей специальности в ЮИ МИИТа. Составьте список вузов и поместите в таблицу –
Название вуза, город.

Всю эту информацию (текст не более 1 страницы) поместите в отчёт.

РЕКОМЕНДАЦИЯ: Не задерживайтесь долго на одном задании, если Вы не можете ответить на все вопросы задания, внесите в отчёт полученные результаты и переходите к следующему. Потом вернетесь и дополните ответ.

Сохраните на диске в папке с вашей фамилией файл отчёта.

Сообщите преподавателю об окончании работы.

Контрольные вопросы:

1. Какие поисковые системы вы знаете?
3. Что такое релевантность?
4. Назовите домены первого уровня и их примерное количество.
5. Какие серверы Интернета вы знаете?
6. Что такое стратегия поиска?
7. На каких двух предположениях основано функционирование современных ИПС?
8. Как связаны полнота и точность поиска?
9. Являются ли полнота и точность поиска взаимосвязанными показателями?
10. Какова роль «роботов» в поисковых системах Интернета?

Литература/источники:

1. Олифер, В. Г. Компьютерные сети (4-е изд.) / В. Г Олифер, Н. А. Олифер. – СПб. : Питер, 2010.
2. Леонтьев, В. Новейшая Энциклопедия ИНТЕРНЕТА 2009 / В. Леонтьев. – М. : ОЛМА Медиа Групп, 2009.
3. Глушаков, С. В. Новейшая Энциклопедия работы в Интернете / С. В. Глушаков и др.. – М. : АСТ, 2008.
4. Жёлтые страницы INTERNET, электронный ресурс, доступ:
<http://yp.piter.com/>

Работа № 7

ИНФОРМАЦИОННЫЙ ПОИСК ПРОФИЛЬНОЙ ИНФОРМАЦИИ В ИНТЕРНЕТЕ

Цель работы: Получить навыки по поиску информации по отрасли/профессии в Интернете с помощью поисковых систем.

В результате выполнения данной работы студенты должны изучить методы поиска и обработки информации, полученной через поисковые системы Интернет, по своей будущей специальности.

В конце лабораторной работы студенты должны представить отчёт преподавателю и ответить на его вопросы.

Порядок оформления работы

1. Откройте редактор MS Word и создайте новый документ с именем файла – «ФИО-Лабораторная работа №...» (*Приложение 1 к работе б*) - файл отчёта по данной лабораторной работе. Сохраните файл отчёта в папке с номером вашей группы.

2. Запустите программу Internet Explorer в Windows.

ПРИМЕЧАНИЕ. У Вас одновременно должны быть запущены три программы (открыты три окна) – методичка, редактор с отчётом и браузер для поиска информации по теме лабораторной работы.

Указания по выполнению работы.

При выполнении данной работы используйте навыки, приобретённые в работе № 6

1. Используя любую поисковую систему, найдите сайт головной организации по Вашему профилю:

- Для таможни – ФТС России
- Для ДОУ – Росархив

Отразите в отчёте электронный адрес, копию первой страницы сайта и почтовый адрес агентства или службы.

1. Найдите и перенесите в отчёт данные о руководителе.
2. Найдите и перенесите в отчёт перечень подчинённых организаций.
3. Найдите пять региональных структур, отразите в отчёте адреса сайтов и первые страницы.

4. Проанализируйте первые страницы этих сайтов. Отразите в отчёте перечни

- a. одинаковых (по названию или содержанию) разделов сайтов;
- b. неповторяющихся разделов;
- c. часто встречающихся разделов;
- d. редко встречающихся разделов;

5. Проанализируйте все пять сайтов. Отразите в отчёте достоинства и недостатки каждого из них.

6. Найдите информацию о международных контактах вашего профильного агентства. Отрадите в отчёте (не более 0,5 страницы).

7. Найдите информацию об аналогичных организациях в царской России. Отрадите наиболее интересную информацию в отчёте (не более 0,5 страницы).

Сообщите преподавателю об окончании работы.

Контрольные вопросы:

1. Какие поисковые системы Вы знаете?
3. На каких языках реализован сайт головной организации?
4. Какова статистика посещений сайта головной организации?
5. Когда размещена на сайте последняя новость?
6. Каковы основные направления деятельности вашей организации?
7. Как используется сайт для работы с населением?
8. Можно ли на сайте найти документы организации по ее реквизитам и как?
9. Можно ли на сайте подписаться на новости и как?
10. Какова роль сайта в профильных организациях?

Литература/источники:

1. Леонтьев, В. Новейшая Энциклопедия ИНТЕРНЕТА 2009 / В. Леонтьев. – М. : ОЛМА Медиа Групп, 2009.
2. Глушаков, С. В. Новейшая Энциклопедия работы в Интернете / С. В. Глушаков и др.. – М. : АСТ, 2008.
3. Жёлтые страницы INTERNET ,электронный ресурс, доступ:
<http://yp.piter.com/>

Работа № 8

ИЗУЧЕНИЕ ОСНОВ ЯЗЫКА ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ ДОКУМЕНТОВ - HTML

Цель работы: Изучить основы и получить навыки подготовки информации для размещения ее в Интернете.

В результате выполнения данной работы студенты должны изучить на практике способы подготовки информации для размещения ее в Интернет.

В конце лабораторной работы студенты должны представить отчёт по работе преподавателю и ответить на его вопросы.

Предварительные указания по порядку выполнения работы

1. Изучите теоретическую вводную часть.
2. В приложении к сборнику лабораторных работ находится справочник по тэгам языка HTML
3. Отчётом по данной работе является страница (файл) HTML, имя которого является фамилией студента с инициалами.
4. При выполнении работы создание и редактирование текста выполняется при помощи редактора Блокнот (Notepad).
5. Просмотр страниц выполняется из браузера Internet Explorer.

Введение в язык HTML. HTML-документы представляют собой ASCII-файлы, доступные для просмотра и редактирования в любом редакторе текстов.

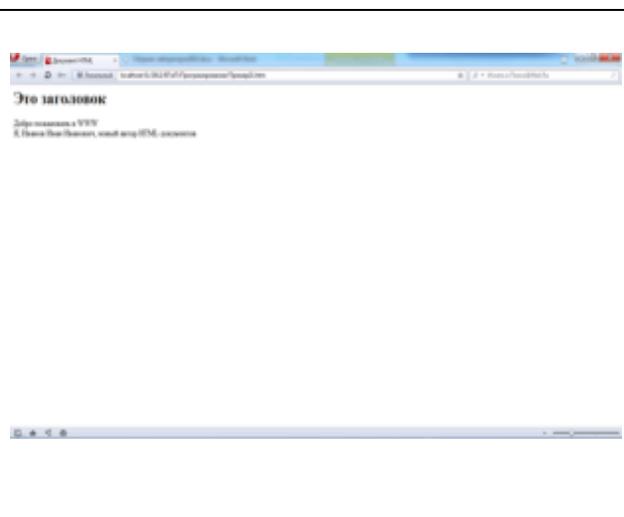
Правила форматирования документа в HTML определяются при помощи специальных команд – тегов. Все теги имеют схожий вид. Начинаются они с имени тега в угловых скобках, а заканчиваются именем тега с косой чертой также в угловых скобках. Например, чтобы указать, что текст должен располагаться в середине экрана, используется следующая команда:

```
<center> Привет посетителям моей страницы. </center>
```

Минимальный HTML-документ

Простой документ на языке HTML и его отражение в браузере представлены ниже:

```
<HTML>
<HEAD>
<TITLE>Документ HTML </TITLE>
</HEAD>
<BODY>
<H1> Это заголовок </H1>
Добро пожаловать в WWW <br> Я,
Иванов Иван Иванович, новый автор
HTML-документов
</BODY>
</HTML>
```



Здесь использованы следующие теги:

<HTML> и парный ему </HTML> для описания всего документа (его границ)

<HEAD> и парный ему </HEAD> – информационный блок

<TITLE> и парный ему </TITLE> для задания заголовка документа

<BODY> и парный ему </BODY> для описания тела документа

<H1> и парный ему </H1> для задания текстового заголовка первого уровня

 для указания на конец параграфа. Внимание, тег
 не имеет парного тега.

ПРИМЕЧАНИЕ. В тегах языка HTML не различаются символы верхнего и нижнего регистров: тег <TITLE> эквивалентен тегу <title><Title>.

Основные теги языка HTML.

Полная справочная таблица тегов HTML находится в приложении к сборнику лабораторных работ.

Заголовок документа

Тэг <TITLE> и парный ему тег </TITLE>

Каждый HTML-документ должен иметь заголовок. Заголовок используется для поиска документов и обычно отображается отдельно от документа, например, как заголовок окна, в котором отображается сам документ. Обычно заголовок содержит краткое описание содержимого документа.

ПРИМЕЧАНИЕ. Заголовок не является частью документа.

Заголовки разделов внутри документа. Для структуризации текста в документе HTML возможно использование заголовков шести уровней – от 1 до 6. Заголовок первого уровня считается самым главным (крупным) и используется в качестве первого заголовка документа. Заголовки могут выделяться цветом или шрифтом.

Синтаксис:

<Hx>Текст заголовка</Hx>,

где x – это число от 1 до 6, указывающее уровень заголовка.

Пример использования заголовков различных уровней.

<HTML>

<HEAD>

<TITLE>Пример использования тегов текстовых заголовков</TITLE>

</HEAD>

<BODY BGCOLOR=SILVER>

<P ALIGN=CENTER>

Заголовки различных уровней!

</P>

<H1> Заголовок первого уровня </H1>

.....

<H6> Заголовок шестого уровня </H6>
</BODY>
</HTML>

Логические стили

Логические стили, наряду с шестью стилями заголовков используются для формирования текстовой части документа. Начертание текста выделенного с помощью логических стилей зависит от настроек браузера.

 –служит для выделения текста (обычно это утолщённый шрифт).

<SMALL> – уменьшенный шрифт.

<BIG> – увеличенный шрифт.

<SUP> (superscript) – для верхних индексов.

<SUB> (subscript) – для нижних индексов.

Физические стили

Начертание шрифта при использовании таких стилей не зависит от настроек браузера.

<U> – (underline) подчёркивание

 – (bold) – полужирный

<I> – (italic) курсив

Чтобы ввести в текст символ произвольного вида, можно задать его код в следующем формате **&#n**, где n ASCII – код вводимого символа.

Чтобы ввести в текст пробел надо написать ** **.

Размер, гарнитура и цвет шрифта

Тег <BAZEFONT SIZE=...> задаёт размер шрифта, который используется в документе по умолчанию

Тег задаёт размер, гарнитуру и цвет символов

Пример

COLOR=«#ffaa55» – шестнадцатеричное значение

ПРИМЕЧАНИЕ. Следует учесть, что шрифты, которые вы используете должны быть установлены не только у Вас, но и у клиентов, которые будут просматривать Ваш документ.

Можно задать несколько допустимых шрифтов, например: . Браузер при этом будет использовать первый из найденных шрифтов.

Теги абзаца и перевода строки

Тег перевода строки
 отделяет строку от последующего текста или графики.

Тег абзаца <P> тоже отделяет строку, но ещё добавляет пустую строку, которая зрительно выделяет абзац. Оба тега являются непарными, т.е. не имеют закрывающих тегов. Чтобы закончить строку или абзац в определённом месте, там надо поставить соответствующий тэг.

Тэг <NoBR> предотвращает произвольный разрыв текста и перемещения графики при переполнении строки окна браузера. Этот тэг парный. Если же в неразбиваемом блоке текста надо в каком-то месте допустить разрыв используется тэг <WBR>. Он ставится между тегами <NoBR> и </NoBR>

В качестве аргумента может применяться одно из трёх ключевых слов: RIGHT, LEFT, ALL, обеспечивающих размещение текста там, где свободно левое, правое или оба поля соответственно (не надо многократно писать
).

Выравнивание текста и графики. Теги <LEFT>, <RIGHT>, <CENTER> выравнивают текст и графику по левому краю, правому краю или центру, соответственно.

Более современные браузеры для выравнивания текста используют директиву ALIGN в теге <P>, которая может принимать значения left, right, center, justify.

Пример

<P ALIGN=left> – выключка по левому краю </P>

<P ALIGN=right> – выключка по правому краю </P>

<P ALIGN=center> – центрирование </P>

<P ALIGN=justify> – выравнивание текста </P>

ПРИМЕЧАНИЕ.

Чтобы поместить в текст документа комментарий – пояснительный текст, который не должен отображаться на экране, следует использовать следующий тег:

<!-- -->

Текст комментария располагается вместо точек.

Разделительные полосы. Тег <HR > (horizontal rule –горизонтальная полоса) разбивает текст (аналогично тегу <P>) и проводит горизонтальную полосу шириной во весь экран.

Вид стандартных полос можно изменить с помощью необязательных атрибутов: SIZE – толщина полосы, WIDTH –длина полосы, ALIGN – выравнивание полосы. Атрибут NOSHADE создаёт сплошную чёрную полосу без оттенения.

Пример

<HR SIZE=10 WIDTH=200 ALIGN=CENTER NOSHADE>

Цвета фона, текста и ссылок определяются в начале файла HTML в теге BODY:

<BODY BGCOLOR=«#aaefff» TEXT= «black» LINK =«#ff0000» VLINK=«red» BACKGROUND = "recbg.jpg" BGPROPERTIES="fixed" >

Атрибуты тега:

BGCOLOR – цвет фона,

TEXT – цвет основного текста,

LINK – цвет не просмотренных гипертекстовых ссылок,

VLINK – цвет просмотренных ссылок,

BACKGROUND – имя графического файла, используемого в качестве фона.

Фоновое изображение – это графический файл (расширение GIF или JPG) с изображением небольшой прямоугольной области. При просмотре в браузере эти

изображения многократно повторяются, заполняя все окно. Чем меньше размер изображения, тем быстрее будет загружаться весь фон.

Цвета ссылок можно определить в теге стилей <style> (это так называемые каскадные стили). При этом значения атрибутов, определенных в тэге <style> имеют приоритет перед определениями, сделанными в теге BODY или в теле документа.

Пример

```
<head>
<style>
a:link {color:#ff5500; font-weight:normal; }
a:visited {color:#00ff00; font-weight:normal; }
a:hover {color:blue; font-weight:normal; }
</style>
</head>
```

Фоновый звук. Internet Explorer распознает оба формата оцифрованного звука (WAV, AU), а также формат MIDI. Большие файлы требуют много времени для передачи и загрузки.

Тег <BGSOUND SRC=«file.wav» LOOP=10> обеспечивает загрузку и воспроизведение на соответствующем аппаратном обеспечении,

где:

- SRC – это сокращение от Source (источник);
- LOOP определяет число повторений звукового клипа. Чтобы звук повторялся бесконечно, надо задать ключевое слово INFINITE.

Графика. Графические файлы используются в HTML-документах с целью повышения наглядности и привлекательности документов, а также для вставки текстов, например, математических формул, которые трудно воспроизвести при помощи текста.

 – это минимальная форма записи вставки графического файла, image1.jpg – это имя графического файла, вставляемого в документ.

 – это развёрнутая (максимальная) форма.

LOWSRC – необязательный параметр, используемый для увеличения скорости передачи графики. Файл image1-1.jpg содержит то же изображение, что и image1.jpg, но выполнен с низким разрешением. Этот файл загружается вместе с текстом, а затем уже загружается основное изображение.

ALT – задаёт имя файла изображения, которое отображается при наведении курсора на ссылку. Особенно полезно, когда программа по каким-то причинам не находит файла изображения. Этот файл загружается вместе с текстом, а затем уже загружается основное изображение.

WIDTH, HEIGHT – ширина и высота изображения. Использование этих атрибутов существенно ухудшает качество изображения и не обеспечивает более быстрой загрузки изображения, так как фактический размер файла изображения (в

байтах) при этом не изменяется (качество не ухудшится, если точно выдержать соотношение размеров изображения по ширине и высоте).

Атрибут ALIGN принимает значения:

LEFT – обеспечивает выравнивание изображения по левому краю. При этом текст обтекает изображение справа.

RIGHT – обеспечивает выравнивание изображения по правому краю. При этом текст обтекает изображение слева.

TOP – выравнивает одну строку текста (например название рисунка) по верху изображения. При этом текст располагается слева или справа от изображения, в соответствии с расположением текста относительно изображения в исходном тексте. При этом не надо ставить знака
.

BOTTOM – выравнивание строки текста внизу изображения (по высоте).

MIDDLE – выравнивание строки текста по середине изображения (по высоте).

HSPACE – горизонтальное поле (отступ в пикселях) вокруг изображения.

VSPACE – вертикальное поле (отступ в пикселях) вокруг изображения.

BORDER=2 ширина черной рамки вокруг изображения.

ПРИМЕЧАНИЕ. Анимационные GIF-файлы встраиваются в документ точно таким же образом, и не требуют дополнительных параметров.

Указания по выполнению работы

1. Создайте HTML-документ (Web-страницу), содержащую четыре параграфа текста, выполненного различными типами шрифтов, различными цветами, имеющих различные выключки (выравнивания).

ПРИМЕЧАНИЕ. Для создания HTML документа используйте программу «Блокнот» (Notepad).

2. В первом параграфе документа должна быть фраза:

«Работа студента группы Иванова И.И.». Далее наименование вуза, факультета и специальности.

3. Параграфы должны начинаться заголовками различных уровней и содержать до трех строк текста. Заголовок первого параграфа – «Данные об авторе»

4. В HTML-документе (п. 1 работы) создайте фоновый рисунок (тег BODY, атрибут BACKGROUND).

5. Введите в документ между параграфами текста разделительные полосы (тег HR), придав атрибутам полосы различные значения высоты, ширины и выравнивания.

6. Из списка вариантов подготовьте пять графических изображений для вставки в HTML-документ. Измените (при необходимости) размеры изображений в соответствии с особенностями параграфов создаваемого документа и его общей компоновкой. Выполните подрисовочные надписи.

7. Проверьте через браузер правильность выполнения заданий данной работы.

Сообщите преподавателю об окончании работы.

Контрольные вопросы

1. Какова структура HTML-документа?
3. Что такое HTML?
4. Сколько размеров заголовков можно использовать в тексте HTML документа?
5. Какими тегами оформляется тело HTML-документа?
6. Как изменить размер графической вставки в HTML-документе?
7. Как ввести в текст символ произвольного вида в коде ASCII?
8. Бывают ли непарные теги?
9. Что такое тег?
10. С помощью какой программы в данной работе ведётся создание и редактирование текста HTML-документа?

Литература/источники

1. Леонтьев, В. Новейшая Энциклопедия ИНТЕРНЕТА 2009 / В. Леонтьев. – М. : ОЛМА Медиа Групп, 2009.
2. Глушаков, С. В. Новейшая Энциклопедия работы в Интернете / С. В. Глушаков и др.. – М. : АСТ, 2008.
3. Жёлтые страницы INTERNET, электронный ресурс, доступ:
<http://yp.piter.com/>
4. Справочник по HTML, электронный ресурс: <http://htmlbook.ru/html>
5. Верстка сайтов, электронный ресурс: <http://htmlcssjs.ru/>

Работа № 9

ИЗУЧЕНИЕ ОСНОВ СОЗДАНИЯ МНОГОСТРАНИЧНЫХ САЙТОВ

Цель работы: Изучить основы и получить навыки подготовки многостраничной информации для размещения ее в Интернет.

В результате выполнения данной работы студенты должны освоить на практике основы подготовки многостраничной информации для размещения ее в Интернете и использования ссылок.

В конце лабораторной работы студенты должны представить отчет по работе преподавателю и ответить на его вопросы.

Предварительные указания по порядку выполнения работы

1. Изучите теоретическую вводную часть к данной работе.
2. При необходимости, используйте вводную часть к работе № 8.
3. При выполнении лабораторной работы воспользуйтесь справочником по тегам языка HTML, расположенным в конце издания.
4. Отчетом по данной работе является страница (файл) HTML, имя которого является фамилией студента с инициалами.
5. При выполнении работы создание и редактирование текста выполняются при помощи редактора Блокнот (Notepad).
6. Просмотр страниц выполняется из браузера Internet Explorer.

Введение. Организация связи документов на сайте (ссылок).

Ссылки позволяют щелчком по выделенному слову или фразе перейти к другому документу или части того же документа. Гипертекстовые ссылки выделяются в тексте документа цветом.

Чтобы включить ссылку в документ необходимо:

1. Поместить в текст тег <A >. После символа A следует пробел.
2. Указать имя файла, на который происходит ссылка в формате: HREF=«имя файла» и закрывающую скобку >.
3. Ввести текст ссылки.
4. Указать на завершение тега

Пример

```
<A HREF=«GLAVA2.HTM»> Глава вторая </A>
```

В данном примере фраза «Глава вторая» является ссылкой на HTML-документ, GLAVA2.HTM, находящийся в том же каталоге (папке), что и текущий документ

Если же документ находится в другом каталоге, то необходимо указать имя этого каталога (или путь), например:

```
<A HREF=«BOOK/GLAVA2.HTM»> Глава вторая </A>
```

Такие ссылки называются относительными. При необходимости можно задать и абсолютные имена файлов. Последние обычно используются для ссылок на несвязанные между собой документы, например:

 Глава вторая
Здесь http://www.cpress.ru – так называемый URL – Uniform Resource Locator (унифицированный адрес ресурсов).

Ссылки на фрагмент документа

Для создания ссылки из документа DOCUM1 на какой-то раздел (абзац) документа, содержащегося в файле DOCUM2, необходимо сначала создать именованную ссылку (метку) в документе DOCUM2:

```
<A NAME=«РЕТЈА»>Петя Иванов</A>
```

а затем ссылку на неё из документа DOCUM1, указывая не только имя файла, но и имя ссылки, разделив их символом #.

```
<A HREF=«DOCUM2#РЕТЈА»>Привет от Петра</A>
```

В этом примере фраза «Привет от Петра» является ссылкой на метку «РЕТЈА» документа DOCUM2.

Для организации ссылок на разделы одного и того же документа используется та же технология, но в адресе ссылки не указывается имя файла. Например, чтобы перейти в именованный раздел РЕТЈА в документе DOCUM2, надо создать следующую ссылку:

```
<A HREF=«#РЕТЈА»>
```

Цвета ссылок можно определить в теге стилей <style> (это так называемые каскадные стили.) При этом значения атрибутов, определенных в тэге <style> имеют приоритет перед определениями, сделанными в теге BODY или в теле документа.

Пример:

```
<head>
```

```
<style>
```

```
a:link {color:#ff5500; font-weight:normal; }
```

```
a:visited {color:#00ff00; font-weight:normal; }
```

```
a:hover {color:blue; font-weight:normal; }
```

```
</style>
```

```
</head>
```

Графические ссылки

Такие ссылки позволяют щёлкнуть на изображении и перейти к другому файлу.

