

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА  
(РОСАВИАЦИЯ)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)

---

Кафедра вычислительных машин, комплексов, систем и сетей

Н.И. Романчева

# ИНФОРМАТИКА

## ОБРАБОТКА МУЛЬТИМЕДИЙНОЙ ИНФОРМАЦИИ

**Учебно-методическое пособие**  
по выполнению лабораторных работ № 7, 8

*для студентов  
направления 09.03.01 (бакалавриат)  
очной формы обучения*

Москва  
ИД Академии Жуковского  
2021

УДК 004.9  
ББК 6Ф6.5  
Р69

Рецензент:

*Павлова Л.В.* – заместитель директора по эксплуатации отдела ИС ПАО «Аэрофлот»

**Романчева Н.И.**

Р69 Информатика. Обработка мультимедийной информации [Текст] : учебно-методическое пособие по выполнению лабораторных работ № 7, 8 / Н.И. Романчева. – М.: ИД Академии Жуковского, 2021. – 32 с.

Данное учебно-методическое пособие издается в соответствии с рабочей программой учебной дисциплины «Информатика. Обработка мультимедийной информации» по учебному плану для студентов направления 09.03.01 (бакалавриат) очной формы обучения.

Рассмотрено и одобрено на заседаниях кафедры 25.02.2021 г. и методического совета 25.02.2021 г.

**УДК 004.9**  
**ББК 6Ф6.5**

*В авторской редакции*

Подписано в печать 14.05.2021 г.  
Формат 60x84/16 Печ. л. 2 Усл. печ. л. 1,86  
Заказ № 733/0330-УМП02 Тираж 30 экз.

Московский государственный технический университет ГА  
125993, Москва, Кронштадтский бульвар, д. 20

Издательский дом Академии имени Н. Е. Жуковского  
125167, Москва, 8-го Марта 4-я ул., д. 6А  
Тел.: (495) 973-45-68  
E-mail: zakaz@itsbook.ru

© Московский государственный технический  
университет гражданской авиации, 2021

## СО Д Е Р Ж А Н И Е

1. Основные требования и порядок выполнения лабораторных работ	4
2. Лабораторная работа № 7. РАБОТА С ГРАФИЧЕСКИМ ПАКЕТОМ VISIO PROFESSIONAL	7
2.1. Цель работы	7
2.2. Перечень компетенций, формируемых в ходе выполнения лабораторной работы	7
2.3. Задание на выполнение работы	7
2.4. Основные теоретические сведения	8
2.5. Порядок выполнения задания	18
2.6. Вопросы к защите лабораторной работы	19
2.7. Список рекомендуемой литературы	19
3. Лабораторная работа № 8. СОЗДАНИЕ HTML-ДОКУМЕНТОВ	20
3.1. Цель работы	20
3.2. Перечень компетенций, формируемых в ходе выполнения лабораторной работы	20
3.3. Задание на выполнение работы	20
3.4. Основные теоретические сведения	21
3.5. Основные приемы работы	28
3.6. Вопросы к защите лабораторной работы	32
3.7. Список рекомендуемой литературы	32

## 1. ОСНОВНЫЕ ТРЕБОВАНИЯ И ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Настоящее учебно-методическое пособие предназначено для студентов направления подготовки Информатика и вычислительная техника (бакалавриат), выполняющих лабораторные работы по дисциплине Информатика в соответствии с ФГОС3++. В данное учебно-методическое пособие включены материалы для выполнения лабораторных работ № 7,8.

Продолжительность каждой лабораторной работы - 4 часа.

Целью проведения лабораторных работ является закрепление основных теоретических положений, изложенных в лекциях по дисциплине Информатика.

В процессе выполнения лабораторных работ студенты должны получить практические навыки обработки мультимедийной информации на примере разработки пользовательских фигур и изображений, их редактирования в графическом редакторе Visio Professional, создания простого и составного HTML-документа со встроенной графикой.

Лабораторная работа состоит из следующих этапов:

- 1) домашняя подготовка;
- 2) выполнение работы на компьютере в соответствии с заданием;
- 3) сдача выполненной работы преподавателю на персональном компьютере;
- 4) распечатка результатов работы на принтере;
- 5) оформление отчета;
- 6) защита лабораторной работы.

В процессе домашней подготовки студент:

- изучает лекционный материал,
- материал по теме лабораторной работы данного учебно-методического пособия и дополнительной литературы;
- знакомится с заданием на выполнение лабораторной работы;
- готовит проект отчета по выполнению лабораторной работы.

**ВЫПОЛНЕНИЕ** лабораторной работы проводится во время занятий в лаборатории кафедры ВМКСС МГТУГА в присутствии преподавателя в соответствии с расписанием, утвержденным проректором по УМР МГТУ ГА. В процессе выполнения лабораторной работы студент последовательно выполняет задание. По завершению работы - демонстрирует преподавателю результаты.