Для того, чтобы сделать графическую ссылку, надо набрать тот же код, что и для текстовых ссылок. Но в том месте, где стоял текст, выделенный для ссылки, поставьте графическое изображение, на котором будет производиться щелчок.

Пример

```
<A HREF=«DOG.HTM»><IMG SRC=«images/dog1.gif» ALT=«Собака»  
WIDTH=60 HEIGHT=80></A>
```

Целесообразно для графических ссылок не уменьшать размер исходного изображения при помощи атрибутов WIDTH и HEIGHT, а создавать отдельные

миниатюрные изображения. Это существенно ускорит процесс загрузки документа.

Указания по выполнению работы

1. Используя коллекцию графики из набора вариантов (см. Приложение к работе № 8), создайте графическое изображение (размер не более 500×500 пикселей) для использования в качестве статичного изображения заднего плана документа.

2. Создайте первую страницу HTML-документа, аналогично предыдущей работе.

3. Создайте три дополнительные страницы с различным цветом фона страницы:

- страница 2;
- страница 3;
- страница 4.

Главному документу присвойте имя INDEX.HTM. При помощи тега Текстовая ссылка организуйте связь главного документа с тремя другими документами, обеспечивая перекрашивание гипертекстовой ссылки при наведении на неё курсора.

4. Перенесите текст параграфа 2 из работы № 8 на страницу 2 и создайте ссылку со слов «Параграф 2» на первой странице на страницу 2.

5. Повторите п. 4 для параграфов 3 и 4 из работы № 8.

6. Снабдите текст комментарием – Это сайт студента из страниц.

7. Проверьте через браузер правильность выполнения заданий данной работы.

Сообщите преподавателю об окончании работы.

Контрольные вопросы

1. Каким тегом определяется ссылка?
3. Для чего служат теги <A> и ?
4. Как задать графическую ссылку?
5. Как создать ссылку для отправки электронной почты?
6. Как задать текстовый комментарий?
7. Как задать статичное изображение заднего плана документа?
8. Как обеспечить перекрашивание гипертекстовой ссылки при наведении на неё курсора?
9. Рекомендуются ли для графических ссылок уменьшать размер исходного изображения при помощи атрибутов WIDTH и HEIGHT?
10. Существует ли ограничение на длину фразы, являющейся гиперссылкой?

Литература/источники

1. Леонтьев, В. Новейшая Энциклопедия ИНТЕРНЕТА 2009 / В. Леонтьев. – М. : ОЛМА Медиа Групп, 2009.
2. Глушаков, С. В. Новейшая Энциклопедия работы в Интернете / С. В. Глушаков и др.. – М. : АСТ, 2008.
3. Жёлтые страницы INTERNET, электронный ресурс, доступ: <http://yp.piter.com/>
4. Справочник по HTML, электронный ресурс: <http://htmlbook.ru/html>
5. Верстка сайтов, электронный ресурс: <http://htmlcssjs.ru/>

Работа № 10

РАЗРАБОТКА ТАБЛИЦ В СОСТАВЕ HTML-ДОКУМЕНТОВ

Цель работы: Изучить основы и получить навыки создания таблиц и структурирования информации HTML документов с их помощью, для размещения в Интернете.

В результате выполнения данной работы студенты должны изучить на практике способы подготовки информации в табличной форме для размещения ее в Интернете.

В конце лабораторной работы студенты должны представить отчёт по работе преподавателю и ответить на его вопросы.

Предварительные указания по порядку выполнения работы

1. Изучите теоретическую вводную часть к данной работе.
2. При необходимости, используйте вводную часть работ №8 и 9..
3. При выполнении лабораторной работы воспользуйтесь справочником по тегам языка HTML, расположенным в конце издания.
4. Отчётом по данной работе является страница (файл) HTML, имя которого является фамилией студента с инициалами.
5. При выполнении работы создание и редактирование текста выполняются при помощи редактора Блокнот (Notepad).
6. Просмотр страниц выполняется из браузера Internet Explorer.

Введение

Таблицы используются для представления информации в виде горизонтальных рядов и вертикальных колонок или для форматирования текста и графики. Каждая клетка таблицы называется ячейкой.

Ячейки могут содержать в себе текст, графику или другую таблицу. Текст и графика внутри ячейки могут представлять собой внешние или внутренние ссылки.

Начинается каждая новая таблица тегом <TABLE>, а заканчивается парным ему тегом </TABLE>. Тег <TABLE> задаёт базовые характеристики таблицы – расстояния между ячейками, ширину таблицы относительно экрана и т. п. Например, следующая пара тэгов:

```
<TABLE ALIGN=CENTER BORDER=2 WIDTH=50% >
```

```
.....
```

```
</TABLE>
```

описывает таблицу, выровненную по центру экрана, занимающую 50 % ширины экрана (атрибут WIDTH=) и имеющую рамку размером в 2 пикселя (атрибут BORDER=). Если атрибут BORDER не задан или равен 0, рамка не отображается.

Ширину таблицы можно задавать в абсолютных значениях (в пикселях). Если ширина не задана, размер таблицы по ширине определяется программно и

зависит от размеров и особенностей форматирования текста (или графики) внутри ячеек.

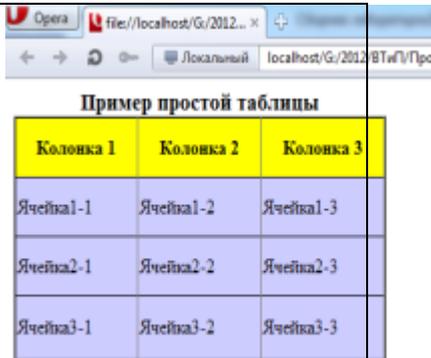
Высота всей таблицы и, соответственно, отдельных ее ячеек может быть определена атрибутом HEIGHT= тега <TABLE>. Если этот атрибут не задан, высота программно устанавливается в зависимости от размера шрифта и особенностей форматирования текста внутри ячеек.

Каждый ряд таблицы описывается парой тегов <TR> и </TR>. Колонки внутри каждого ряда описываются тегами <TD> и </TD>. Содержимое каждой ячейки (это может быть текст, графическое изображение новая таблица, ссылка и т.п.) помещается между тегами <TD> и </TD>.

Тег <CAPTION> внутри тега <TABLE> описывает заголовок таблицы. При помощи атрибута Align заголовок может быть размещён сверху (значение TOP) или внизу таблицы (значение BOTTOM).

Теги <TH> и </TH> внутри тега <TABLE> задают названия столбцов.

Пример простой таблицы

<pre> <table width=350 height=200 border=2 frame=box rules=all align=left bgcolor=#ccccff> <caption align=top><h3> Пример простой таблицы </caption></h3> <TR align=center bgcolor=#ffff00> <TH> Колонка 1 </TH><TH>Колонка 2 </TH> <TH>Колонка 3 </TH> </TR> <tr> <td>Ячейка 1-1</td><td>Ячейка 1- 2</td><td>Ячейка 1-3</td> </tr> <tr> <td>Ячейка 2-1</td><td>Ячейка 2- 2</td><td>Ячейка 2-3</td> </tr> <tr> <td>Ячейка 3-1</td><td>Ячейка 3- 2</td><td>Ячейка 3-3</td> </tr> </table> </pre>	
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------

Определение структуры и цветовое оформление таблиц

Атрибут BGCOLOR тега <TABLE> задаёт цвет фона всей таблицы, а атрибут BGCOLOR= тега <TR> определяет цвет фона текущей строки таблицы. Аналогично в теге <TD> можно задать цвет фона отдельной ячейки таблицы.

Атрибут bordercolor задаёт цвет всей рамки

Атрибут bordercolorlight задаёт цвет светлой стороны рамки

Атрибут bordercolordark задаёт цвет темной стороны рамки

Атрибут cellspacing задаёт расстояние между ячейками таблицы

Атрибут cellpadding задаёт расстояние между рамкой таблицы и ее содержимым

Атрибут frame определяет форму внешней рамки таблицы:

vsides отображаются только вертикальные линии;

hsides отображаются только горизонтальные линии;

box отображается вся рамка.

Атрибут rules определяет форму рамок внутри таблицы:

All отображаются все части рамки внутри таблицы

Rows отображаются горизонтальные части рамки

Cols отображаются вертикальные части рамки.

Пример таблицы, содержащей графику

```
<table height=200 border=2 align=right bgcolor=#ccccff cellspacing=10 cellpadding=5>
```

```
<caption align=top><h3>Пример таблицы с графикой и ссылками  
</caption></h3>
```

```
<TR bgcolor=#ffff00>
```

```
<TH>Колонка 1 </TH><TH>Колонка 2 </TH> <TH>Колонка 3 </TH>
```

```
</TR>
```

```
<TR>
```

```
<td>Ячейка1-1</td><td>&nbsp; </td><td></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Ячейка2-1</td><td>Ячейка2-2</td><td>Ячейка2-3</td>
```

```
</tr>
```

```
<tr>
```

```
<td><a href="test-2.htm"></a></td><td>Ячейка3-2  
</td><td>Ячейка3-3 </td>
```

```
</tr>
```

```
</table>
```

 – описывает изображение car.jpg, отображаемое в ячейке таблицы

Выравнивание в таблицах

Если выравнивание не задавать, то название таблицы и заголовки столбцов центрируются, а содержимое прочих ячеек по горизонтали выравнивается по левому краю, а по вертикали – по средней линии ячейки.

Атрибут Align в теге <TR> обеспечивает горизонтальное и вертикальное выравнивание содержимого всего ряда.

Если нужно сделать горизонтальное и вертикальное выравнивание в отдельных ячейках, это делают с помощью атрибута Align в тегах <TH> и <TD>.

Тэг COLGROUP распространяет действие выравнивания (атрибут Align=) в одной ячейке на весь столбец.

Атрибут объединения Span= распространяет действие тэга <COLGROUP> на несколько столбцов.

Пример

```
<table align=left width=350 height=300 border=6 frame=box rules=all bgcolor=#ccccff
cellspacing=8 bordercolorlight=#ffffff bordercolordark=#666666>
<colgroup align=center span=2>
<colgroup align=right>>
<caption align=left> <h3>Пример простой таблицы </caption></h3>
<TR bgcolor=#ffff00>
<TH> Колонка 1 </TH><TH>Колонка 2 </TH> <TH align=center>Колонка 3
</TH>
</TR >
<tr align=center>
<td align=top>Ячейка1-1 </td><td>Ячейка1-2</td><td>Ячейка1-3</td>
</tr>
<tr>
<td>Ячейка2-1</td><td>Ячейка2-2</td><td>Ячейка2-3</td>
</tr>
<tr>
<td>Ячейка3-1</td><td>Ячейка3-2</td><td>Ячейка3-3</td>
</tr>
</table>
```

Составные ячейки

Несколько ячеек можно объединять в одну как по горизонтали, так и по вертикали.

Атрибут COLSPAN= объединяет несколько ячеек в ряду в одну широкую ячейку.

Атрибут ROWSPAN= объединяет несколько ячеек по высоте в одну высокую ячейку.

Пример 3

```
<table align=center width=300 border=2 bgcolor = ccccff>
<caption><font size=4 color=#aa0000> Пример составной таблицы !</caption>
<tr bgcolor =#ffee00>
<TH width=35% >Заголовок 1</TH><th colspan=2 >Заголовок 2</th>
</tr>
<tr>
<td>Ячейка1-1</td><td>Ячейка1-2</td><td>Ячейка1-3</td>
</tr>
<tr>
```

```

<td>Ячейка2-1<br></td><td colspan=2>Ячейка2-2</td>
</tr>
<tr>
<td>Ячейка3-1</td><td rowspan=2 align=center bgcolor=#ffee33>Ячейка3-
2</td><td>Ячейка3-3</td>
</tr>
<tr>
<td>Ячейка4-1</td><td>Ячейка4-2</td>
</tr>
<tr>
<td>Ячейка5-1</td><td>Ячейка5-2</td><td>Ячейка5-3</td>
</tr>
<tr>
<td colspan=3 align=center bgcolor=#ffee00> Конец составной таблицы !</td>
</tr>
</table>

```

Указания по выполнению работы

1. Создайте простую таблицу, содержащую заголовки столбцов, четыре столбца и пять строк с тремя различными видами оформления рамок таблицы и ячеек
 2. Создайте таблицу, содержащую ячейки с тестом и графическими изображениями.
 3. Разработайте составную таблицу. Включите в неё текстовые и графические ссылки на следующие страницы связанного HTML-документа.
 4. Задайте таблице название, расположите над ней.
 5. Задайте каждой строке таблицы свой цвет фона ячеек.
 6. Объедините в последней строке таблицы ячейки попарно.
 7. Проверьте через браузер правильность выполнения заданий данной работы.
- Сообщите преподавателю об окончании работы.

Контрольные вопросы

1. Какой атрибут задаёт цвет фона всей таблицы?
2. Если ширина таблицы не задана в пикселях, то чем определяется размер таблицы по ширине?
3. Может ли текст внутри ячейки представлять собой внешние или внутренние ссылки?
4. Может ли графика внутри ячейки представлять собой внешние или внутренние ссылки?
5. Можно ли несколько ячеек таблицы объединять в одну по горизонтали?

6. Можно ли несколько ячеек таблицы объединять в одну по вертикали.?
7. Для чего используются таблицы в HTML?
8. Если не задано выравнивание, то как размещается название таблицы и заголовки столбцов?
9. Если не задано выравнивание, то как размещается содержимое прочих ячеек?
10. Что такое составная ячейка?

Литература/источники

4. Леонтьев, В. Новейшая Энциклопедия ИНТЕРНЕТА 2009 / В. Леонтьев. – М. : ОЛМА Медиа Групп, 2009.
5. Глушаков, С. В. Новейшая Энциклопедия работы в Интернете / С. В. Глушаков и др.. – М. : АСТ, 2008.
6. Жёлтые страницы INTERNET, электронный ресурс, доступ:
<http://yp.piter.com/>

Работа № 11

ИЗУЧЕНИЕ РЕКОРДЕРА МАКРОСОВ VBA в MS WORD

Цель работы: Изучить и получить навыки практической работы с рекордером макросов в среде программирования VBA.

В результате выполнения данной работы студенты должны научиться записывать макросы при помощи встроенного рекордера и получить первые навыки работы в среде программирования VBA, а также навыки разработки простейших программ с использованием инструментария данной среды.

В конце лабораторной работы студенты должны представить отчёт по работе преподавателю в виде демонстрации работы тестовой программы.

Указания по порядку выполнения работы

Работа начинается с того, что создаётся документ Microsoft Word, который должен иметь название:

Введение. Пример записи рекордера макросов Сидорова.doc.

Затем необходимо открыть этот документ.

Прежде чем приступить к записи нового макроса, рекомендуется ознакомиться с тем, какие инструменты потребуются для работы с макросами и где они расположены.

Запись нового макроса с использованием макрорекордера, как правило, включает в себя несколько основных этапов: задание стартовых условий, запуск макрорекордера, присвоение ему имени и назначение места хранения, выполнение записываемых действий, остановка макрорекордера и запуск либо выполнение нового макроса. Рассмотрим все этапы подробнее.

Работа начинается с задания для макроса стартовых условий, т.е. условий, которые должны соблюдаться во время воспроизведения данного макроса. Выполнение или воспроизведение макроса означает отработку записанного в этом макросе программного кода.

Пример. Вам необходимо создать макрос, который будет устанавливать определённый формат ячейки либо форматировать текст, применяя к нему заданный шрифт, размер и цвет. Стартовыми условиями для этого макроса будет открытый документ с выделенным блоком текста (если это Word).

Стартовые условия для макроса необходимо задавать до того, как будет запущен макрорекордер, потому что макрорекордер будет записывать абсолютно все действия, которые Вы осуществляете. Если Вы запустите макрорекордер, а только после этого откроете какой-либо документ и выделите текст, то эти действия станут частью макроса и на выходе получится макрос, всегда открывает один и тот же документ и форматирует один и тот же текст.

Запускаем макрорекордер (Меню «Вид»-> Макросы -> ЗаписьМакроса) и присваиваем имя макросу (рис. 11.1).



Рис. 11.1. Запуск записи макроса

Одновременно с запуском макрорекордера Вы должны присвоить имя своему макросу и выбрать место, где он будет сохранен. Сохраняются макросы в специально отведённые для них места – модулях. В свою очередь модули есть как в обычных файлах, так и в личных книгах макросов в Excel, либо глобальных шаблонах в Word.

При запуске макрорекордера Вы можете также назначить для запуска макроса сочетание «горячих клавиш» либо привязать макрос к какому-либо меню или панели инструментов (в зависимости от того, записываете ли Вы макрос в Excel или в Word) (рис.11.2).

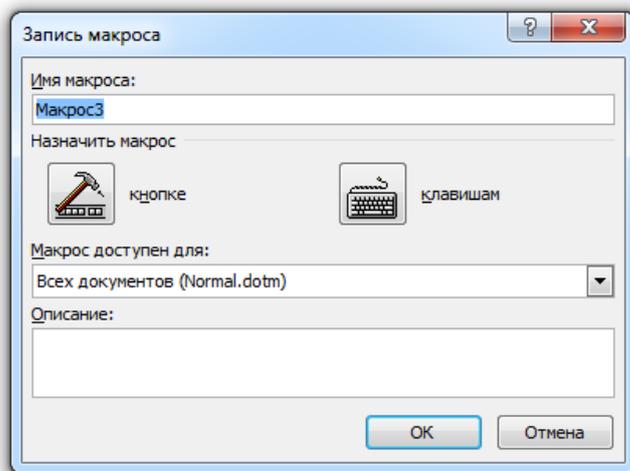


Рис.11.2. Диалоговое окно «Запись макроса»

Имена макросов должны начинаться с буквы, хотя могут содержать и числа. Имена не должны включать в себя пробелы и знаки препинания.

Выполняем действия, которые необходимо записать, чтобы впоследствии эти действия использовать.

Вы можете совершать любые действия, которые можно выполнить при помощи клавиатуры и мыши, включая выполнение уже записанных макросов, все эти действия будут записаны макрорекордером. Выполняемые Вами действия должны зависеть от задачи, которую Вам необходимо выполнить.

Останавливаем макрорекордер. Остановка записи макроса производится нажатием кнопки, показанной на нижеприведённом рисунке, или Меню «Вид»->Макросы -> ОстановитьЗапись).

Когда вы останавливаете макрорекордер, Ваши действия перестают записываться и больше не сохраняются в теле макроса. После остановки макрорекордера новый записанный макрос сразу готов к использованию.

Запускаем новый макрос. После того, как Вы записали рекордером макрос, этот макрос можно выполнить. Для того чтобы запустить макрос, можно использовать также и горячие клавиши Alt+F8. В появившемся диалоговом окне по названию выбирается необходимый макрос и нажимается кнопка «Выполнить». Вы можете также назначить макросу кнопку или графический объект, помещённый непосредственно на рабочий лист Excel либо в документ Word.

С помощью макрорекордера, как правило, можно записать лишь самые простейшие макросы, но уже и это может существенно упростить и облегчить выполнение однообразных, часто повторяющихся действий.

Указания по порядку выполнения работы

1. В MS Word создайте новый файл с именем «ФамилияСтудента.doc» со следующим содержанием (расположенным между --- Начало текста ---- и --- Конец текста ----):

--- Начало текста ----

Данный файл создан Ивановым И.И. (укажите здесь Ваши Ф.И.О.) студентом группы (укажите Вашу группу) для демонстрации работы макрорекордера.

--- Конец текста ----

2. Задайте имя первого макроса в виде «МакросФорматирования» в соответствии с пунктом «**Пример записи рекордера**».

3. Запустите запись макрорекордера.

✓ Выделите Фамилию, для чего мышкой установите курсор на начальную позицию Фамилии и при нажатой клавише Shift и стрелкой → (вправо) выделите текст.

✓ Затем, нажимая на кнопку увеличения шрифта, или сочетанием клавиш [Ctrl+!] увеличьте размер выделенного текста. Учтите, что клавишу «!» получить только при одновременном нажатии клавиш «Shift» и «1».

✓ Выделите номер группы и оформите его курсивом, используя сочетание клавиш [Ctrl+I].

✓ Остановите запись макроса.

✓ Сохраните файл.

4. Задайте имя второго макроса в виде «МакросСбросФорматирования» в соответствии с пунктом «**Пример записи рекордера**».

Запустите запись макрорекордера.

✓ Выделите фамилию, для чего мышкой установите курсор на начальную позицию фамилии и при нажатой клавише Shift и стрелкой → (вправо) выделите текст.

✓ Затем, нажимая на кнопку уменьшения шрифта или используя сочетание клавиш [Ctrl+)], уменьшите размер выделенного текста до исходного. Учтите, что клавишу «)» можно получить только при одновременном нажатии клавиш «Shift» и «0».

✓ Выделите номер группы и снимите курсив, используя сочетание клавиш Ctrl+I.

✓ Остановите запись макроса.

✓ Сохраните файл.

О завершении работы сообщите преподавателю и продемонстрируйте работу Вашей программы.

Контрольные вопросы

1. Как запустить макрорекордер VBA?
3. Как настроить макрорекордер?
4. Какими правилами надо руководствоваться при задании имен макросов?
5. Как запустить проект на исполнение?
6. Как остановить запись макроса?
7. Как сохранить проект, чтобы он был доступен из любого файла?

Литература/источники

1. Программирование на VBA и создание макросов, электронный ресурс: http://www.lessons-tva.info/edu/e-inf2/m2t3_7.html
2. Самоучитель программирования на VBA в MS Office, электронный ресурс: http://www.libray.narod.ru/rapid/vba_programmirovanie_office.htm
3. Стив Камминг «VBA для чайников», электронный ресурс: http://www.proklondike.com/books/vb/vb_Cummings_VBA_Dummies.html

Работа № 12

ИЗУЧЕНИЕ СРЕДЫ ПРОГРАММИРОВАНИЯ VBA в MS WORD

Цель работы: Изучить и получить навыки практической работы в среде программирования VBA.

В результате выполнения данной работы студенты должны получить первые навыки работы в среде программирования VBA и разработки простейших программ с использованием инструментария данной среды.

В конце лабораторной работы студенты должны представить отчёт по работе преподавателю в виде демонстрации работы тестовой программы.

Указания по порядку выполнения работы

Проект, который должен быть создан в данной работе, выводит на экран пользовательскую форму с надписью: «Я программирую в системе VBA». Кроме надписи, на основной пользовательской форме должны присутствовать две экранные кнопки. Одиночный щелчок левой кнопкой мыши на одной из этих экранных кнопок должен выводить в нижней части формы вторую надпись со сведениями об авторе программы. При двойном щелчке мышью на этой же кнопке надпись со сведениями об авторе должна исчезнуть. Вторая экранная кнопка нужна для того, чтобы закрыть основную экранную форму и завершить работу данной программы.

Порядок выполнения:

1. Создайте документ Microsoft Word, который должен иметь название: Первая программа Сидорова.doc.

ПРИМЕЧАНИЕ. Фамилия должна соответствовать фамилии студента.

2. Откройте этот документ и войдите в систему программирования VBA с помощью последовательности команд меню **Сервис** → **Макрос** → **Visual Basic**. Выполнение этой последовательности команд открывает систему программирования VBA (или используйте комбинацию клавиш Alt-F11) (рис. 12.1).

3. В системе VBA создайте экранную форму с помощью команд меню **Insert** → **UserForm** (рис.12.2).

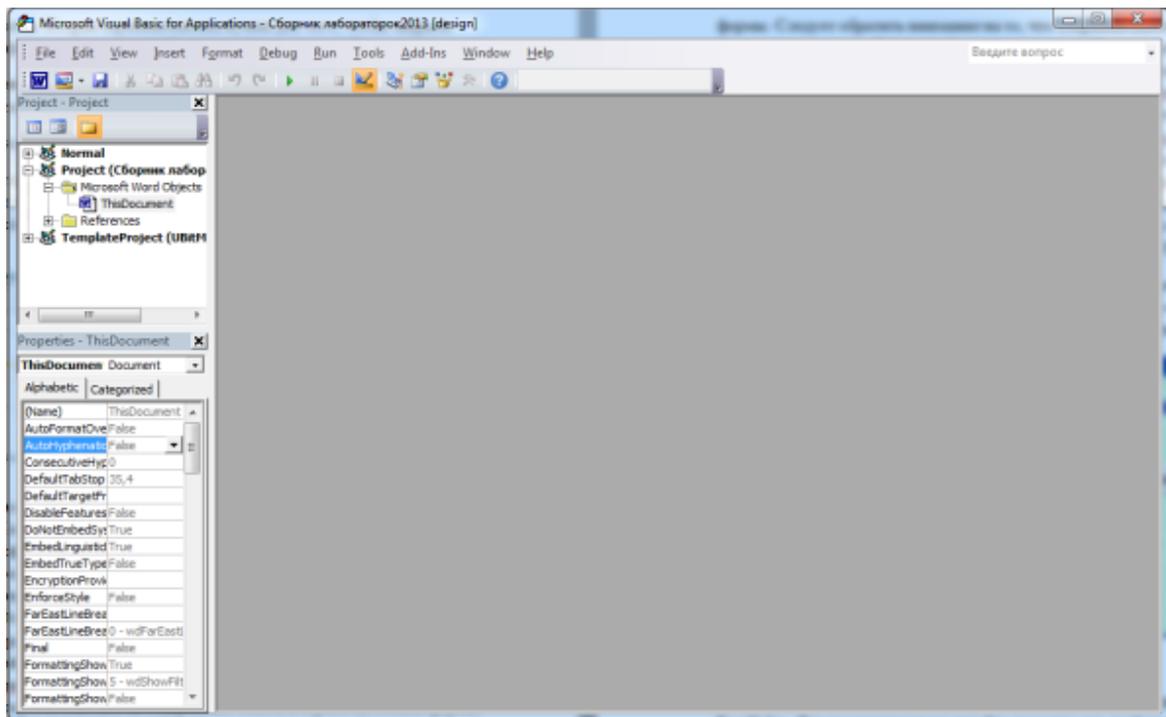


Рис.12.1. Стартовое окно среды VBA

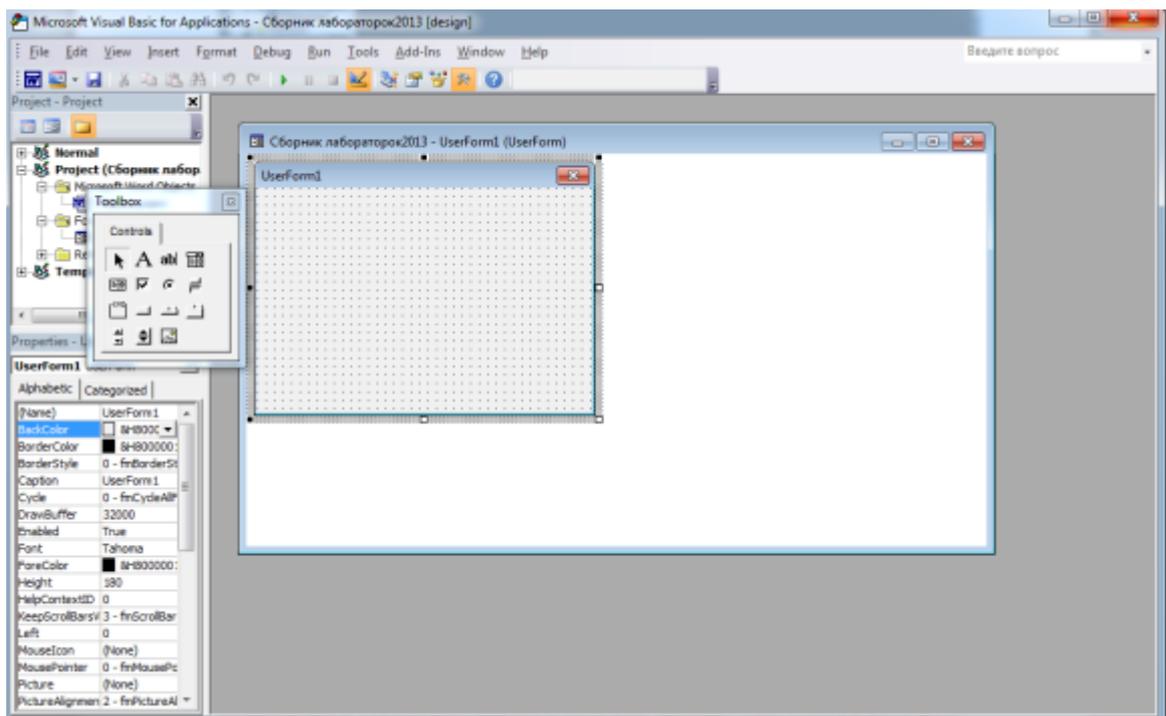


Рис. 12.2. Создание экранной формы

Пользовательская форма является таким же объектом программы, как и все элементы интерфейса, которые на ней расположены. Поэтому можно менять свойства формы, используя окно свойств **Properties**. Окно формы имеет строку заголовка, и одним из свойств формы является содержимое текста, находящегося в строке заголовка. По умолчанию в этой строке выводится имя основной формы – **UserForm1**.

Для того чтобы изменить содержание заголовка, сделав его более информативным, необходимо обратиться к окну свойств **Properties**. Так как на форме в начальный момент нет ещё ни одного объекта, то окно **Properties** показывает свойства самой формы.

4. В окне свойство **Caption**, которое и отвечает за содержимое текста в строке заголовка формы, щёлкните мышью в поле, находящемся справа от названия свойства. В этом поле появится вертикальный курсор, наличие которого говорит о том, что данное поле можно редактировать. Удалите находящееся в этом поле значение **UserForm1** и вместо него введите с клавиатуры следующее значение:

Первая программа Сидорова в VBA.

Этот же текст теперь отобразится и в строке заголовка экранной формы (рис.12.3).

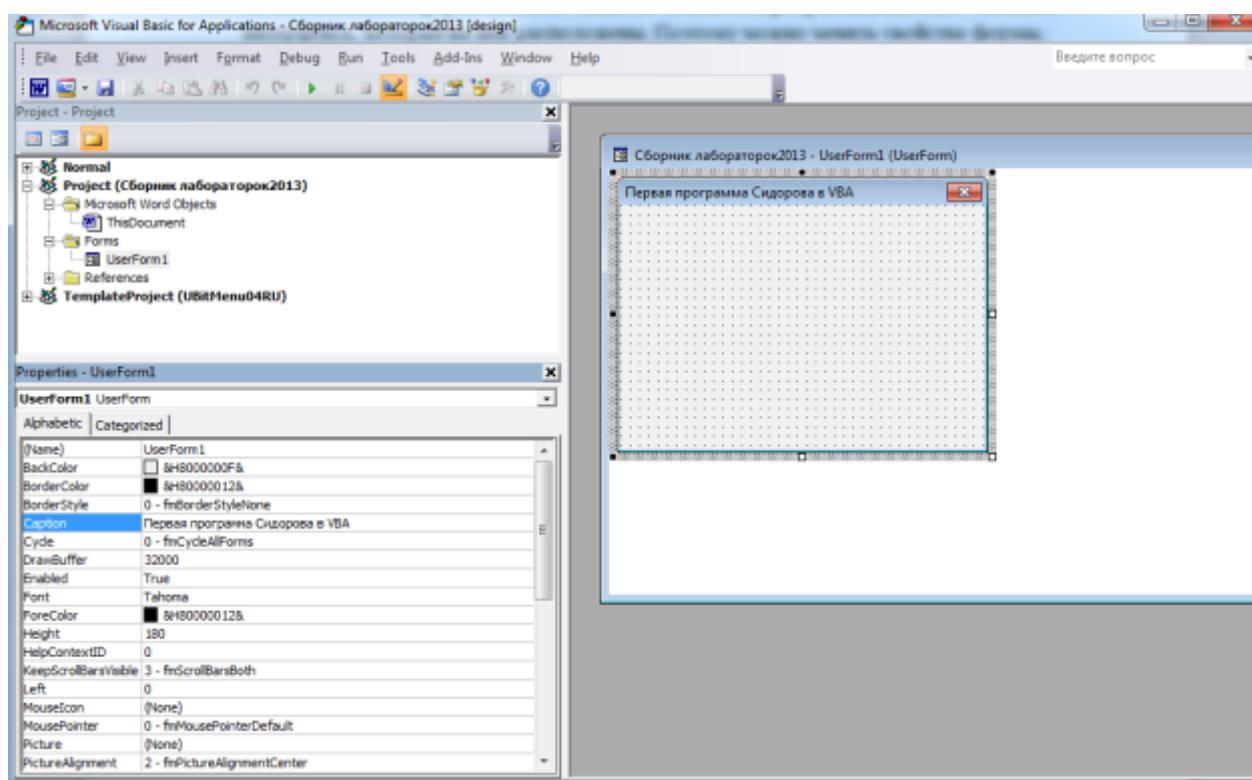


Рис. 12.3. Задание заголовка формы

5. Следующим шагом в настройке свойств основной формы является изменение фонового цвета самой формы.

По умолчанию цвет основной формы является светло-серым. Для изменения цвета следует в окне **Properties** найти свойство **BackColor**, которое отвечает за изменение фонового цвета формы. Щёлкните по раскрывающей стрелке, находящейся в поле справа от названия свойства и откройте палитру для выбора другого цвета формы.

ВНИМАНИЕ. Открывающаяся палитра имеет две вкладки: *System* (системная) и *Palette* (собственно палитра). По умолчанию открывается вкладка *System*, которая содержит огра-

ниченный набор цветов. Переключившись щелчком мыши на вкладку *Palette*, можно открыть палитру с большим количеством цветов.

Для данного проекта (работы) выбрать в этой палитре светло-зелёный цвет, который должен стать фоновым цветом формы.

6. Создайте на форме объект **Label** (Надпись), для чего на Панели элементов щёлкните кнопку с прописной буквой **A**. Появившийся на экранной форме объект получает по умолчанию имя **Label1**.

Задайте для данного объекта содержание текста надписи, для чего найдите в окне **Properties** объекта **Label1** свойство **Caption** и вместо стоящего там по умолчанию значения «**Label1**» введите новый текст (рис.12.4) :

«**Я программирую в системе VBA**».

7. Настройте шрифт надписи, для чего найдите в окне **Properties** свойство **Font** и щелчком на кнопке-построителе откройте диалоговое окно «**Шрифт**». В диалоговом окне задайте размер шрифта – 18 пунктов и начертание шрифта – жирный.

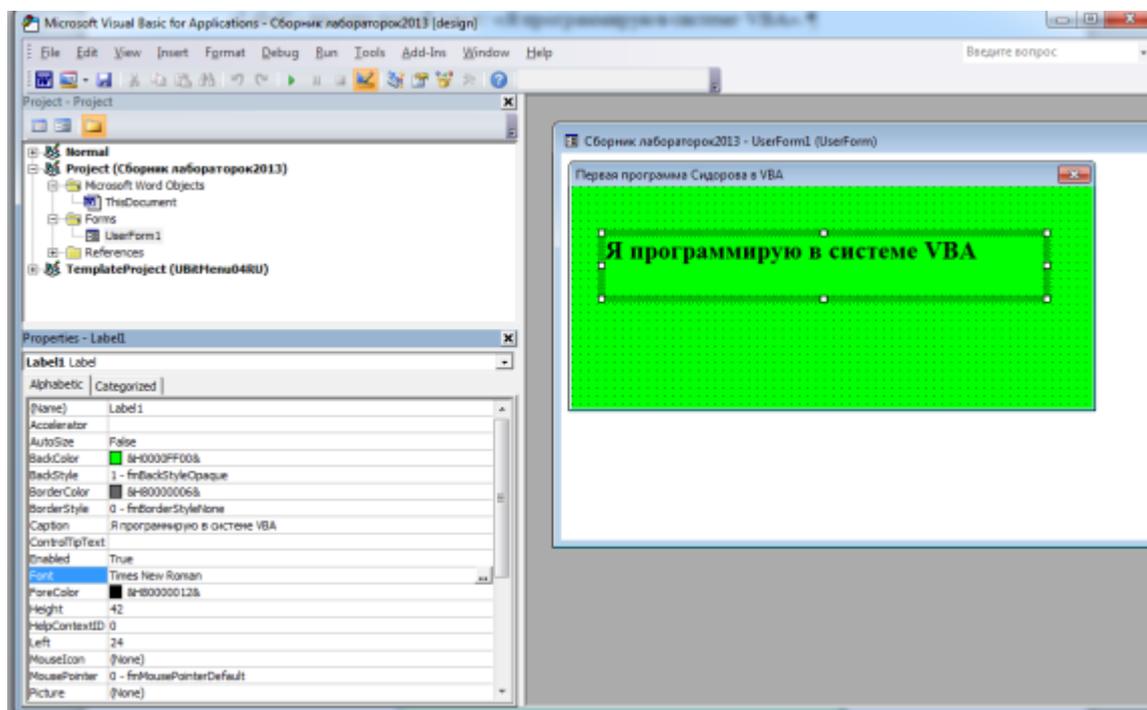


Рис. 12.4. Создание надписи

8. Установите цвет шрифта надписи. Цвет шрифта определяется свойством **ForeColor**. Для изменения цвета шрифта найдите это свойство в окне **Properties**. Затем в поле справа от свойства **ForeColor** выберите раскрывающую стрелку. В результате этих действий открывается цветовая палитра, аналогичная палитре для свойства **BackColor**. В открывшейся палитре перейдите на вкладку **Palette** и выберите на ней цвет шрифта (в данном проекте – синий).

На этом настройку свойств первой надписи можно считать завершённой.

9. Создайте на экранной форме с помощью Панели элементов две экранные кнопки – **CommandButton1** и **CommandButton2**. Для кнопки

CommandButton1 в окне **Properties** найдите свойство **Caption** и введите следующее содержимое текста: «Сведения об авторе». На кнопке **CommandButton2** создайте текст, состоящий из одного слова: «Выход».

Для обеих кнопок с помощью свойства **Font** установите размер шрифта, равный 14 пунктам. Цвет шрифта в данном случае не меняем – он останется черным.

10. Ниже экранных кнопок поместите надпись, которая по умолчанию получает имя **Label2**. Эта надпись будет содержать сведения о разработчике приложения.

11. Затем для данной надписи проведите настройку ее свойств. Для свойства **Caption** установите содержание: «Программу разработал ...» (вместо многоточия ставим фамилию и инициалы студента, выполняющего данную работу). Для свойства **Font** укажите размер шрифта, равный 16 пунктам, и жирное начертание. Используя свойство **ForeColor**, задайте цвет шрифта надписи – красный.

Задайте ещё одно свойство данной надписи. Эта надпись, согласно условиям задания, по умолчанию должна быть невидимой. Для этого в окне **Properties** найдите свойство **Visible** и установите для него значение **False**.

На этом работа по созданию и настройке графического интерфейса пользователя в данном проекте будет завершена. Получившийся в результате проведённой работы пользовательский интерфейс должен соответствовать рис. 12.5.

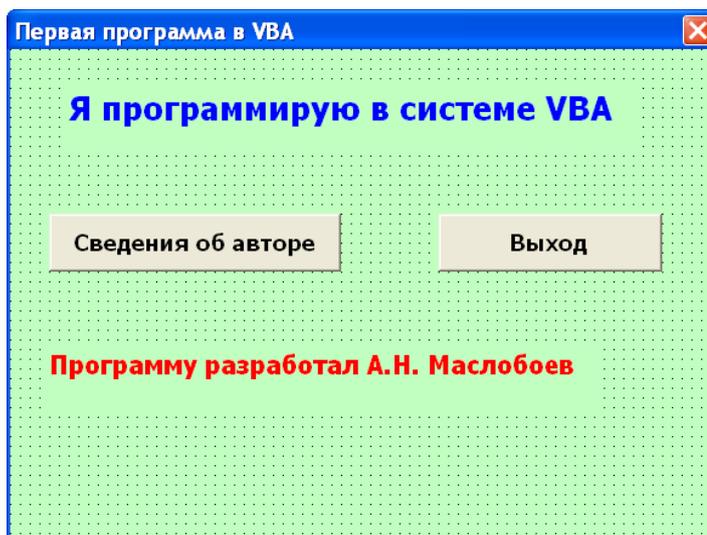


Рис. 12.5. Общий вид интерфейса пользователя

Следующим этапом в работе над проектом является написание программного кода. В данном проекте активными объектами, которые реагируют на программные события, являются две экранные кнопки. Для них нужно написать процедуры, описывающие их реакции на различные внешние воздействия.

Для того чтобы открыть окно кода для какого-либо объекта, достаточно дважды щёлкнуть его мышью. Основным программным событием, на которое должен реагировать любой объект в системе VBA, считается одиночный щелчок левой кнопкой мыши.

12. Дважды щёлкните мышью на экранной кнопке **CommandButton1**, и в результате откроется окно кода с заготовкой для процедуры, соответствующей этой кнопке. Данная заготовка выглядит следующим образом:

```
Private Sub CommandButton1_Click()
```

```
End Sub
```

Первая строка этой заготовки является заголовком процедуры. В заголовке пишутся служебное слово **Sub** (сокращение от **Subroutine**, что по-английски означает подпрограмма) и собственное имя процедуры. Собственное имя процедуры состоит из имени объекта, для которого она пишется, и название события, на которое должен реагировать этот объект. Между этими двумя частями собственного имени процедуры ставится символ подчёркивания. В данном случае имя объекта **CommandButton1**, а название события – **Click** (по-английски так называется одиночный щелчок мышью). В нижней строке заготовки стоит команда **End Sub** – конец процедуры. Между ними может быть размещен текст самой процедуры.

13. Опишите действие, которое превращает невидимый ранее объект (нижнюю надпись **Label2**) в видимый. Для этого нужно изменить значение свойства **Visible** с **False** (объект не виден) на **True** (объект виден), или, иначе говоря, присвоить свойству **Visible** новое значение. Оператор (команда языка Visual Basic), который выполняет это действие, называется оператором присваивания.

Синтаксис данного оператора следующий. Сначала указывается полное имя изменяемого свойства, которое состоит из имени объекта и собственного имени свойства. Имя объекта и имя свойства отделяются друг от друга с помощью точки. Таким образом, полное имя свойства в данном случае будет: **Label2.Visible**. Затем указывается операция присваивания, которая обозначается в языке **Visual Basic** знаком равенства. После знака равенства указывается новое значение, которое присваивается этому свойству.

Таким образом, целиком оператор, делающий невидимую ранее надпись видимой, будет выглядеть следующим образом:

```
Label2.Visible = True
```

Этот оператор должен быть записан в пустой строке, находящейся между заголовком и концом процедуры. В целом же процедура, описывающая реакцию на щелчок левой кнопкой мыши, будет выглядеть таким образом:

```
Private Sub CommandButton1_Click()
```

```
Label2.Visible = True
```

```
End Sub
```

Но на этом работа с объектом **CommandButton1** ещё не завершена.

Для данной экранной кнопки необходимо написать ещё одну процедуру. Вторая процедура должна при двойном щелчке на кнопке **CommandButton1** делать надпись **Label2** со сведениями об авторе снова невидимой.

14. Для того чтобы создать вторую процедуру, выполните следующие действия. В окне кода в правой верхней его части находится вспомогательное текстовое поле. В нем содержится список программных событий, на которые может реагировать данный объект. По умолчанию в списке указано событие **Click**

(одиночный щелчок левой кнопкой мыши). В текстовом поле справа от названия события имеется направленная вниз треугольная стрелка. Щёлкнув по этой стрелке, можно увидеть весь список программных событий, на которые может реагировать данный объект. В этом списке щелчком мыши выбираем событие **DbClick** (двойной щелчок левой кнопкой мыши), а окне кода появляется шаблон для второй процедуры. Этот шаблон имеет следующий вид:

```
Private Sub CommandButton1_DbClick (ByVal Cancel  
As MSForms.ReturnBoolean)  
End Sub
```

В пустую строку, находящуюся между заголовком и концом процедуры, запишите программный код, который выполняет действие, обратное действию, описанному в предыдущей процедуре, – скрывает надпись со сведениями об авторе. Этот код выглядит следующим образом:

```
Label2.Visible=False
```

В итоге для описания реакции экранной кнопки **CommandButton1** на двойной щелчок левой кнопкой мыши должна быть сформирована процедура следующего вида:

```
Private Sub CommandButton1_DbClick (ByVal Cancel  
As MSForms.ReturnBoolean)  
Label2.Visible=False  
End Sub
```

На этом программирование кнопки **CommandButton1** завершается.

Возвращаемся к основной экранной форме, для чего можно использовать команду меню **View → Object**.

15. Напишите программный код для экранной кнопки **CommandButton2**. Для чего дважды щелкните мышью по этой кнопке, и на экране появится окно кода с шаблоном уже для третьей процедуры, имеющим следующий вид:

```
Private Sub CommandButton2_Click()  
End Sub
```

Этот шаблон аналогичен шаблону первой процедуры, за исключением названия объекта. В пустую строку между заголовком и концом процедуры нужно вставить команду, которая будет закрывать приложение при одиночном щелчке мышью по экранной кнопке **CommandButton2**. Эта команда состоит из одного служебного слова и выглядит так:

```
End
```

А вся процедура для описания реакции второй экранной кнопки на щелчок левой кнопки мыши должна выглядеть так:

```
Private Sub CommandButton2_Click()  
End  
End Sub
```

На этом завершается этап разработки программного кода. Весь текст программы должен выглядеть следующим образом:

```
Private Sub CommandButton1_Click()  
Label2.Visible = True
```

```

End Sub
Private Sub CommandButton1_DblClick(ByVal Cancel As
MSForms.ReturnBoolean)
Label2.Visible = False
End Sub
Private Sub CommandButton2_Click()
End
End Sub

```

16. После завершения разработки программы, необходимо ее запустить на выполнение и проверить в работе. Для этого нужно использовать команду меню **Run** → **Run** или просто щёлкнуть треугольную стрелку в панели инструментов.