**СДАЧА РАБОТЫ** преподавателю на персональном компьютере заключается в демонстрации выполненной работы и выполнении непосредственно при преподавателе индивидуального дополнительного задания.

**ОТЧЕТ** по каждой лабораторной работе должен содержать: название работы; цель лабораторной работы; задание на выполнение лабораторной

работы; краткие комментарии по выполнению лабораторной работы; распечатки файлов результатов или диаграмм, подписанные преподавателем.

**ЗАЩИТА** лабораторной работы преподавателю проводится по контрольным вопросам и при наличии оформленного отчета (распечатки результатов выполнения лабораторной работы должны быть аккуратно приклеены). После защиты лабораторной работы преподавателем делается соответствующая запись на отчете студента.

**РЕЙТИНГОВЫЙ КОНТРОЛЬ** по лабораторным работам производится при их сдаче во время лабораторных занятий. Перед выполнением лабораторной работы производится экспресс-опрос для определения готовности студентов к выполнению работы (знания теоретического материала, целей работы и т.д.).

#### 1) Допуск к ЛР

Допуск к выполнению ЛР происходит в виде устного опроса (при условии наличия у студента печатной версии титульного листа отчета по лабораторной работе).

Студент, не прошедший устный опрос, к выполнению лабораторной работы не допускается.

#### 2) Выполнение и защита ЛР

Минимальная оценка выставляется за выполненную и защищенную лабораторную работу, а дополнительными баллами оценивается качество выполненной лабораторной работы и полнота знаний, показанная студентами при ее защите. Лабораторная работа считается сделанной, если она выполнена полностью в соответствии с заданием.

#### *Сроки сдачи лабораторных работ*

Номер лабораторной работы	Лабораторная работа	Срок сдачи*
7	Лабораторная работа №7	Лабораторная работа №7
8	Лабораторная работа №8	Лабораторная работа №8

\*Срок сдачи лабораторной работы – в соответствии с расписанием проведения лабораторных работ.

#### 3) Отчет по ЛР

Отчет по лабораторной работе представляется в печатном виде в формате, предусмотренном шаблоном отчета по лабораторной работе.

#### 4) Защита ЛР

Отчет не может быть принят и подлежит доработке в случае:

- небрежного выполнения,
- низкого качества представленного материала (неполное решение, не указаны единицы измерения),
- отсутствия необходимых разделов отчета,
- отсутствия необходимого графического материала,

- отсутствия выводов по работе;
  - некорректной обработки результатов выполнения лабораторной работы, и
- т.п.

## **2. ЛАБОРАТОРНАЯ РАБОТА №7 РАБОТА С ГРАФИЧЕСКИМ ПАКЕТОМ VISIO PROFESSIONAL**

### **2.1. Цель работы**

Целью данной работы является получение практических навыков работы с пакетом Microsoft Visio Professional, принципами разработки пользовательских фигур и изображений, их редактирования.

### **2.2. Перечень компетенций, формируемых в ходе выполнения лабораторной работы**

Общепрофессиональная компетенция ОПК-2:

**ОПК-2** - Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.

**ИД-2<sub>ОПК-2</sub>** - современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности, в части:

ОПК-2.2.5 - умеет классифицировать объекты графического редактора выполнить разработку и настройку пользовательских фигур; создавать высокоточный графический объект, собственный трафарет;

ОПК-2.3.3 - владеет основными приемами работы с графическим редактором.

### **2.3. Задание на выполнение работы**

- 1) Изучить главные элементы интерфейса редактора Visio Professional и способы их настройки.
- 2) Выполнить разработку и настройку пользовательских фигур: рисование, модификация готовых фигур, вращение, работа с объединенными или сгруппированными фигурами, изменение отдельных цветов или цветовых схем, добавление к тексту.
- 3) Изучить основные возможности управления взаимным расположением фигур: переместить точку отсчета; выполнить параметры настройки сетки; вставить точку привязки на лист; выполнить автоматическое управление фигурами - выравнивание, соединение, порядок следования.
- 4) Разработать пользовательское изображение, используя схемы, приведенные в лекции "Основы компьютерных коммуникаций".
- 5) Выполнить редактирование изображения: изменить формат, преобразовать формы, вставить текстовые блоки, выполнить их объединение и группировку.
- 6) Создать трафарет графических объектов Visio Professional для предметной области в соответствии с предложенным вариантом.
- 7) Создать блок-схему алгоритма решения квадратного уравнения.
- 8) Создать схему организации циклических вычислений для варианта с верхним окончанием.