На экране должно появиться рабочее окно приложения, внешний вид которого приведён на рис. 12.6.

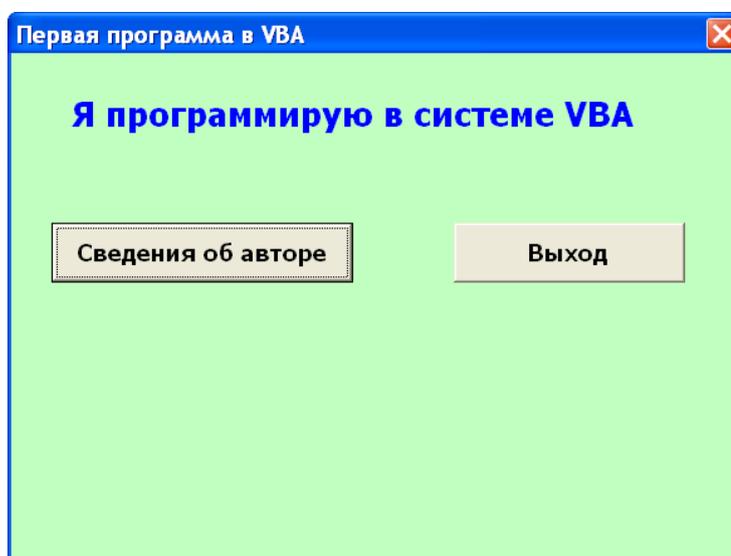


Рис. 12.6. Внешний вид интерфейса созданной программы

ПРИМЕЧАНИЕ. Обратите внимание на то, что при переходе программы в режим выполнения исчезает покрывавшая основное окно координатная сетка, а также становится невидимой надпись со сведениями об авторе.

17. Протестируйте созданную программу. Для этого:
- а. один раз щелкните мышью на экранной кнопке «**Сведения об авторе**», – надпись со сведениями об авторе должна появиться ниже этой кнопки.
 - б. Затем дважды щелкните ту же экранную кнопку, – надпись должна исчезнуть.
 - с. завершаем работу приложения одиночным щелчком на экранной кнопке «**Выход**».

ПРИМЕЧАНИЕ. Если где-то в программном коде разработчик допустил ошибку, то при запуске программы на выполнение появится сообщение об ошибке. Такое сообщение выводится в виде диалогового окна, содержащего текст: «**Compile error**» (Ошибка компиляции).

Ниже, как правило, выводится дополнительная информация о конкретном типе ошибки. Кроме того, как правило, та строка, в которой содержится ошибка, выделяется в окне кода синим цветом, что существенно облегчает поиск этой ошибки. После нахождения и исправления ошибки можно повторно запустить эту программу на выполнение.

18. Когда устранены все ошибки, и программа при тестировании показала свою работоспособность, она должна быть сохранена. Поскольку программа на **Visual Basic** сохраняется вместе с исходным файлом, то следует закрыть систему программирования одним из указанных выше способов (через меню **File** или через закрывающую кнопку в строке заголовка), а затем закрыть файл Первая программа.doc с сохранением сделанных в нем изменений.

О завершении работы сообщите преподавателю и продемонстрируйте работу вашей программы.

Контрольные вопросы

1. Как войти в среду VBA?
3. Как изменить тон фона?
4. Как создать надпись?
5. Как запустить проект на исполнение?
6. Как сохранить проект?
7. Как сохранить проект, чтобы он был доступен из любого файла?
8. Как сделать надпись невидимой?
9. Можно ли поменять фон экранной формы после сохранения проекта?
10. Что такое CommandButton?

Литература/источники

1. Программирование на VBA и создание макросов, электронный ресурс: http://www.lessons-tva.info/edu/e-inf2/m2t3_7.html
2. Самоучитель программирования на VBA в MS Office, электронный ресурс: http://www.libray.narod.ru/rapid/vba_programmirovanie_office.htm
3. Стив Камминг «VBA для чайников», электронный ресурс: http://www.proklondike.com/books/vb/vb_Cummings_VBA_Dummies.html

Работа № 13

ОСВОЕНИЕ СРЕДЫ ПРОГРАММИРОВАНИЯ VBA в MS WORD

Цель работы: Расширить навыки практической работы в среде программирования VBA.

В результате выполнения данной работы студенты должны закрепить навыки работы в среде программирования VBA и разработки простейших программ с использованием инструментария данной среды.

В конце лабораторной работы студенты должны представить отчёт по работе преподавателю в виде демонстрации работы разработанной программы.

Программа, которая разрабатывается в данной работе, должна изменять цвет экранной формы с помощью группы из 8 переключателей. В Windows переключатели (другое название этих элементов – радиокнопки) – это группа элементов управления, в которой активным (включённым) всегда может быть один и только один элемент, а остальные выключены. В данном случае каждый переключатель отвечает за один из цветов, в который будет окрашена экранная форма: белый, чёрный, красный, жёлтый, синий, зелёный, фиолетовый, бирюзовый.

Указания по порядку выполнения работы

1. Создайте новый файл Microsoft Word (например, Цвета.doc), аналогично предыдущей работе.

2. С помощью команды **Сервис** → **Макрос** → **Редактор Visual Basic** (или комбинацию клавиш **Alt-F11**) откройте экран системы VBA. На этом экране с помощью команды **Insert** → **UserForm** создайте основную пользовательскую форму.

3. На форме создайте пользовательский интерфейс проекта. Пользовательский интерфейс должен включать следующие элементы: рамка, содержащая группу переключателей, сами переключатели, экранные кнопки **«Выход»** и **«Сведения об авторе»**.

4. Задайте текст в строке заголовка пользовательской формы. Для этого нужно щёлкнуть мышью на любом свободном месте внутри формы и затем в окне **Properties** (Свойства) найти свойство **Caption**. В поле, расположенном справа от названия этого свойства, ввести текст: **«Изменение цвета формы»**. Этот текст будет отображаться в строке заголовка основной формы проекта.

5. Создайте рамки, внутри которой будут размещены переключатели, следующим образом – на панели элементов **Toolbox** найдите элемент **Frame** (в русифицированных версиях VBA он называется – «Рамка»). Этот элемент выглядит следующим образом:



Щёлкнув мышью по данному элементу, а затем по экранной форме, поместите этот элемент на форму, создавая новый объект **Frame1**.

Объект **Frame1** имеет на границах маркеры (маленькие белые квадраты), с помощью которых можно увеличивать или уменьшать размеры этого объекта. Размеры рамки следует изменить таким образом, чтобы в ней разместились 8 переключателей с надписями.

Подобно другим элементам пользовательского интерфейса в VBA, объект **Frame1** имеет свойство **Caption**, отвечающее за содержание текста, который виден в верхней части рамки. В окне **Properties** для рамки найти свойство **Caption** и в поле справа ввести его значение: «**Выберите цвет формы**».

6. Поместите внутрь рамки переключатели. Элемент **OptionButton** (переключатель) находится на панели элементов **Toolbox** и выглядит так:



Последовательно разместите переключатели один под другим. Этим элементам управления присвойте имена: **OptionButton1**, **OptionButton2**, **OptionButton3**, **OptionButton4**, **OptionButton5**, **OptionButton6**, **OptionButton7**, **OptionButton8**. Каждый из этих переключателей также имеет свойство **Caption**. В данном случае свойство **Caption** отвечает за надпись, которая расположена справа от самого переключателя. Для переключателя **OptionButton1** в свойстве **Caption** задать наименование «**Белый**», для **OptionButton2** в этом свойстве - «**Чёрный**», для **OptionButton3** – «**Красный**», **OptionButton4** – «**Жёлтый**», **OptionButton5** – «**Синий**», **OptionButton6** – «**Зелёный**», **OptionButton7** – «**Фиолетовый**», **OptionButton8** – «**Бирюзовый**».

7. Создайте экранные кнопки **CommandButton1** и **CommandButton2**. Для первой кнопки в **Caption** задайте фразу «**Сведения об авторе**», для второй — «**Выход**». Для всех надписей на рамке, переключателях и экранных кнопках настройте необходимый шрифт с помощью свойства **Font**.

На этом создание графического интерфейса пользователя для данной программы будет завершено. Внешний вид интерфейса должен соответствовать указанному на рис. 13.1.

8. Теперь напишите программный код для переключателей и кнопок. Процедура, описывающая работу каждого переключателя, должна окрашивать основную экранную форму **UserForm1** в один из перечисленных цветов. Экранная форма, как известно из работы 11, имеет свойство **BackColor** (цвет фона), которому и присваивается цветовое значение.

Цвет в Visual Basic может задаваться двумя способами: в виде шестнадцатеричного числа или в виде словесной константы этого языка. В данной работе следует использовать второй способ обозначения цвета, как более наглядный.

Константа, соответствующая определённому цвету, состоит из приставки **vb** (сокращение от Visual Basic) и названия конкретного цвета. Белому цвету соответствует константа **vbWhite**, черному – **vbBlack**, красному – **vbRed**, жёлтому – **vbYellow**, синему – **vbBlue**, зелёному – **vbGreen**, фиолетовому – **vbMagenta**, бирюзовому – **vbCyan**.

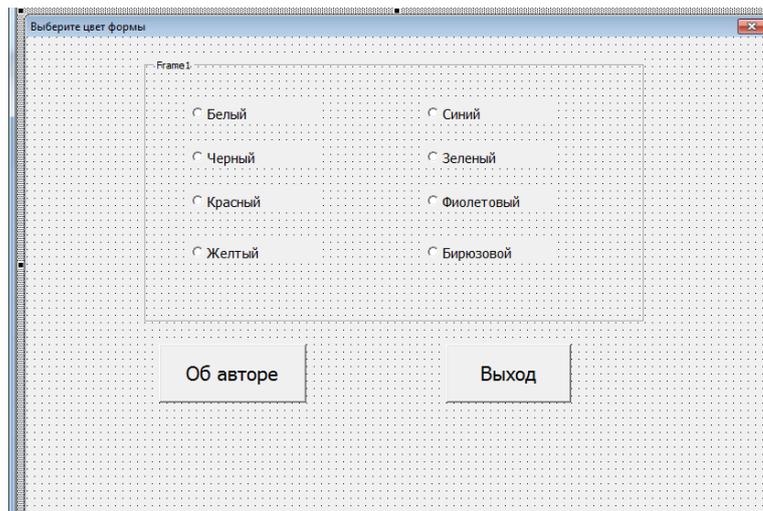


Рис. 13.1. Графический интерфейс пользователя

Для окрашивания основной формы в белый цвет создайте следующую запись:

```
UserForm1.BackColor = vbWhite
```

9. Аналогичным образом сформируйте операторы для окрашивания формы в другие цвета. Следует обратить внимание на то, что пространство, находящееся внутри рамки, данными командами не закрашивается.

10. Сформируйте программный код для первого переключателя, для чего на форме дважды щелкните мышью переключатель **OptionButton1**, появится окно кода с шаблоном следующего вида:

```
Private Sub OptionButton1_Click()  
End Sub
```

В пустую строку, находящуюся между заголовком и концом процедуры, запишите сформированный выше оператор задания цвета.

Процедура для первой радиокнопки должна выглядеть следующим образом:

```
Private Sub OptionButton1_Click()  
UserForm1.BackColor = vbWhite  
End Sub
```

11. Сформируйте программный код для второго переключателя, для чего вернитесь к основной форме и дважды щелкните мышью на переключателе **OptionButton2**. В результате появится окно кода с шаблоном для второй процедуры. В пустой строке этой процедуры запишите оператор:

```
UserForm1.BackColor = vbBlack
```

Процедура для второго переключателя **OptionButton2** в целом будет выглядеть так:

```
Private Sub OptionButton2_Click()  
UserForm1.BackColor = vbBlack  
End Sub
```

Выполнение этой процедуры при щелчке по второму переключателю будет окрашивать экранную форму в чёрный цвет

12. Аналогичным образом запрограммируйте остальные шесть переключателей, находящихся внутри рамки.

13. Напишите программный код для экранной кнопки **CommandButton1** (кнопка «**Об авторе**»). В данной работе эта кнопка должна работать иначе, чем в предыдущей, поскольку пользовательский интерфейс данной программы и так содержит достаточно большое количество элементов, и его не следует перегружать, создавая на основной форме ещё одну надпись. В данном проекте щелчок мышью по кнопке «**Об авторе**» должен выводить на экран дополнительное диалоговое окно, содержащее информацию о разработчике программы, и экранную кнопку «**Ок**», закрывающую диалоговое окно.

Для создания такого окна используется команда **Msgbox**. В данной команде после служебного слова **Msgbox** в круглых скобках и в кавычках указывается текст, выводимый в появляющемся диалоговом окне.

Пример реализации такой команды:

```
Msgbox(«Программу разработала М.А. Ежова»)
```

Запишите указанную команду в соответствующую строку процедуры **CommandButton1_Click**, расположенную между заголовком процедуры и концом процедуры. Процедура в целом должна выглядеть так:

```
Private Sub CommandButton1_Click()
```

```
Msgbox(«Программу разработала М.А.Ежова»)
```

```
End Sub
```

Выполнение этой команды при щелчке на экранной кнопке и выведет на экран диалоговое окно, которое появится поверх основной экранной формы. Внешний вид данного окна приведен на рис. 13.2. После ознакомления с данным окном пользователь может закрыть его щелчком на экранной кнопке «**ОК**» и вернуться к работе с основным окном приложения.

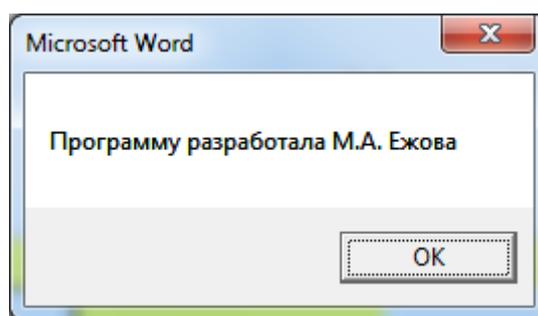


Рис. 13.2. Диалоговое окно с информацией о разработчике

14. Запрограммируйте экранную кнопку **CommandButton2** («**Выход**»), закрывающая данное приложение по окончании его работы аналогично тому, как это делалось в предыдущей работе.

В итоге после написания программного кода для всех объектов приложения должен получиться следующий листинг:

```
Private Sub CommandButton1_Click()
```

```
MsgBox ("Программу разработала М.А. Ежова")
```

```
End Sub
```

```

Private Sub CommandButton2_Click()
End
End Sub
Private Sub OptionButton1_Click()
UserForm1.BackColor = vbWhite
End Sub
Private Sub OptionButton2_Click()
UserForm1.BackColor = vbBlack
End Sub
Private Sub OptionButton3_Click()
UserForm1.BackColor = vbRed
End Sub
Private Sub OptionButton4_Click()
UserForm1.BackColor = vbYellow
End Sub
Private Sub OptionButton5_Click()
UserForm1.BackColor = vbBlue
End Sub
Private Sub OptionButton6_Click()
UserForm1.BackColor = vbGreen
End Sub
Private Sub OptionButton7_Click()
UserForm1.BackColor = vbMagenta
End Sub
Private Sub OptionButton8_Click()
UserForm1.BackColor = vbCyan
End Sub

```

15. После завершения работы над программным кодом запустите программу на выполнение и убедитесь, что интерфейс спроектирован верно и на экране вид аналогичный рис. 13.3, где активным является переключатель, окрашивающий экранную форму в зеленый цвет.

16. Протестируйте работу всех восьми переключателей, имеющих на форме, и кнопки, открывающей окно со сведениями об авторе. Завершите работу приложения щелчком на экранной кнопке «Выход».

*ПРИМЕЧАНИЕ. Если где-то в программном коде допущена ошибка, то при запуске программы на выполнение появится сообщение об ошибке. Такое сообщение выводится в виде диалогового окна, содержащего текст: «**Compile error**» (Ошибка компиляции). Ниже, как правило, выводится дополнительная информация о конкретном типе ошибки. Кроме того, как правило, та строка, в которой содержится ошибка, выделяется в окне кода синим цветом, что существенно облегчает поиск этой ошибки. После нахождения и исправления ошибки можно повторно запустить эту программу на выполнение*

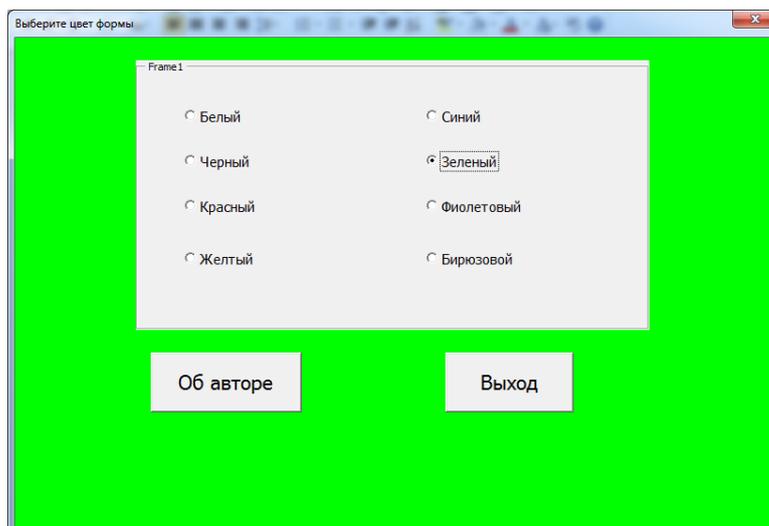


Рис. 13.3. Окончательный вид интерфейса программы

17. Когда устранены все ошибки, и программа при тестировании показала свою работоспособность, она должна быть сохранена. Поскольку программа на Visual Basic сохраняется вместе с исходным файлом, то нужно закрыть систему программирования одним из указанных выше способов (через меню File или через закрывающую кнопку в строке заголовка), а затем закрыть файл **Первая программа.doc** с сохранением сделанных в нем изменений.

О завершении работы сообщите преподавателю и продемонстрируйте работу Вашей программы.

Контрольные вопросы

1. Что такое макрокоманда (макрос)?
2. Какую последовательность действий нужно выполнить в VBA для создания группы переключателей?
3. Какую последовательность команд нужно использовать в программе Word для того, чтобы войти в среду программирования VBA?
4. Как вывести в среде VBA на экран окно программного кода?
5. Какое свойство определяет фоновый цвет объекта?
6. Какое свойство определяет видимость или невидимость объекта?
7. Какие основные объекты можно создавать в VBA с помощью панели элементов?
8. Какие свойства определяют основные характеристики шрифта, которым сделана надпись на объекте?
9. Что содержит Toolbox?
10. Для чего в работе используется OptionButton?

Литература/источники

1. Программирование на VBA и создание макросов, электронный ресурс: http://www.lessons-tva.info/edu/e-inf2/m2t3_7.html
2. Самоучитель программирования на VBA в MS Office, электронный ресурс: http://www.libray.narod.ru/rapid/vba_programmirovanie_office.htm
3. Стив Камминг «VBA для чайников», электронный ресурс: http://www.proklondike.com/books/vb/vb_Cummings_VBA_Dummies.html
4. Программирование на VBA, электронный ресурс: <http://itteach.ru/vba/programmirovanie-na-vba>

Работа № 14 ИСПОЛЬЗОВАНИЕ УСЛОВНОГО ОПЕРАТОРА

Цель работы: Расширить навыки практической работы в среде программирования VBA.

В результате выполнения данной работы студенты должны закрепить навыки работы в среде программирования VBA и разработки простейших программ с использованием оператора условного перехода.

В конце лабораторной работы студенты должны представить отчет по работе преподавателю в виде демонстрации работы тестовой программы.

Постановка задачи. Некоторая компьютерная фирма для стимулирования клиентов к приобретению ее продукции предложила следующую схему. Любой клиент, который приобретает у фирмы ее продукцию на сумму свыше 10000 руб., получает скидку в размере 5 % от стоимости приобретаемой продукции. Требуется разработать программу, которая в зависимости от введенной пользователем суммы определяет размер скидки на продукцию и сообщает пользователю стоимость приобретаемой им продукции с учетом скидки. Если же приобретаемая пользователем продукция стоит менее 10000 руб., то программа должна выдать сообщение о том, что данный пользователь не имеет права на скидку.

Введение

При решении многих задач из области программирования возникает необходимость выбора одного варианта действий из двух или более возможных. Для того, чтобы правильно осуществить этот выбор, в языках программирования используются специальные операторы. Такие операторы есть и в языке VBA. Если необходимо выбрать один вариант из двух возможных, то применяется оператор условного перехода. Оператор условного перехода выбирает тот или иной вариант действия в зависимости от значения некоторого условия. Общий вид этого оператора в языке VBA следующий:

If условие Then вариант 1 Else вариант 2.

Механизм работы этого оператора следующий: вначале проверяется условие, находящееся после служебного слова If. В качестве такого условия обычно выступает какая-либо операция сравнения. Результатом выполнения такой операции является логическая величина, которая может принимать только два значения: True (истина) или False (ложь). Если условие истинно, то программа выполняет вариант 1, находящийся после служебного слова Then. Если же условие ложно, то выполняется вариант 2, который находится после служебного слова Else. Каждый из этих вариантов в свою очередь может представлять собой один оператор или группу операторов языка VBA.

Приведенная выше форма записи условного оператора If ... Then ... Else называется однострочной, так как все служебные слова и соответствующие им варианты записаны в одну строку. Эта форма записи не всегда бывает удобна для программиста. Если какой-либо из вариантов (или оба) представляют собой до-

статочно длинную последовательность операторов, то строка выйдет за пределы экрана, и для того чтобы отредактировать такую строку (или даже просто посмотреть ее содержимое), придется постоянно «прокручивать» ее в ту или иную сторону. Поэтому наряду с однострочной формой записи применяется и другая форма записи условного оператора, называемая блочной. Для блочной формы записи общий вид данного оператора будет следующим:

If условие Then
вариант 1
Else
вариант 2
End If

Синтаксис оператора при такой форме записи отличается наличием служебных слов End If, которые необходимы для того, чтобы можно было определить, где заканчивается данный оператор. Механизм работы условного оператора при блочной форме записи не отличается от механизма при однострочной, но блочная форма записи часто бывает удобней для восприятия и понимания текста программы.

Такой вид условного оператора называется полным условным оператором (это относится и к однострочной, и к блочной форме записи). Наряду с ним в программировании используется и сокращенный условный оператор. Общий вид сокращенного условного оператора (для однострочной формы записи) следующий:

If условие Then вариант

В этом случае механизм работы условного оператора изменится следующим образом. Если условие, находящееся после служебного слова If, является истинным, то выполняется тот вариант, который указан после слова Then (в данном случае описываемый вариант так же может представлять собой один оператор или группу операторов). Но если условие, находящееся после If, оказывается ложным, то данный оператор не выполняет никаких действий и программа переходит к выполнению следующего оператора. При блочной форме записи сокращенный условный оператор в общем виде будет выглядеть следующим образом:

If условие Then
вариант
End If

Можно применять условный оператор и для выбора одного варианта из трех имеющихся. В этом случае в общем виде условный оператор будет выглядеть так:

If условие 1 Then
вариант 1
ElseIf условие 2 Then
вариант 2
Else
вариант 3
End If

Для такого условного оператора применяется следующий механизм работы. В начале проверяется условие 1, находящееся в верхней строке оператора после

служебного слова If. Если это условие истинно (его значение равно True), то выполняется вариант 1, расположенный между служебными словами Then и ElseIf.

Если же исходное условие 1 ложно (его значение равно False), то производится новая проверка. На этот раз проверяется условие 2. Если значение этого условия равно True, то выполняется вариант 2, находящийся между вторым служебным словом Then и Else. Если же условие 2 тоже оказывается ложным, то выполняется вариант 3, расположенный между служебными словами Else и End If.

Порядок выполнения

1. Создайте интерфейс заданной программы, который должен включать следующие элементы:

1.1. Три текстовых окна. В первое окно TextBox1 будет вводиться исходная сумма. Во втором окне TextBox2 будет выводиться сумма скидки. Это окно по умолчанию сделайте невидимым, так как в случае отсутствия у пользователя права на скидку в наличии окна TextBox2 на экранной форме нет смысла. Это окно будет видимым только в том случае, если сумма заказа превышает 10000 руб.

Для того чтобы сделать окно TextBox2 невидимым, в окне свойств для данного объекта найдите свойство Visible и установите для него значение False. При работе в режиме проектирования программы это окно все равно будет оставаться видимым, но как только вы запустите программу на выполнение, это окно «исчезнет» с экранной формы и «проявится» только при указанном выше условии.

В третьем текстовом окне TextBox3 будет отображаться стоимость покупки с учетом положенной клиенту скидки, которая равна исходной сумме покупки за вычетом этой скидки. Если скидка клиенту не положена, то величина, отображаемая в текстовом окне TextBox3, будет равна исходной величине, вводимой в окно TextBox1.

1.2. Три надписи Label1, Label2 и Label3, которые поясняют пользователю назначение каждого из текстовых окон. Для текста первой надписи Label1 («Введите сумму заказа») задайте синий цвет, для текстов второй и третьей надписи (вывод результатов расчетов) – красный цвет. Вторую надпись Label2 («Сумма скидки составляет»), подобно окну TextBox2, также сделаем по умолчанию невидимой, так как ее наличие имеет смысл только при сумме заказа, превышающей 10000 руб. Третья надпись Label3 сообщает об итоговой сумме покупки.

1.3. Четыре экранные кнопки CommandButton1, CommandButton2, CommandButton3 и CommandButton4. Щелчок на кнопке CommandButton1 («Расчет») позволяет подсчитать сумму скидки и итоговую стоимость покупки с учетом скидки. Кнопка CommandButton2 («Сброс») очищает все три текстовых окна и делает невидимым второе текстовое окно и вторую надпись, для того чтобы подготовить программу к новому вводу данных и дальнейшим расчетам. Кнопка CommandButton3 («Об авторе») выводит на экран дополнительное диалоговое окно со сведениями о разработчике программы. Кнопка CommandButton4 («Выход») закрывает основную экранную форму (рис. 14.1).

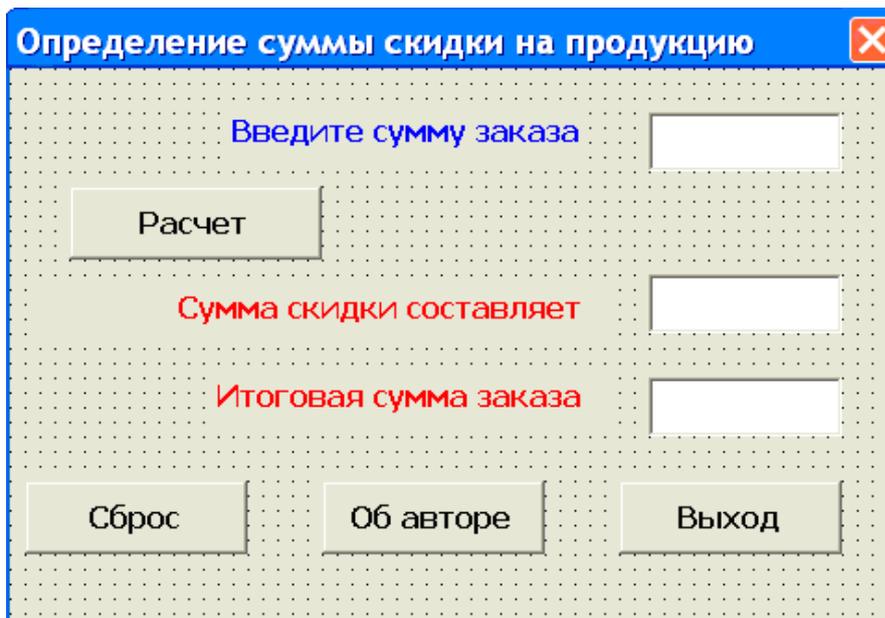


Рис. 14.1. Пользовательский интерфейс программы

2. Написание программного кода начните с кнопки `CommandButton1`. В процедуре `CommandButton1_Click`, описывающей реакцию этой экранной кнопки на щелчок левой кнопкой мыши введем следующие локальные переменные: `Sum` – исходная стоимость приобретаемой продукции; `Skidka` – размер скидки, положенной клиенту фирмы; `Itog` – итоговая стоимость приобретаемой продукции, вычисляемая с учетом скидки. Описываем `Sum` как переменную целого типа, а переменные `Skidka` и `Itog` – как вещественные переменные двойной точности.

3. Затем с помощью функции `Val` преобразуйте значения исходной суммы, введенного в окно `Textbox1`, из текстовой формы в числовую и присвойте полученное числовое значение переменной `Sum`.

4. Создайте в процедуре условный оператор и в заголовке этого оператора после служебного слова `If` запишите следующее условие: `Sum > 10000`.

5. Дальнейший ход действий в программе должен зависеть от значения переменной `Sum`. Если значение этой переменной более 10000, то необходимо выполнить следующие действия:

- Вычислите величину скидки, взяв 5 % от исходной суммы, и присвойте полученное значение переменной `Skidka`.
- Подсчитайте итоговую стоимость приобретаемой продукции, которая будет равна исходной за вычетом скидки. Полученное значение присвойте переменной `Itog`.
- Сделайте видимыми текстовое окно `Textbox2` и соответствующую ему надпись `Label2`.
- Надписи `Label2` присвойте значение: «Сумма скидки составляет».
- Преобразуйте вычисленную сумму скидки и итоговую стоимость из числовой формы в текстовую с помощью функции `Str` и выведите эти текстовые величины соответственно в окнах `Textbox2` и `Textbox3`.

Все эти действия запишите в условном операторе после служебного слова `Then`.

6. Если сумма приобретаемой продукции меньше 10000, то нужно сделать следующее:

– Присвойте переменной Itog значение, равное переменной Summa, так как итоговая сумма будет равна исходной ввиду отсутствия скидки, преобразуйте это значение переменной Itog в текстовую форму и выведите это значение в окне TextBox3.

– Сделайте видимой надпись Label2 и присвойте этой надписи следующее значение: «Извините, скидка Вам не положена».

Эти действия запишите в условном операторе после служебного слова Else.

В целом, исходя из вышеуказанного, условный оператор в данной процедуре должен выглядеть следующим образом:

```
If Sum > 10000 Then  
Skidka = Sum * 0.05  
TextBox2.Text = Str(Skidka)  
TextBox2.Visible = True  
Itog = Sum - Skidka  
TextBox3.Text = Str(Itog)  
Label2.Caption = " Сумма скидки составляет"  
Label2.Visible = True  
Else  
Itog = Sum  
TextBox3.Text = Str(Itog)  
Label2.Visible = True  
Label2.Caption = "Извините, скидка Вам не  
положена"  
End If
```

7. Далее, запишите код процедуры для экранной кнопки CommandButton2. Эта процедура должна делать следующее:

– Очистить все имеющиеся в программе текстовые окна – TextBox1, TextBox2 и TextBox3.

– Сделать невидимой надпись Label2 и текстовое окно TextBox2 до следующего ввода исходных данных.

В случае с первой операцией свойству Text всех трех текстовых окон присваивается «нулевое» значение. Для выполнения второй операции свойству Visible обоих рассматриваемых объектов присваивается значение False. Затем пишите процедуры для экранных кнопок CommandButton3 («Об авторе») и CommandButton4 («Выход»).

Данные кнопки программируются аналогично тому, как это делалось в предыдущих проектах.

После написания программного кода для всех четырех экранных кнопок создание программного модуля для данного проекта можно считать завершённым

8. Сопоставьте полученный текст программного модуля с приведенным ниже:

```
Private Sub CommandButton1_Click()
```

```

Dim Sum As Integer
Dim Skidka, Itog As Double
Sum = Val(TextBox1.Text)
If Sum > 10000 Then
Skidka = Sum * 0.05
TextBox2.Text = Str(Skidka)
TextBox2.Visible = True
Itog = Sum - Skidka
TextBox3.Text = Str(Itog)
Label2.Caption = " Сумма скидки составляет"
Label2.Visible = True
Else
Itog = Sum
TextBox3.Text = Str(Itog)
Label2.Visible = True
Label2.Caption = "Извините, скидка Вам не положена"
End If
End Sub
Private Sub CommandButton2_Click()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
Label2.Visible = False
TextBox2.Visible = False
End Sub
Private Sub CommandButton3_Click()
MsgBox ("Программу разработал И.И. Иванов")
End Sub
Private Sub CommandButton4_Click()
End
End Sub

```

9. После завершения работы над программой следует запустить ее на выполнение и протестировать при различных исходных данных, вводимых в первое текстовое окно.

10. Продемонстрируйте преподавателю работу программы на тестовых примерах.

Контрольные вопросы

1. Что такое условный переход?
2. Какую функцию реализует условный переход?
3. Какую последовательность команд нужно использовать в программе для реализации условного перехода?
4. Как вывести в среде VBA на экран окно программного кода?
5. Поясните принцип работы условного оператора

IF....THEN.....ELSE на языке программирования VBA.

6. Какое свойство определяет видимость или невидимость объекта?
7. Поясните принцип работы оператора MsgBox .
8. Что такое сокращенный условный оператор ?
9. Что содержит Toolbox?
10. Чем отличается полная и сокращенная формы записи условного оператора?

Литература/источники

1. Программирование на VBA и создание макросов, электронный ресурс: http://www.lessons-tva.info/edu/e-inf2/m2t3_7.html
2. Самоучитель программирования на VBA в MS Office, электронный ресурс: http://www.libray.narod.ru/rapid/vba_programmirovanie_office.htm
3. Стив Камминг «VBA для чайников», электронный ресурс: http://www.proklondike.com/books/vb/vb_Cummings_VBA_Dummies.html
4. Программирование на VBA, электронный ресурс: <http://itteach.ru/vba/programmirovanie-na-vba>

Работа № 15

СОЗДАНИЕ GIF-АНИМАЦИИ В ПАКЕТЕ Adobe Photoshop

Цель работы: Освоение способов создания анимированных изображений в формате GIF в онлайн версии многофункционального графического редактора Adobe Photoshop.

Введение

GIF (англ. Graphics Interchange Format – формат для обмена изображениями) – формат растровых графических изображений, позволяющий создавать простую анимацию – чередующиеся кадры. Анимация представляет собой последовательность из нескольких статичных кадров и информацию о времени показа каждого кадра на экране. Можно сделать, чтобы вслед за последним кадром начиналось воспроизведение первого кадра и т.д. Такая анимация называется цикличной.

Формат GIF способен хранить сжатые данные (алгоритм сжатия LZW, Лемпеля-Зива-Велча). Не зависящий от аппаратного обеспечения формат GIF был разработан в 1987 г. фирмой CompuServe для передачи растровых изображений по компьютерным сетям. В 1989-м формат был модифицирован, была добавлена поддержка анимации.

Число цветов палитры – от 2 до 256. Каждый цвет палитры имеет глубину 24 бита на 1 пиксел (выбор из 16 миллионов цветов). Число повторов анимации – бесконечное. Время показа одного кадра анимации – от 0,01 секунды до 655 секунд. Число кадров анимации – неограниченно.

Процесс подготовки GIF-анимации можно условно разделить на несколько этапов:

- создание отдельных кадров;
- генерация движущегося изображения;
- оптимизация целевого файла.

Для создания последовательности растровых изображений и для их преобразования в GIF-анимацию можно использовать многофункциональные растровые редакторы (Adobe Photoshop, CinePaint, cosmigo Pro Motion, PhotoLine) или специальные редакторы GIF-анимации (Gif Construction Set(tm) Professional, Animagic Gif, Ulead Gif Animator, WWW Gif Animator, Xara Webster, Gif Movie Gear).

В Интернете также есть ресурсы, предлагающие создание GIF-анимации онлайн: <http://gifovina.ru/>, <http://minimultik.ru/>, <http://ru.toolson.net/GifAnimation>.

При создании GIF-анимации можно задать величину задержки появления каждого кадра, чем она меньше, тем лучше качество анимации. Кроме того, можно установить количество повторений (от одного до бесконечности) последовательности кадров, хранящихся в GIF-файле. Большое количество кадров ведет к лучшему качеству анимации, но при этом увеличивает размер GIF-файла.

Хотя к настоящему времени создан новый формат PNG, формат GIF по-прежнему широко используется и особенно востребован и широко используется на страницах сайтов в Интернете.

Указания по выполнению работы

Выполнение работы начинается с создания файла отчета по форме, указанной в Приложении к данной работе и заполнении его личными данными студента (дисциплина, группа, ФИО). Именем файла с отчетом должна быть фамилия студента с его инициалами. К файлу отчета прилагается файл GIF-анимации. Именем файла GIF-анимации должна быть фамилия студента с его инициалами.

ПРИМЕЧАНИЕ. В данной работе не рекомендуется использовать Internet Explorer. Рекомендуется – Chrome.

После создания файла отчета можно приступать к выполнению работы.

1. Запустите браузер.

2. Откройте страницу <http://photoshop.domfailov.ru/> (рис. 15.1).

На данном сайте находится бесплатная онлайн версия Adobe Photoshop.

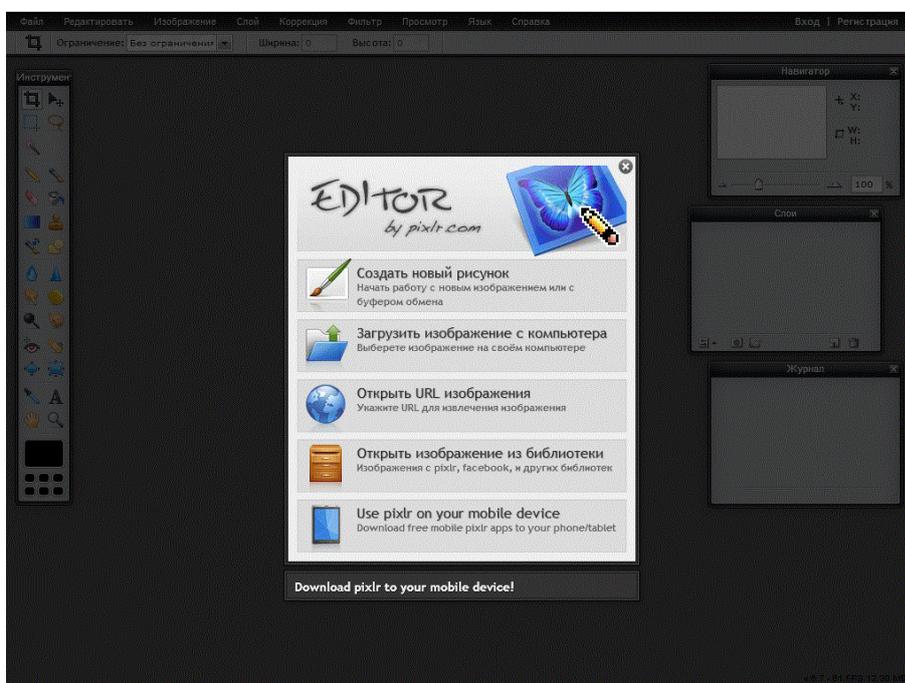


Рис. 15.1. Первая страница сервера <http://photoshop.domfailov.ru>

3. Нажмите **Загрузить изображение с компьютера**. Из папки с номером учебной группы загрузите файл Часы.jpg (рис. 15.2).

4. С помощью панели «Навигатор» увеличьте масштаб изображения до 250 % и далее работайте с таким масштабом.

Для перемещения по изображению используйте палитру инструмент «Рука» .

5. Используя инструмент **Многоугольное лассо** , выделите секундную красную стрелку (рис. 15.3), для чего выполните следующие действия:  .

6. В строке меню нажмите **Редактировать**, затем **Вырезать**. Вырезанную стрелку поместите на новый рисунок. Для создания нового рисунка в строке меню

нажмите **Файл**, затем **Новый рисунок**. Затем нажмите **Редактировать**, затем **Вставить** (рис. 15.4).

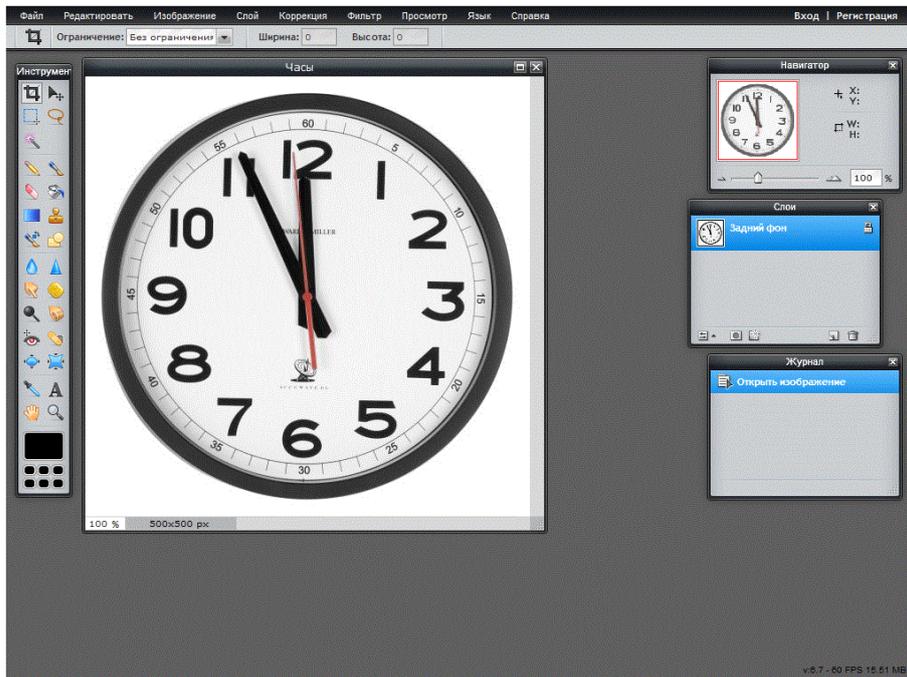


Рис.15.2. Вид загруженного файла ЧАСЫ

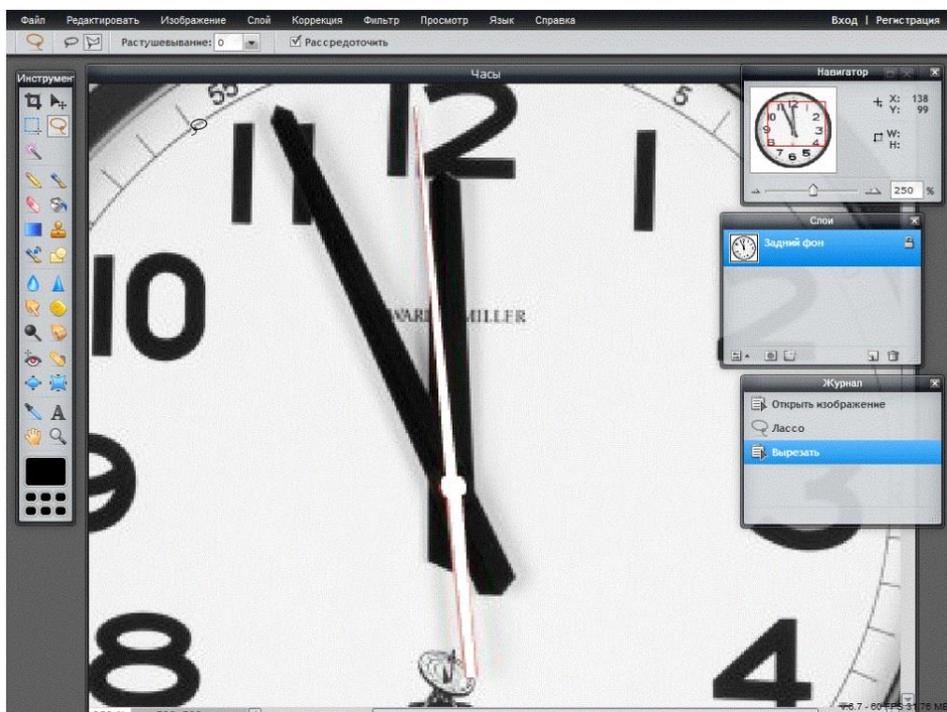


Рис. 15.3. Выделение секундной стрелки

7. Отредактируйте изображения: закрасьте часовую стрелку и другие элементы циферблата часов. Для определения цвета раскрашивания используйте

инструмент **Пипетка**



. Для раскрашивания используйте инструмент **Кисть**



(рис. 15.5).

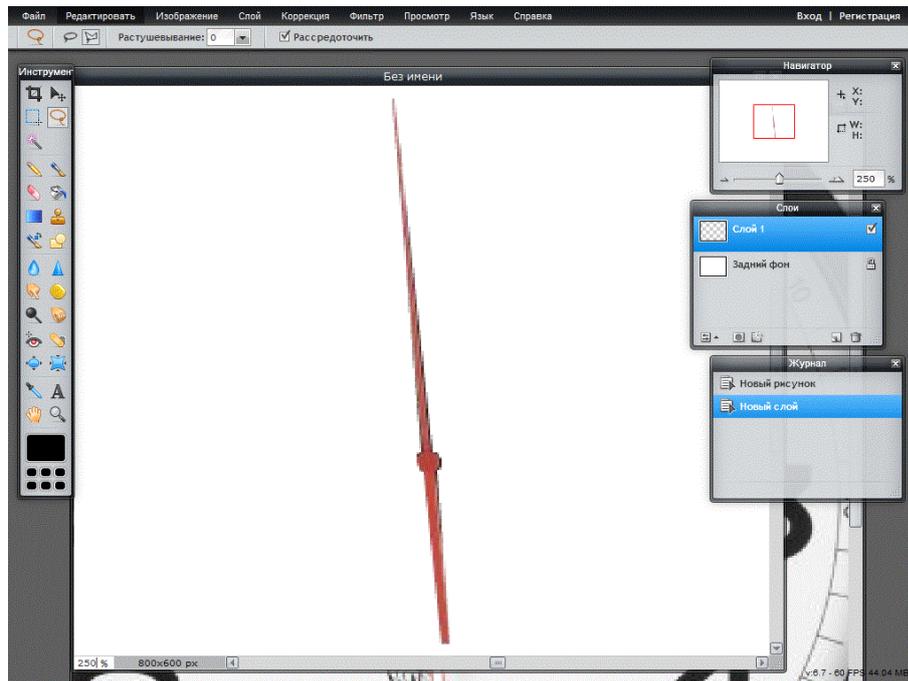


Рис. 15.4. Новый рисунок с секундной стрелкой

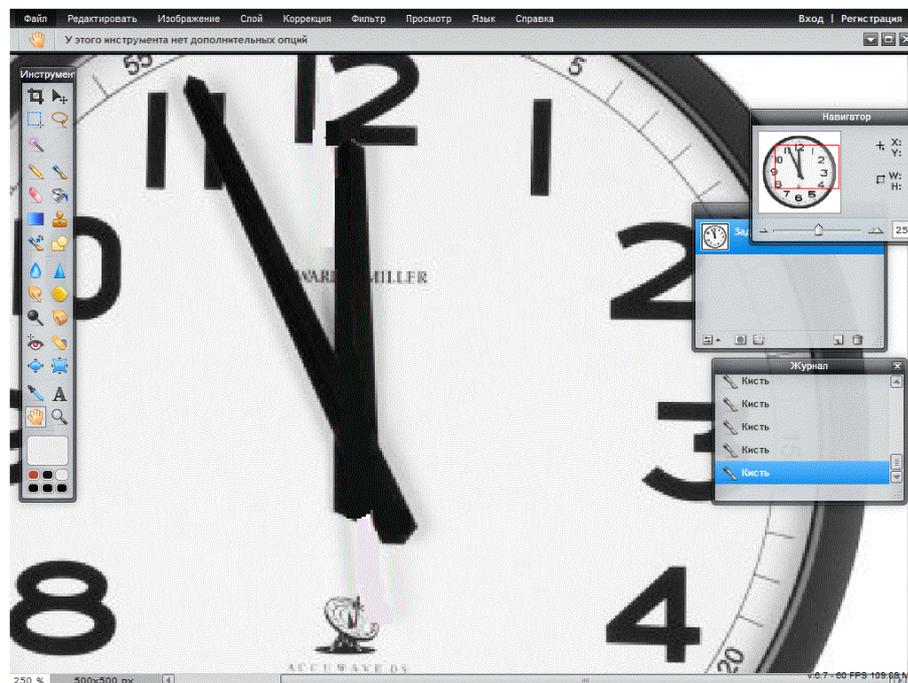


Рис. 15.5.

8. Верните минутную стрелку на циферблат часов (рис. 15.6). При помощи инструмента **Перемещение** расположите стрелку в предназначенное для нее место на циферблате. В строке меню выберите **Редактировать**,

Free transform. Либо используйте элемент  на панели инструментов. Разверните стрелку, чтобы она показывала на 60 секунд.



Рис. 15.6

9. Сохраните изображение в формате JPG. Для этого в строке меню нажмите **Файл**, затем **Сохранить**.

10. В строке меню выберите **Редактировать**, затем **Free transform**. Разверните стрелку, чтобы она показывала на 1 секунду (рис. 15.7). Сохраните изображение в формате JPG, дав файлу другое имя.

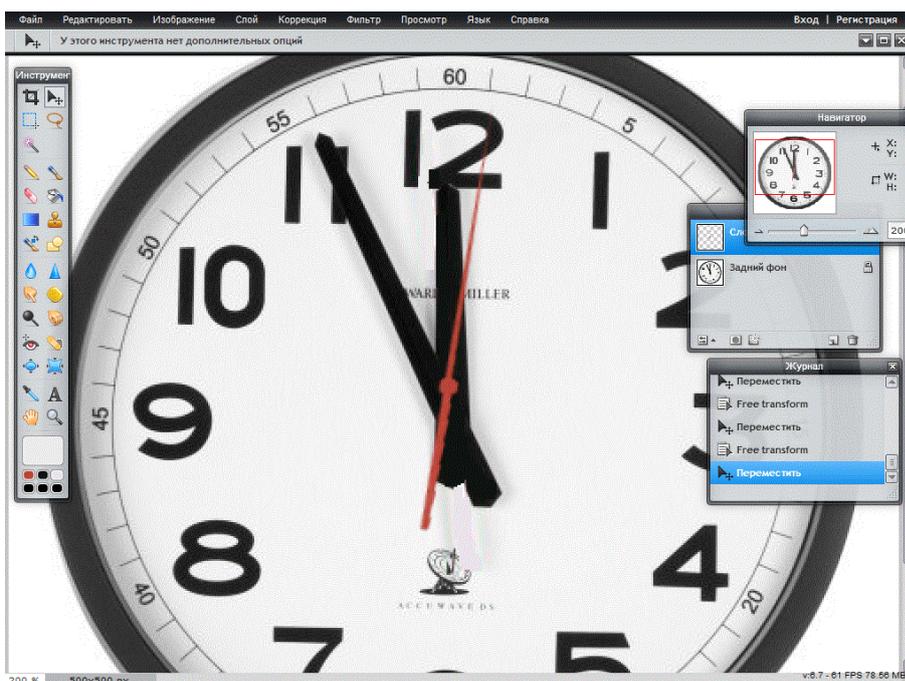


Рис. 15.7. Секундная стрелка показывает 1 секунду

11. Последовательно сохраните изображения часов с разными показаниями секундной стрелки 1, 2, ... , 5 секунд в папке с именем, соответствующим вашей фамилии (рис. 15.8).

ВНИМАНИЕ. Сохраните файлы с различными показаниями секундной стрелки для использования в работе № 16.



Рис. 15.8. Секундная стрелка показывает 5 секунд

12. Откройте <http://gifovina.ru/> (рис. 15.9) (С браузером Opera возможны проблемы с загрузкой).

На данном сайте находится бесплатная программа создания GIF-анимации онлайн.

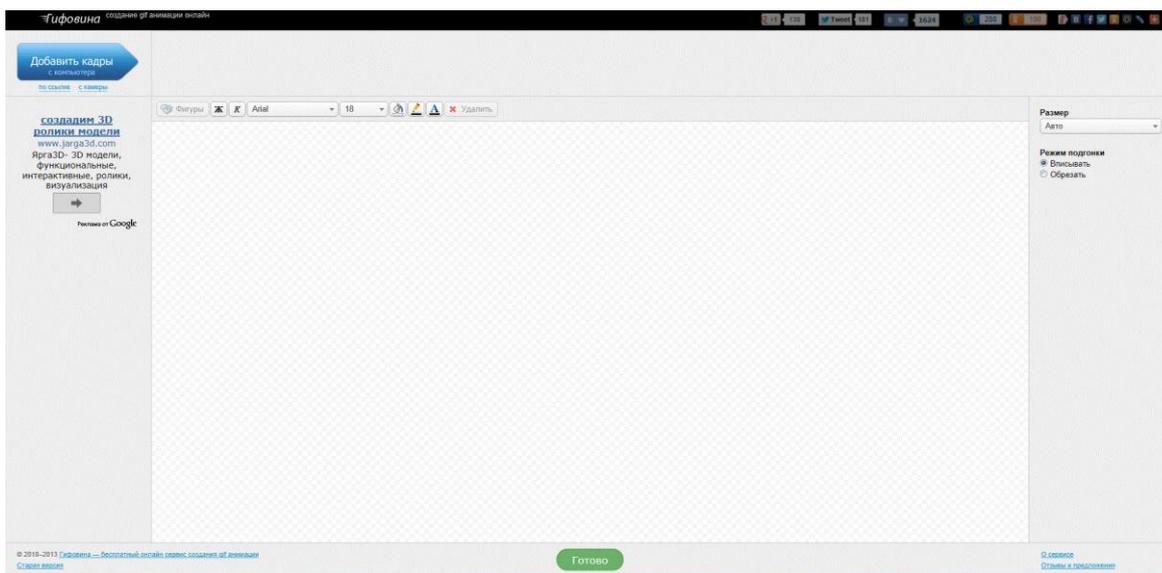


Рис. 15.9. Первая страница сервера <http://gifovina.ru>

13. В правом верхнем углу нажмите **Добавить кадры с компьютера**. Последовательно загрузите файлы с изображением часов с разными показаниями секундной стрелки 60, 1, 2, ... , 5 секунд (рис. 15.10).

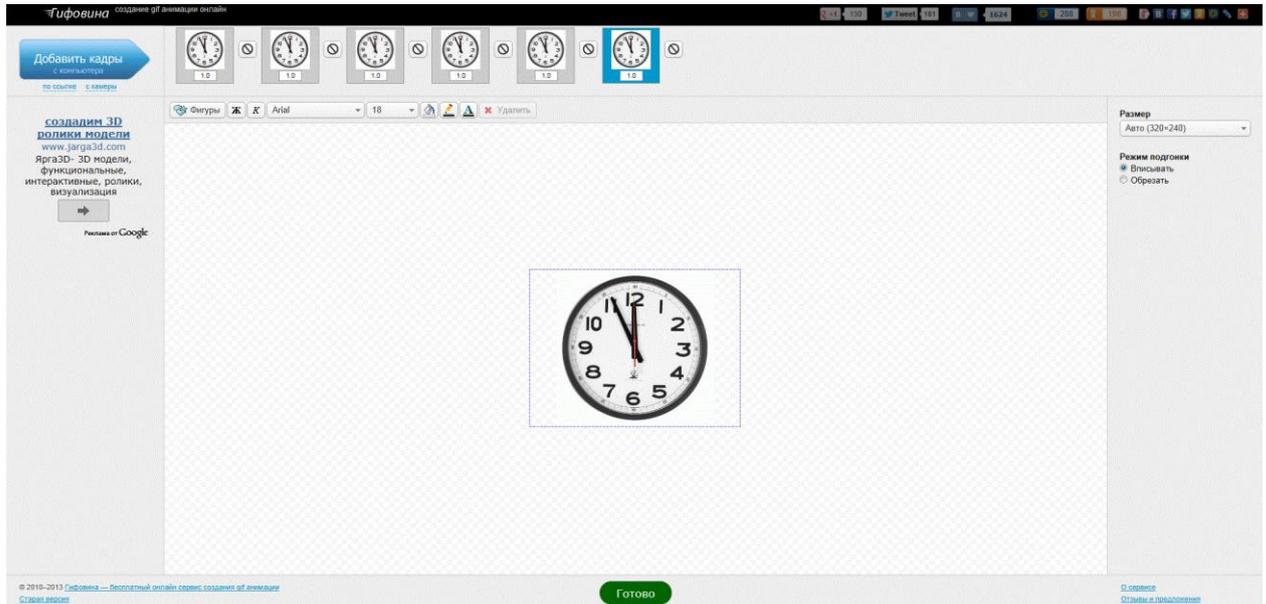


Рис. 15.10. Загруженные файлы со всеми показаниями часов

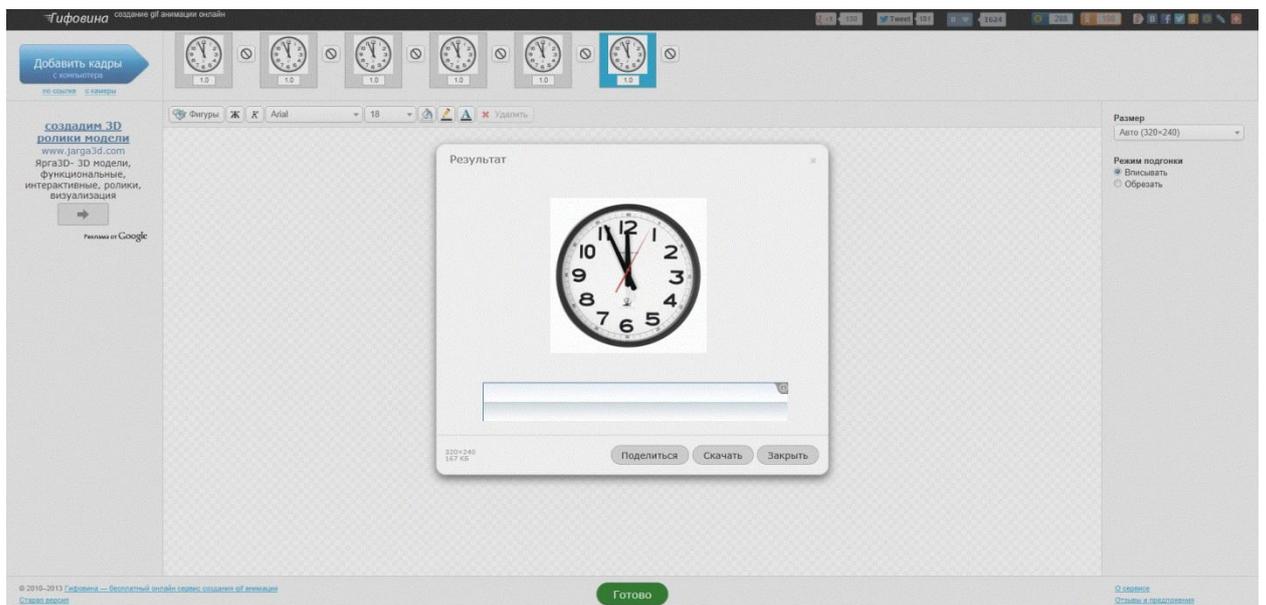


Рис. 15.11. Готовая анимация

14. В нижней части экрана нажмите **Готово**

15. Сохраните GIF-анимацию нажав кнопку **Скачать** (рис.15.11).

16. Просмотрите GIF-анимацию при помощи браузера.

17. Запишите файл с GIF-анимацией в Вашу папку и скопируйте ее в папку преподавателя (с номером Вашей группы).

18. Сохраните материалы данной работы для их использования в дальнейших работах.

Контрольные вопросы

1. К какому виду компьютерной графики относится формат GIF?
2. Что представляет собой GIF-анимация?
3. Какая GIF-анимация называется цикличной?
4. Какими параметрами управляют GIF-анимацией? Какие значения могут принимать эти параметры?
5. Из каких этапов состоит подготовка GIF-анимации?
6. На каких сайтах Интернета можно создать GIF-анимацию онлайн?
7. Перечислите многофункциональные растровые редакторы, поддерживающие создание GIF-анимации.
8. Перечислите специальные редакторы GIF-анимации.
9. Где в основном используется формат GIF?
10. Какой формат в настоящее время рассматривается как альтернатива формату GIF?

Литература

1. Секреты формата GIF, электронный ресурс:
<http://www.artlebedev.ru/kovodstvo/sections/8/>
2. Степанов, А. Н. Информатика : учебник для вузов / А. Н. Степанов. – 4-е изд. – СПб.: Питер, 2006. – 684 с.
3. Острейковский, В. А. Информатика : учебник для студентов технических и экономических специальностей вузов / В. А. Острейковский,. – 5-е изд. – М. : Высшая школа, 2009. – 511 с.

Работа № 16

СОЗДАНИЕ ВИДЕОФАЙЛА В ФОРМАТЕ MP4 ИЗ ГРАФИЧЕСКИХ ФАЙЛОВ В ФОРМАТЕ JPG

Цель работы: Освоение способов создания видеофайлов в формате MP4 из графических файлов в формате JPG.

Введение

MP4 – это сокращенный вариант формата MPEG-4 Part 14. Впервые он был запущен в 1998 г. Формат MPEG расшифровывается, как Motion Pictures Expert Group (Экспертная группа кинофильмов), которая осуществляет надлежащий контроль над ведением промышленных стандартов в сфере цифровой записи видео и звука.

MP4 формат функционирует подобно другому распространенному формату MP3, но считается довольно сложным. Технология формата MP3 совершила революционный скачок в области развития метода воспроизведения музыкальных файлов и треков. Формат MP4 делает то же самое для воспроизведения видео файлов.

MP4 представлен в виде контейнерного формата, позволяющего осуществлять необходимую комбинацию аудио-, видеотреков в одном единственном файле. Увеличение скорости подключения к глобальной сети Интернет помогло создать MPEG-4 – достаточно доступным для большого числа пользователей сети Интернет. С помощью данного формата посредством Интернет-соединения можно просматривать видео в режиме онлайн. Для этого следует воспользоваться портативными плеерами, персональным компьютером, ноутбуком, специальным устройством для воспроизведения видеофайлов. Самыми распространенными программами по воспроизведению MP4-файлов являются Media Player Classic, Windows Media Player и др.

Проигрыватель Windows Media Player – проигрыватель звуковых и видеофайлов для операционных систем семейства Windows. Версии Windows Media Player были также выпущены для Mac OS. В дополнение к проигрывателю, Windows Media Player содержит возможность копировать музыку с компакт-дисков, записывать диски в Audio CD формате или в виде дисков MP3 CD.

Windows Media Player производится корпорацией Microsoft и прилагается бесплатно к операционным системам семейств Windows.

Указания по выполнению работы

Выполнение работы начинается с создания файла отчета по форме, представленной в Приложении и заполнении ее личными данными студента (дисциплина, группа, ФИО). Именем файла с отчетом должна быть фамилия студента с его инициалами. К файлу отчета прилагается видеофайл в формате MP4. Его именем должна быть также фамилия студента с его инициалами.

После создания файла отчета можно приступать к выполнению работы.

ПРИМЕЧАНИЕ. В данной работе Вам понадобятся результаты работ 4 и 15.

1. Для выполнения работы Вам понадобятся графические файлы в формате JPG с различными показаниями секундной стрелки, которые были получены при помощи онлайн версии Adobe Photoshop в работе 15.

2. Откройте браузер. Перейдите по ссылке

<https://accounts.google.com/SignUp?continue=http%3A%2F%2Fwww.google.ru%2F&hl=ru>

На данной странице можно создать аккаунт Google, если у вас его нет.

3. Создайте аккаунт Google, если Вы еще не имеете его. Запомните Ваши учетные данные.

4. Перейдите по ссылке <http://www.youtube.com/editor>.

На данной странице находится видеоредактор YouTube. В правом верхнем углу нажмите **Войти**. Введите Ваши учетные данные аккаунта Google (рис.16.1).

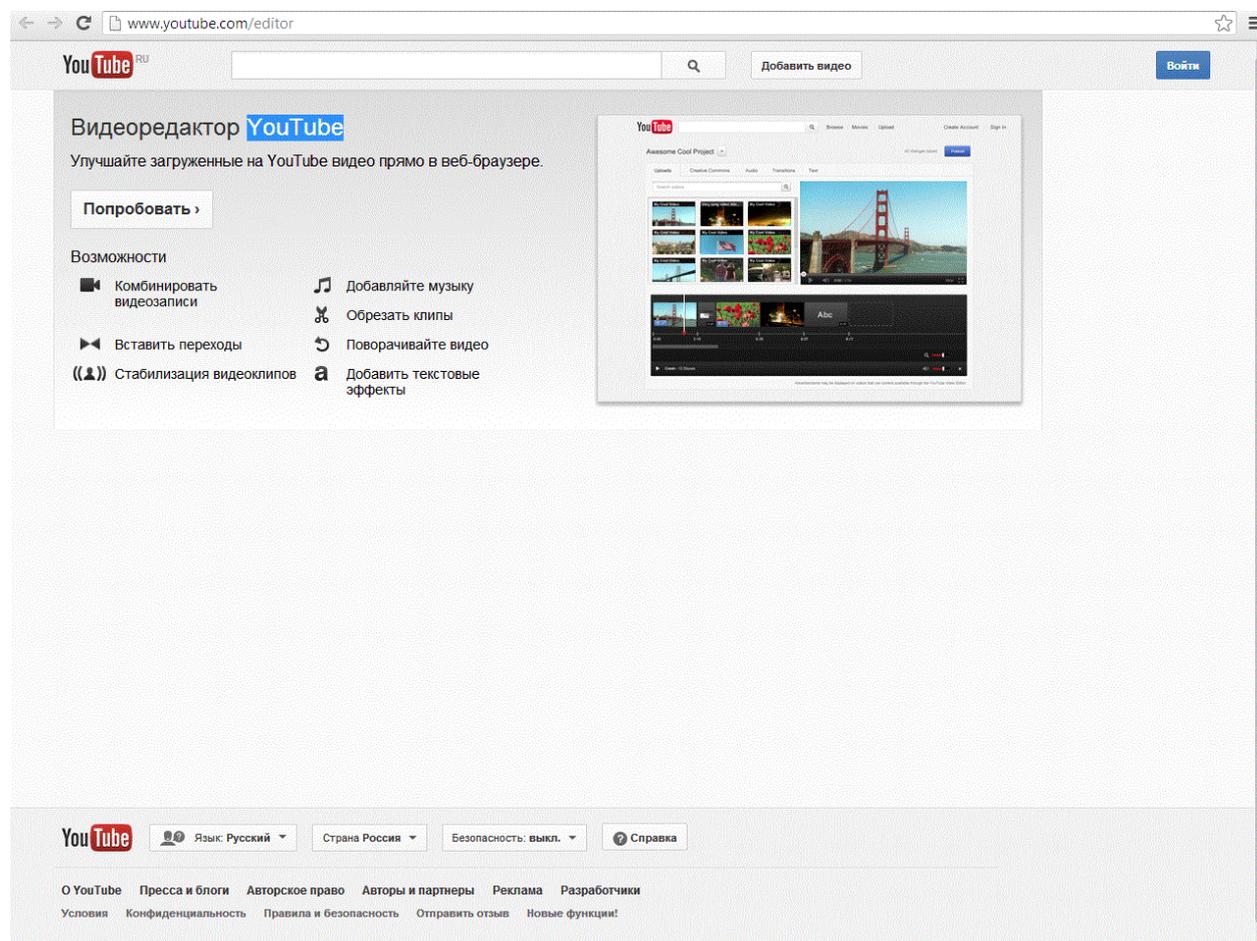


Рис. 16.1. Первая страница видеоредактора YouTube

5. Нажмите пиктограмму **Фотографии** . Затем нажмите **Добавить фото**. В появившемся окне выберите **Загрузка фотографий**. Затем нажмите **Выбрать фотографии с компьютера** (рис. 16.2).

6. Последовательно загрузите файлы с изображением часов с разными показаниями секундной стрелки 60, 1, 2, ... , 5 секунд, полученные при помощи онлайн версии Adobe Photoshop в работе 15.

7. Теперь при помощи правой кнопки мыши перетащите изображения на видеодорожку в нижней части экрана (рис.16.3).

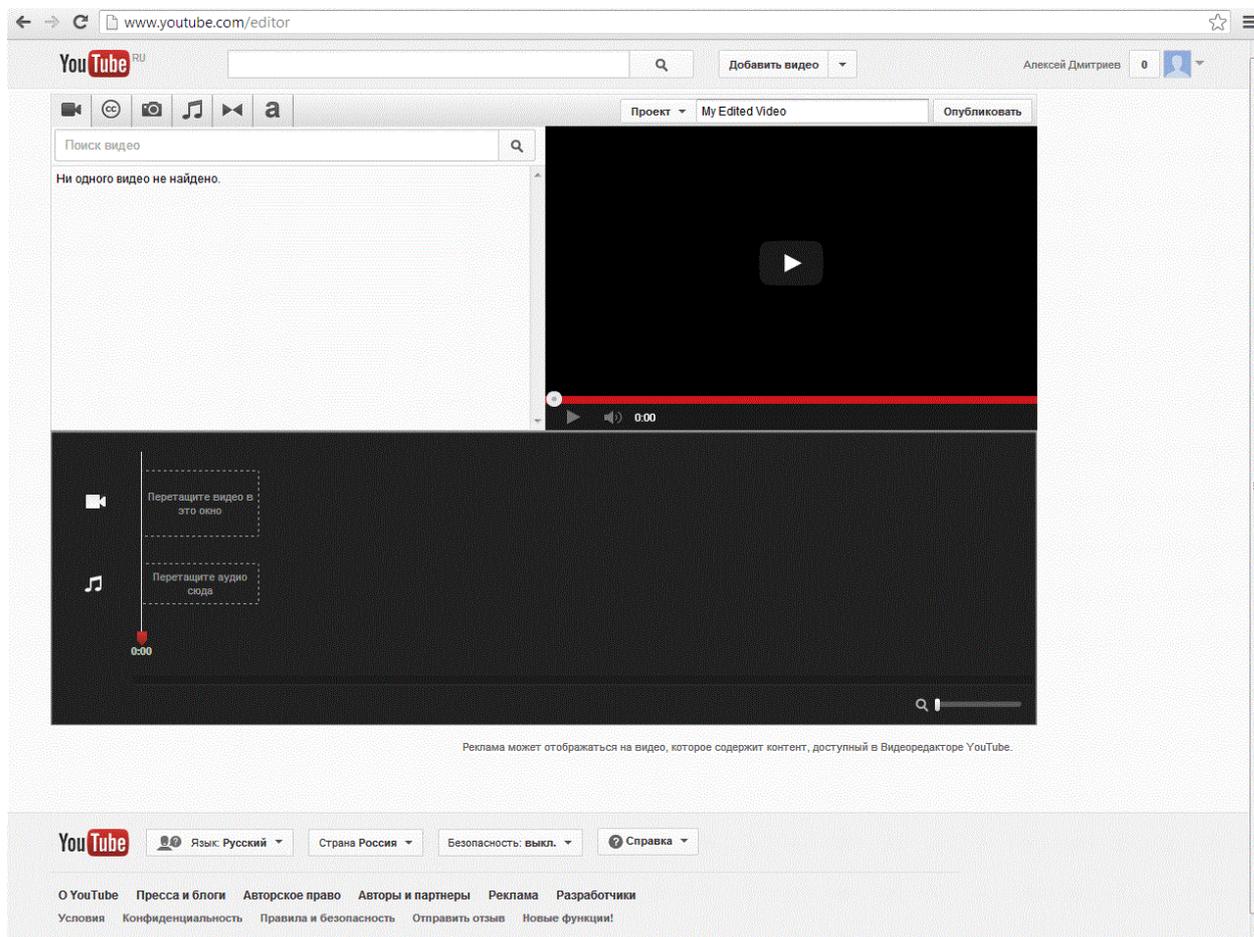


Рис. 16.2. Загрузка фотографий

8. Правой кнопкой мыши выделите первый кадр. Отпустив правую кнопку мыши, подведите курсор к правому краю кадра. Когда курсор изменит вид, захватите кадр нажатием правой кнопки мыши и уменьшите его ширину. Это приведет к тому, что продолжительность отображения кадра уменьшится (в правом нижнем углу каждого кадра указано время отображения, по умолчанию 5 секунд).

9. Уменьшите вышеописанным способом время отображения каждого кадра до 1 секунды.

10. Последним кадром добавьте текст. Для этого в левом верхнем углу нажмите **Текст** .

11. В появившемся меню образцов текста выберите **Текст в центре**. Захватив кадр с образцом текста правой кнопкой мыши, перетащите его на место последнего кадра видеодорожки.

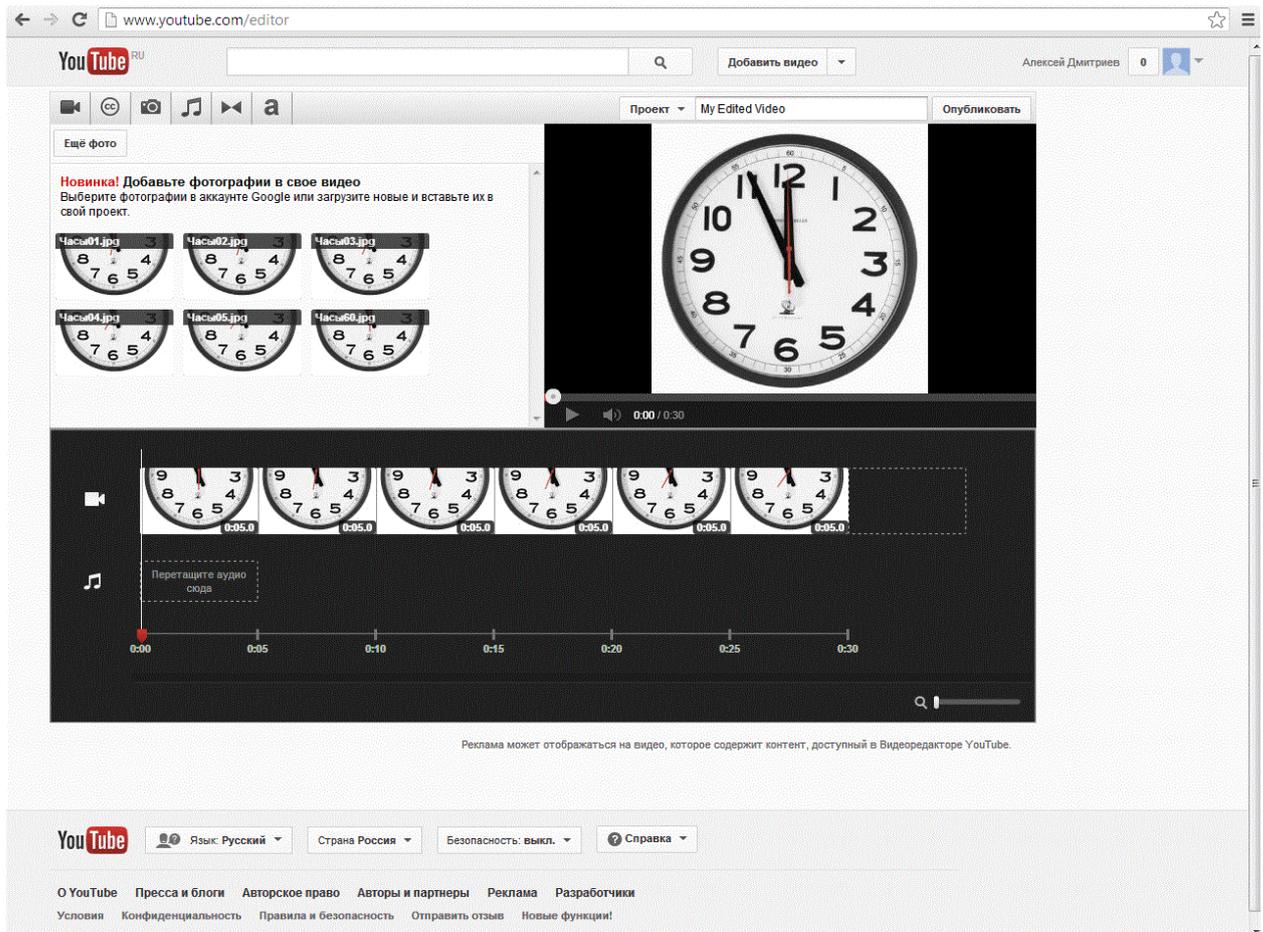


Рис. 16.3. Загрузка изображений на видеодорожку

12. В появившемся окне в соответствующее поле для ввода текста введите ваши личные данные (дисциплина, группа, ФИО). Затем нажмите **Сохранить** (или **Done**) (рис.16.4).

13. Далее в левом верхнем углу нажмите **Аудио** . В появившейся строке поиска введите Tick Tock Goes The Clock-Two Steps from Hel.

14. Найденный аудиоролик перетащите в аудиодорожку. Для этого нажмите правой кнопкой мыши на него и двигайте мышкой его к соответствующему месту в нижней части экрана.

15. Наведите курсор мыши на начало аудиодорожки и, захватив край правой кнопкой мыши, двигайте его в правую сторону. При этом будет происходить обрезака аудиодорожки. Обрежьте первые 1, 2 секунды аудиодорожки.

16. Аналогичным образом обрежьте правый край аудиодорожки таким образом, чтобы общая длина аудио ролика составила 8 секунд.

17. Видео смонтировано. Нажмите кнопку **Опубликовать** ближе к правому верхнему углу. Просмотрите ваш видеофайл.

18. Сохраните получившийся видеофайл. Для этого перейдите по ссылке <http://ru.savefrom.net/1-быстрый-способ-скачать-с-youtube/> и в строке поиска вве-

дите адрес вашего видео, предварительно скопировав его из адресной строки браузера, когда вы просматривали ваше видео.

19. Покажите ваш ролик преподавателю.

К файлу отчета прилагается видеофайл в формате MP4

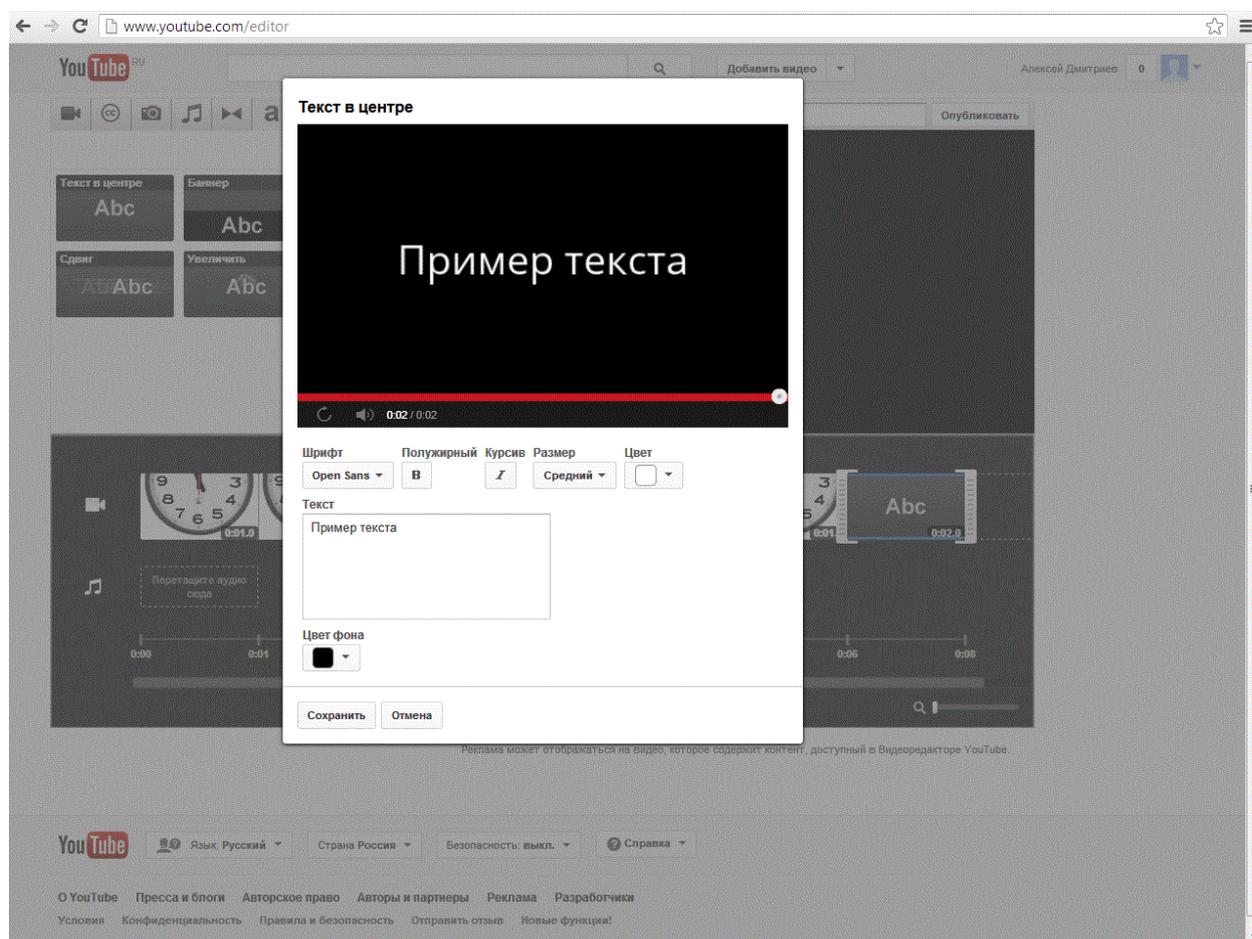


Рис. 16.4. Ввод личных данных

Контрольные вопросы

1. Что представляет собой формат MP4?
2. В каком году был выпущен формат MP4?
3. Какие преимущества формата позволили ему получить широкое распространение в сети Интернет?
4. Перечислите самые распространенные программы по воспроизведению MP4-файлов.
5. В каких операционных системах может работать проигрыватель Windows Media Player?
6. Перечислите функциональные возможности проигрывателя Windows Media Player.
7. Какой корпорацией производится проигрыватель Windows Media Player?

Литература

1. Официальный сайт проигрывателя Windows Media Player
<http://windows.microsoft.com/ru-ru/windows/windows-media-player>
2. Описание видео формата MP4 http://3gpfilm.org/video_mp4.html
3. Степанов, А. Н. Информатика : учебник для вузов / А. Н. Степанов. – 4-е изд. – СПб.: Питер, 2006. – 684 с.
4. Острейковский, В. А. Информатика : учебник для студентов технических и экономических специальностей вузов / В. А. Острейковский. – 5-е изд. – М. : Высшая школа, 2009. – 511 с.

Работа № 17

СОЗДАНИЕ GIF-АНИМАЦИИ ИЗ ВИДЕОФАЙЛА В ФОРМАТЕ WMV

Цель работы: Освоение способов создания анимированных изображений в формате GIF, на основе отдельных чередующихся кадров видеофайла в формате WMV.

Введение

GIF (англ. Graphics Interchange Format – формат для обмена изображениями) – формат растровых графических изображений, позволяющий создавать простую анимацию – чередующиеся кадры. Анимация представляет собой последовательность из нескольких статичных кадров и информацию о времени показа каждого кадра на экране.

Для создания последовательности растровых изображений и для их преобразования в GIF-анимации можно использовать многофункциональные растровые редакторы (Adobe Photoshop, CinePaint, cosmigo Pro Motion, PhotoLine) или специальные редакторы GIF-анимации (Gif Construction Set(tm) Professional, Animagic Gif, Ulead Gif Animator, WWW Gif Animator, Xara Webster, Gif Movie Gear). В Интернете также есть ресурсы, предлагающие создание GIF-анимации онлайн: <http://gifovina.ru/>, <http://minimultik.ru/>, <http://ru.toolson.net/GifAnimation>.

В качестве кадров GIF-анимации можно использовать отдельные кадры видеоизображения, например в формате WMV. Формат видео WMV (Windows Media Video) – формат, разработанный компанией Microsoft. Он содержит аудио-, видео- или смешанные записи. Видеофайлы с расширением WMV очень часто можно встретить в сети Интернет. Фильм в таком формате занимает сравнительно мало места на жестком диске, а воспроизведение не требует больших аппаратных мощностей.

Файлы, использующие расширение WMV, можно открыть практически любым проигрывателем персонального компьютера. Многие проигрыватели видео, такие как Media Player Classic (MPC) и т.д., уже имеют в комплекте все необходимые кодеки для качественного воспроизведения видео в формате WMV.

Media Player Classic (MPC) – мощный мультимедийный проигрыватель, впервые появившийся в июле 2002 г. Media Player Classic (MPC) имеет удобный интерфейс, сочетающийся с вполне современным, но без излишеств, набором функций. Media Player Classic (MPC) можно бесплатно загрузить с сайта <http://mpc.darkhost.ru/>.

Положительные особенности плеера: не требует инсталляции, позволяет произвольно перемещать и масштабировать изображение, имеет небольшой размер, вся программа представляет собой один исполняемый файл, позволяет сохранять кадры, накладывать субтитры на любой поддерживаемый формат, настраивает горячие клавиши и т.д.

К недостаткам можно отнести следующие: нет возможности настраивать яркость/контрастность, нет эквалайзера, нет многоязычного интерфейса (но русский язык есть).

ПРИМЕЧАНИЕ. В данной работе не рекомендуется использовать Internet Explorer. Рекомендуется – Chrome.

Указания по выполнению работы

Выполнение работы начинается с создания файла отчета по форме, представленной в Приложении, и заполнении ее личными данными студента (дисциплина, группа, ФИО). Именем файла с отчетом должна быть фамилия студента с его инициалами. К файлу отчета прилагается файл GIF-анимации. Именем файла GIF-анимации должна быть фамилия студента с его инициалами.

После создания файла отчета можно приступать к выполнению работы.

1. Запустите браузер;

2. Откройте страницу <http://mpc.darkhost.ru/> (рис.17.1).

Данный сайт посвящен бесплатному мощному мультимедийному проигрывателю **Media Player Classic (MPC)**, не требующему установки.

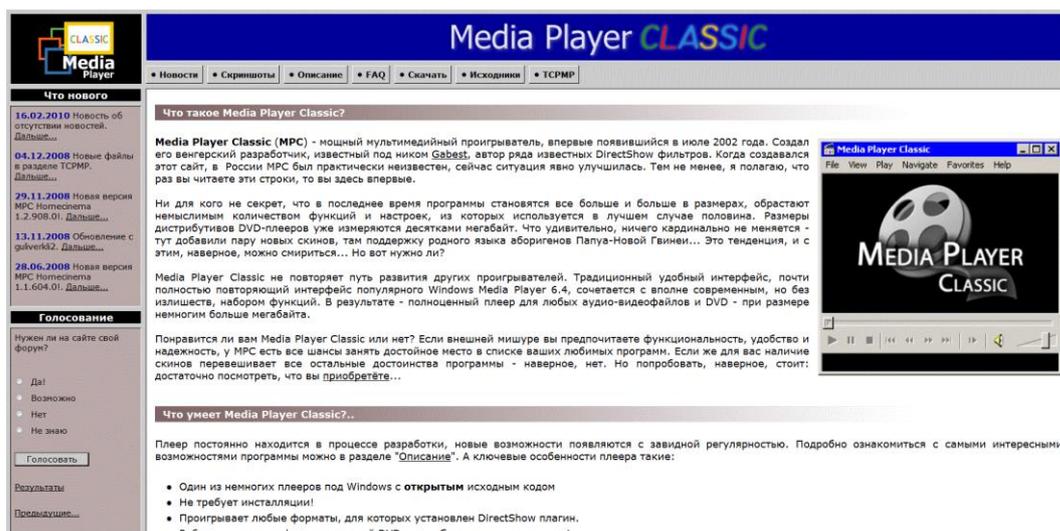


Рис. 17.1. Главная страница сервера <http://mpc.darkhost.ru>

2. Нажмите **Скачать**, затем **Стабильная версия 1.3.1249.0 (x86)** (рис. 17.2).

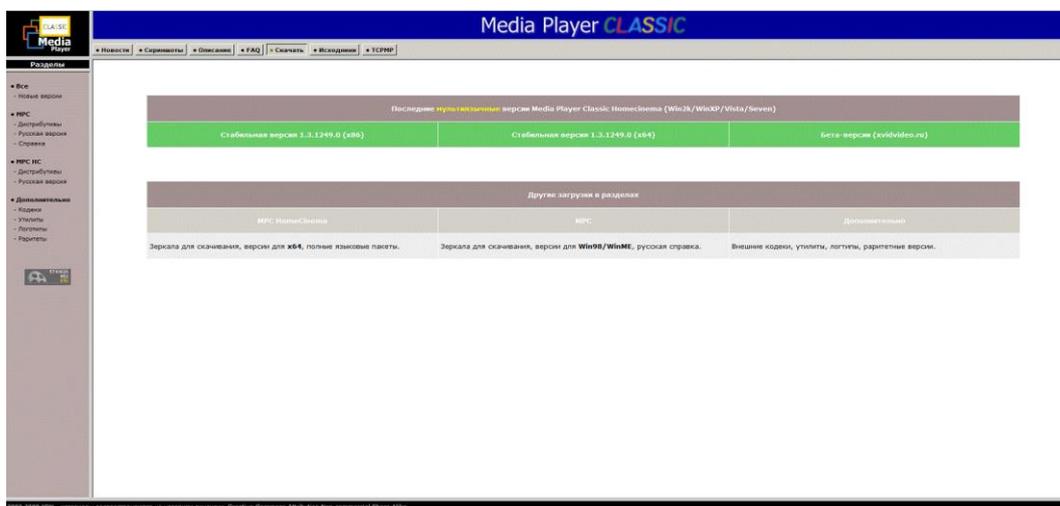


Рис. 17.2. Страница скачивания

4. Загрузите архив **MPC-Homecinema.1.3.1249.0.(x86).zip**

5. При помощи любого архиватора распакуйте архив. Запустите файл **mrc-hc.exe**

6. В панели инструментов нажмите **Файл**, затем **Открыть файл**. В появившемся окне нажмите **Выбрать**. Выберите файл **Живая природа.wmv (Wild-life.wmv)**. Путь к файлу **Локальный диск (C:) → Пользователи → Общие → Общие видео → Образцы видео**.

7. После открытия видеофайла нажмите на паузу . При помощи бегунка  верните видеоролик в начало. Перед Вами должен быть черный экран.

8. При помощи инструмента Шаг  перейдите к первому кадру с изображением (рис.17.3).



Рис. 17.3. Первый кадр файла Живая природа

9. В панели инструментов нажмите **Файл**, затем **Сохранить изображение**.

10. Вновь при помощи инструмента Шаг  перейдите к следующему кадру с изображением и сохраните его, как это описано в предыдущем пункте.

11. Таким способом сохраните первые 30 кадров из сюжета. Для ускорения работы рекомендуется использовать сочетание клавиш **Alt+I**, чтобы сохранять кадры.

12. Вернитесь в браузер. Откройте <http://gifovina.ru/>
На данном сайте находится бесплатная программа создания GIF-анимации онлайн.

13. В правом верхнем углу нажмите **Добавить кадры с компьютера**. Последовательно загрузите файлы – отдельные кадры видеофайла **Живая природа.wmv**. Обратите внимание, что одновременно можно загрузить не более 10 файлов.

14. Под каждым загруженным изображением имеется окно с обозначением длительности этого кадра (по умолчанию 1 секунда) (рис.17.4). Кликнув мышью на это окно, уменьшите время до 0.1 секунды. Прделайте это для каждого кадра.

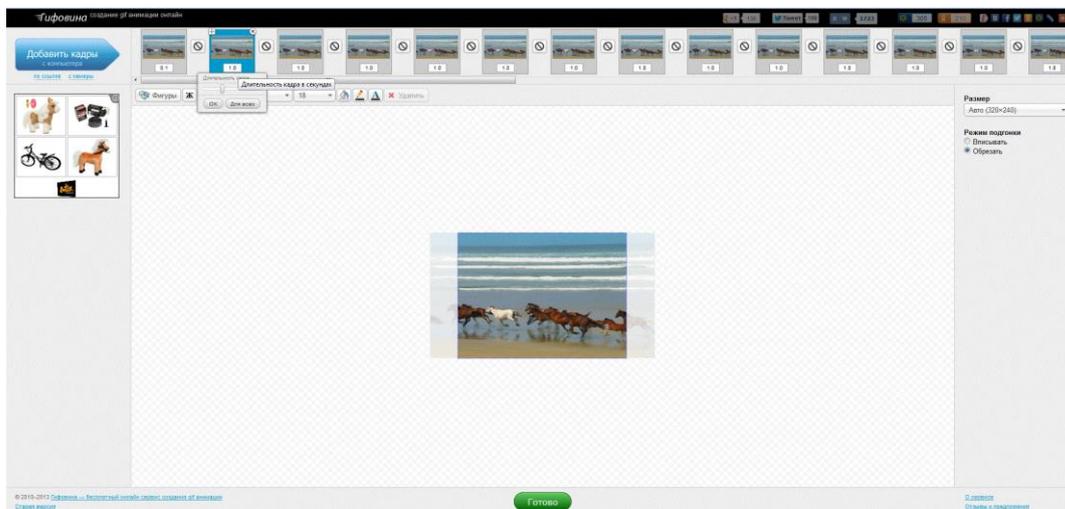


Рис. 17.4. Набор кадров с длительностью

15. В нижней части экрана нажмите **Готово**.
16. Сохраните GIF-анимацию нажав кнопку **Скачать**.
17. Просмотрите GIF-анимацию при помощи браузера.
18. Представьте выполненную работу преподавателю.

Контрольные вопросы

1. Какие видео-форматы существуют в настоящее время?
2. Перечислите многофункциональные мультимедийные плееры, поддерживающие видео-формат WMV.
3. Какая компания разработала формат видео WMV?
4. С какого сайта можно бесплатно загрузить Media Player Classic (MPC)?
5. Перечислите ключевые особенности плеера Media Player Classic.
6. Что можно указать в качестве недостатков плеера Media Player Classic?

Литература

1. Сайт, посвященный бесплатному мощному мультимедийному проигрывателю Media Player Classic (MPC) <http://mpc.darkhost.ru/>
2. Степанов, А. Н. Информатика : учебник для вузов / А. Н. Степанов. – 4-е изд. – СПб.: Питер, 2006. – 684 с.
3. Острейковский, В. А. Информатика : учебник для студентов технических и экономических специальностей вузов / В. А. Острейковский. – 5-е изд. – М.: Высшая школа, 2009. – 511 с.

ПРИЛОЖЕНИЯ

Приложение 1

Форма отчёта

ОТЧЕТ о выполнении практической работы по дисциплине			
Студент	Группа	Дата	Операционная система

Номер пункта работы	Наименование	Результат выполнения пункта работы (копии экрана)	
1			
2			
3			
4			

Варианты выполнения работы

№	Список серверов	№	Список серверов
1	http://www.omsu.ru Омский государственный университет http://dvfu.ru Дальневосточный федеральный университет http://rzd.ru сервер РЖД	11	http://www.sfu-kras.ru Сибирский федеральный университет http://www.ku.dk Университет Копенгагена http://do.ui-miit.ru Сервер системы ДО ЮИ МИИТ
2	http://www.kalmsu.ru Калмыцкий государственный университет http://www.anu.edu.au Австралийский национальный университет http://ciu.nstu.ru Новосибирский государственный технический университет	12	http://bmstu.ru МВТУ им. Н.Э. Баумана http://www.uwa.edu.au Университет Западной Австралии http://www.uoa.gr Афинский университет
3	http://www.asu.ru Алтайский государственный университет http://www.uha.fr Университет верхнего Эльзаса http://www.ntu.edu.tw Национальный университет Тайваня	13	http://www.omsu.ru Омский государственный университет http://english.bjmu.edu.cn Пекинский университет http://www.uniroma1.it Римский университет
4	http://www.nsu.ru Новосибирский государственный университет http://www.du.ac.in Университет Дели http://cu.edu.eg Каирский университет	14	http://rzd.ru сервер РЖД http://cu.edu.eg Каирский университет http://www.unipi.gr Университет Пирея
5	http://www.pgups.ru Петербургский государственный университет путей сообщения http://english.tau.ac.il Университет Тель-Авива	15	http://dvfu.ru Дальневосточный федеральный университет http://www.uowdubai.ac.ae Вуллонгонгский университет ОАЭ

	http://www.uj.ac.za Йоханнесбургский университет		
6	http://www.kamchatgtu.ru Камчатский государственный техический университет http://www.uu.se Упсала университет, Швеция http://www.ufrj.br Федеральный университет Рио- де-Жанейро	16	http://www.aspu.ru Астраханский государственный университет http://www.ntu.edu.sg Технологический университет Сингапура http://www.isu.ru Иркутский государственный университет
7	http://www.fessler.ru Дальневосточная государствен- ная научная библиотека http://www.anu.edu.au Австралийский национальный университет http://www.tusur.ru Томский государственный уни- верситет систем управления и радиоэлектроники	17	http://www.ugatu.ac.ru Уфимский государственный авиационный технический уни- верситет http://www.ntu.edu.tw Национальный университет Тай- ваня http://www.astanauniver.kz/ Университет «Астана»
8	http://sfedu.ru Южный федеральный универ- ситет http://www.sorbonne.fr Сорбонна или Парижский уни- верситет http://www.unal.edu.co Колумбийский Национальный университет	18	http://www.titech.ac.jp/ Токийский Технологический ин- ститут http://www.univ.kiev.ua Киевский национальный универ- ситет http://www.sssu.ru Южно-Российский государ- ственный университет экономи- ки и сервиса
9	http://www.yasu.ru Северо-Восточный федераль- ный университет http://www.nus.edu.ws Национальный университет Самоа http://www.sibadi.org Сибирская автодорожная ака- демия	19	http://www.bsby.by Белорусский государственный университет http://www.cuni.cz Карлов университет, Прага
10	http://www.tsu.ru/ Томский государственный уни- верситет	20	http://www.kamchatgtu.ru Камчатский гос. тех. универси- тет

http://www.cityu.edu.hk/ Университет Гонконга http://www.bsu.edu.ru Национальный исследовательский Белгородский государственный университет	http://cu.edu.eg Каирский университет
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------

Порядок выбора варианта определяется преподавателем:

- по номеру рабочего места
- по списку группы
- назначением персональных вариантов.

*Приложение 3
(к работе № 8)*

Вариант	Изображения
1	
2	
3	
4	
5	
6	
7	
8	

9	
10	
11	
12	
13	
14	
15	

Справочник основных тегов HTML

Основные теги	Описание
<code><html></html></code>	Определяет тип документа как html документ (web страницу).
<code><head></head></code>	Служебная часть документа, не отображаемая в браузере, тут размещаются заголовок, описание, ключевые слова и т.д.
<code><body></body></code>	Между этими тегами располагается непосредственно сама web страница, которую Вы видите на экране, могут размещаться и другие типы данных.
<code><meta name="allow-search" content=""></code>	Определяет способ сканирования данных для роботов поисковых систем.
<code><meta name="robots" content=""></code>	Указывается информация для поисковых роботов.
<code><meta name="author" content=""></code>	В данном теге можно указать авторство.
<code><meta name="keywords" content=""></code>	Для обеспечения работы с поисковыми серверами указываются ключевые слова (через запятую).
<code><base href="www.?"></code>	Применяется для не полностью прописанных ссылок, указывает начальный адрес в интернете.
<code><meta name="description" content=""></code>	Краткое описание web страницы, будет показано в результатах поиска поисковиками.
<code><title></title></code>	Тег для заголовка страницы.
<code><body bgcolor=""></code>	Определяет цвет фона web страницы.
<code><body text=""></code>	Определяет цвет текста web страницы.
<code><body link=""></code>	Задаёт цвет гиперссылок в нормальном состоянии.
<code><body vlink=""></code>	Задаёт цвет уже просмотренных гиперссылок.
<code><body alink=""></code>	Задаёт цвет выделенной гиперссылки.
<code><h1></h1></code>	Тег заголовка, создаёт самый крупный заголовок, цифра 1 может меняться в пределах от 1 до 6. Шесть будет самый мелкий заголовок.
<code></code>	Делает текст полужирным.
<code><i></i></code>	Форматирует текст в наклонный.

<code><tt></tt></code>	Имитация стиля печатной машинки.										
<code><kbd></kbd></code>	Имитация стиля печатной машинки.										
<code><var></var></code>	Название переменных отображается курсивом.										
<code><cite></cite></code>	Выделение цитат курсивом.										
<code><address></address></code>	Создаёт абзац с текстом выделенным курсивом.										
<code></code>	Курсив (воспринимается поисковиками как выделение).										
<code></code>	Полужирное форматирование текста (воспринимается поисковиками как особо сильное выделение)										
<code></code>	Устанавливает размер шрифта, значение от 1 до 7.										
<code></code>	Задаёт цвет текста.										
Оформление гиперссылок в html											
<code></code>	Создаёт гиперссылку, имеются в виду как внутренние, так и внешние гиперссылки.										
<code></code>	Указывает, в каком окне открывать гиперссылку <table border="1"> <thead> <tr> <th><i>Параметр</i></th> <th><i>Значение</i></th> </tr> </thead> <tbody> <tr> <td><code>_Blank</code></td> <td>Открыть в новом окне</td> </tr> <tr> <td><code>_Parent</code></td> <td>Открыть в текущем окне</td> </tr> <tr> <td><code>_Self</code></td> <td>Открыть в окне без использования действующих фреймов</td> </tr> <tr> <td><code>_Top</code></td> <td>Игнорировать фреймы</td> </tr> </tbody> </table>	<i>Параметр</i>	<i>Значение</i>	<code>_Blank</code>	Открыть в новом окне	<code>_Parent</code>	Открыть в текущем окне	<code>_Self</code>	Открыть в окне без использования действующих фреймов	<code>_Top</code>	Игнорировать фреймы
<i>Параметр</i>	<i>Значение</i>										
<code>_Blank</code>	Открыть в новом окне										
<code>_Parent</code>	Открыть в текущем окне										
<code>_Self</code>	Открыть в окне без использования действующих фреймов										
<code>_Top</code>	Игнорировать фреймы										
<code></code>	Гиперссылка для отправления сообщения по электронной почте.										
<code></code>	Создаёт гиперссылку на метку в текущей странице.										
<code></code>	Ставит метку для гиперссылки на текущей странице.										
<code></code>	Гиперссылка на метку другой страницы.										
Форматирование текста											
<code><p></p></code>	Создаёт новый параграф.										
<code><p align="?"></p></code>	Выравнивает параграф относительно одной из сторон документа, значения: left, right, justify или center.										

<code><nobr></code>	Запрещает перевод строки.
<code><wbr></code>	Указывает, где разбивать строку для переноса при необходимости.
<code>
</code>	Вставляет перевод строки.
<code><blockquote></blockquote></code>	Создаёт отступы с обеих сторон текста.
<code><dl></dl></code>	Создаёт список определений.
<code><dt></code>	Определяет каждый из терминов списка.
<code><dd></code>	Описывает каждое определение.
<code></code>	Форматирует текст в пронумерованный список.
<code></code>	Определяет каждый элемент списка и присваивает порядковый номер.
<code></code>	Тег для создания нумерованного списка
<code></code>	Отмечает каждую новую позицию в списке кружком или квадратиком.
<code><div align="?"></div></code>	Данный тег используется для форматирования больших блоков текста HTML документа, также используется в таблицах стилей.
Графические элементы на странице	
<code></code>	Вставляет изображение на страницу.
<code></code>	Форматирует положение изображения в документе, может иметь значения: left, right, center; bottom, top, middle.
<code></code>	Устанавливает толщину рамки вокруг изображения
<code></code>	Устанавливает поля сверху и снизу.
<code></code>	Устанавливает поля слева и справа.
<code></code>	Создаёт всплывающую подсказку.
<code><hr></code>	Добавляет горизонтальную линию.
<code><hr size="?"></code>	Указывает толщину линии.
<code><hr width="?"></code>	Указывает ширину линии в пикселях или процентах.
<code><hr noshade></code>	Линия без тени.
<code><hr color="?"></code>	Определяет цвет линии.
Создание таблиц	
<code><table></table></code>	Тег, создающий таблицу.
<code><tr></tr></code>	Задаёт строку в таблице.

<code><td></td></code>	Задаёт отдельную ячейку в таблице.
<code><th></th></code>	Задаёт заголовок таблицы (обычная ячейка с форматированием по центру и полужирным текстом)
Атрибуты таблицы	
<code><caption></caption></code>	Назначает подпись таблицы
<code><table border="#"></code>	Определяет толщину рамки.
<code><table cellspacing="#"></code>	Определяет расстояние между ячейками.
<code><table cellpadding="#"></code>	Устанавливает поля для текстового содержимого ячейки.
<code><table width="#"></code>	Устанавливает ширину таблицы. (Значение параметра может быть в пикселях или процентах).
<code><table height="#"></code>	Устанавливает высоту таблицы (значение параметра может быть в пикселях или процентах).
<code><tr align="#"></code> или <code><td align="#"></code>	Определяет горизонтальное выравнивание положения ячеек в таблице, может иметь значения: left, center, right.
<code><tr valign="#"></code> или <code><td valign="#"></code>	Определяет вертикальное выравнивание положения ячеек в таблице, может иметь значения: top, middle, или bottom.
<code><td colspan="#"></code>	Указывает количество столбцов, объединенных в одной ячейке.
<code><td rowspan="#"></code>	Указывает количество строк, объединенных в одной ячейке.
<code><td nowrap></code>	Не позволяет программе просмотра делать перевод строки в ячейке таблицы.
<code><td width="#"></code>	Задаёт ширину ячейки таблицы в пикселях или процентах.
<code><td height="#"></code>	Задаёт высоту ячейки таблицы в пикселях или процентах.
Фреймы в документах html	
<code><frameset></frameset></code>	Предшествует тегу <code><body></code> на фреймовой странице.
<code><frameset rows="value,value"></code>	Определяет строки в таблице фреймов, значение высоты которых определяется количеством пикселей или в процентах.

<code><frameset cols="value,value"></code>	Определяет столбцы в таблице фреймов, значение ширины которых определяется количеством пикселей или в процентах.
<code><frame></code>	Задаёт единичный фрейм или область в таблице фреймов.
<code><noframes></noframes></code>	Задаёт информацию, которую увидит посетитель в случаях, когда показ фрейма невозможен.
<i>Используемые атрибуты фреймов html</i>	
<code><frame src="URL"></code>	Привязка web страницы к окну фрейма.
<code><frame name="name"></code>	Назначает фрейму имя для улучшения маршрутизации информации между фреймами и страницами.
<code><frame marginwidth="#"></code>	Отступы по горизонтали внутри фреймового окна.
<code><frame marginheight="#"></code>	Отступы по вертикали внутри фреймового окна.
<code><frame scrolling=VALUE></code>	Создаёт линейку прокрутки фрейма; значение value может быть "yes", "no" или "auto". По умолчанию – auto.
<code><frame noresize></code>	Запрещает изменение размеров фреймового окна посетителем сайта.
<i>Ифрейм и его атрибуты</i>	
<code><iframe></iframe></code>	Создаёт контейнер, который может содержать любые элементы. Другие элементы страницы будут обтекать данный контейнер.
<code><iframe src="URL"></code>	Привязывает web страницу к окну фрейма.
<code><iframe name="name"></code>	Назначает фрейму имя для улучшения маршрутизации информации.
<code><iframe vspase="#"></code>	Задаёт отступы по горизонтали снаружи фрейма.
<code><iframe hspase="#"></code>	Задаёт отступы по вертикали снаружи фрейма.
<code><iframe marginwidth="#"></code>	Задаёт поля по горизонтали внутри фрейма.
<code><iframe marginheight="#"></code>	Задаёт поля по вертикали внутри фрейма.
<code><iframe scrolling=VALUE></code>	Создаёт линейку прокрутки фрейма; значение value может быть "yes," "no," или "auto". По умолчанию – auto.

<code><iframe width="#"></code>	Задаёт ширину ифрейма
<code><iframe height="#"></code>	Задаёт высоту ифрейма;
<code><iframe title="?"></code>	Закрепляет текст контекстной помощи.
Теги относящиеся к формам	
<code><form></form></code>	Тег для создания формы.
<code><select multiple name="NAME" size="?"> </select></code>	Создаёт меню со скроллингом. Атрибут size определяет численность пунктов в меню, видимых сразу без использования линейки прокрутки.
<code><option></code>	Задаёт каждый элемент меню по отдельности.
<code><select name="NAME"></select></code>	Создаёт ниспадающее меню
<code><textarea name="NAME" cols=40 rows=8></textarea></code>	Размещает редактируемое окно для ввода текста. Атрибуты columns – задаёт значение ширины окна, а rows соответственно высоту.
<code><input type="checkbox" name="NAME"></code>	Размещает элемент checkbox.
<code><input type="radio" name="NAME" value="x"></code>	Размещает элемент типа радио-кнопки.
<code><input type="text" name="foo" size=20></code>	Размещает строку для ввода текста. Атрибутом Size задаётся длина в символах.
<code><input type="submit" value="NAME"></code>	Размещает на странице кнопку "Отправить"
<code><input type="image" border="0" name="NAME" src="name.gif"></code>	Размещает на странице "Отправить" с использованием графического изображения.
<code><input type="reset"></code>	Размещает на странице "Очистить".

Таблица CER (Character Entity Reference)

Символ	Название	Ввод	Подстановка	Пример
	Незарядный пробел	Alt 0160	 	А. С. Пушкин
«	Левая французская кавычка	Alt 0171	«	«Война и мир»
»	Правая французская кавычка	Alt 0187	»	«Война и мир»
"	Опознающая кавычка	Alt 0132	"	„Война и мир“
”	Закрывающая кавычка	Alt 0147	”	„Война и мир“
—	Длинное тире	Alt 0151	—	Знамя — сила!
-	Короткое тире	Alt 0150	–	1895-1943
-	Дефис	ߚ	ߚ	Веб-сайт
-	Незарядный дефис	ߛ	ߛ	Веб-сайт
-	Цифровая черточка	ߜ	ߜ	Тел. 111-22-33
-	Малый перенос	Alt 0173	©	
-	Минус	−	−	10 - 5 = 5
±	Плюс-минус	Alt 0177	±	10.0 ± 0.5 м
×	Знак умножения	Alt 0215*	×	4 × 7 = 28
÷	Знак деления	Alt 0247*	÷	28 ÷ 7 = 4
½	Дробь	Alt 0188*	½	28 + ½
<	Знак меньше	<	<	5 < 10
>	Знак больше	>	>	10 > 5
°	Знак градуса	Alt 0176	°	30°
,	Знак ударения	ߛ	ߛ	Большая часть
’	Апостроф	Alt 0146	’	Д’Арман
§	Знак параграфа	Alt 0167	§	§ 5
&	Амперсанд	&	&	
©	Знак охраны авторского права	Alt 0169	©	
®	Символ зарегистрированного товарного знака	Alt 0174	®	Сoca-Cola®
←	Стрелка влево	<arr;	<arr;	
↑	Стрелка вверх	↑	↑	
→	Стрелка направо	→	→	
↓	Стрелка вниз	↓	↓	
€	Евро	Alt 0128*	€	
…	Неготовые	Alt 0133	…	Конц...

СОДЕРЖАНИЕ

Работа № 1	4
Работа № 2	8
Работа № 3	12
Работа № 4	18
Работа № 5	27
Работа № 6	31
Работа № 7	39
Работа № 8	41
Работа № 9	48
Работа № 10	52
Работа № 11	58
Работа № 12	62
Работа № 13	71
Работа № 14	77
Работа № 15	84
Работа № 16	92
Работа № 17	98
Приложения	102

**ДМИТРИЕВ А.И.
ЛОБАЧЕВ С.Л.
МАЛЫГИН О.А.**

КОМПЬЮТЕРНЫЕ СЕТИ, МУЛЬТИМЕДИАТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ

***Методические указания
по выполнению практических
(лабораторных) работ***

Изд. заказ 25

Подписано в печать 24.09.2013

Усл. печ. л. 4,3

Бумага офсетная

Печать трафаретная

Уч.-изд. л. 7,1

Тираж 50 экз.

Цена договорная

Формат 60×90¹/₁₆

127994, Москва, ул. Образцова, д. 9, стр. 9.