- 9) Создать графическую диаграмму, используя данные лабораторной работы №6.
- 10) Сохранить созданные файлы в графическом формате GIF.
- 11) Распечатать подготовленные файлы.

#### **2.4. Основные теоретические сведения**

При цифровом кодировании дискретной информации применяют Microsoft Visio Professional - мощный графический редактор, предназначенный для быстрого и эффективного создания графических изображений любой сложности. С помощью встроенных шаблонов, трафаретов и стандартных модулей можно создавать как простейшие слайды или схемы, так и сложные чертежи или организационные диаграммы.

Приложение Visio Professional является не совсем традиционным графическим редактором: с одной стороны, оно обладает богатыми возможностями для построения сложных чертежей и графических изображений, а с другой стороны - имеет множество полезных и удобных надстроек, обеспечивающих, например, доступ к организационным диаграммам (organization chart) или построение обычных и трехмерных графиков. В программе существует большая библиотека изображений (Shape) с помощью которых можно создавать также наглядные схемы (локальной сети, электрической цепи, блок-схемы), фасад дома, карты местности, и многое другое.

Для работы с системой необходимо знание базовых определений:

- *фигура (Shape)* — базовый графический элемент, из комбинации которых создается схема (drawing);
- *трафарет (Stencil)* — совокупность базовых элементов, объединяемых общностью предметной области;
- *стиль (Style)* — совокупность форматов (размер шрифта, толщина и цвет линий, заполнение), имеющая имя и сохраняемая со стандартом или схемой;
- *шаблон (Template)* — файл, содержащий стили и установки для некоторого вида схем и открывающий соответствующие трафареты. Создание новой схемы начинается с вызова необходимого трафарета или стандарта.

##### **2.4.1 Элементы интерфейса**

Для запуска Visio щелкните мышью на значке запускаемой программы. После того как программа запущена, на экране появляется окно приложения, которое состоит из следующих элементов:

- Строка заголовка (содержит название приложения и имя редактируемого файла).
- Панель меню (открывает доступ к командам, предназначенным для работы с документами).



- Панели инструментов (на них располагаются кнопки и раскрывающиеся списки для быстрого доступа к командам).

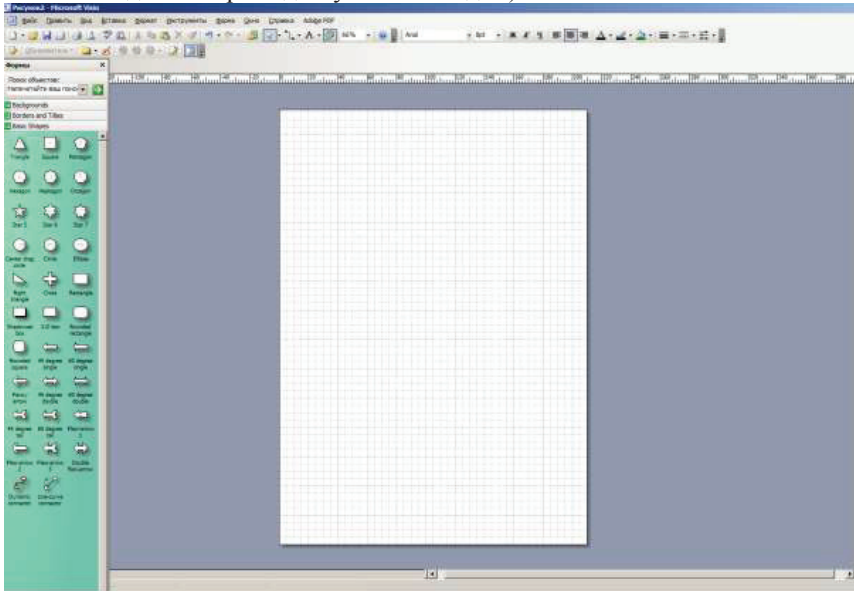


Рисунок 2.1 - Окно приложения Visio

- Лист рисунка (drawing page) - окно редактирования, в поле которого можно вставить фигуры, графические элементы и текст.
- Трафарет - специальные панели, содержащие различные мастера, графические и вспомогательные элементы.
- Строка состояния (отображает текущее состояние документа).

#### 2.4.2 Трафареты

Перечислим некоторые из трафаретов, сохраняя для удобства работы с системой их английские названия:

Basic - правильные многоугольники с числом сторон от 3 до 8, круг, эллипс, овал; разнообразные по форме и направлению стрелки, в том числе криволинейные; соединители фигур;

Block diagram - три комплекта геометрических фигур, стрелок и соединителей в большом числе вариантов, с оттенением и объемной перспективой;

Border - элементы фигурных рамок и заполнителей для орнаментов («греческий», египетский), звезды, шахматная клетка, волны, зигзаг и т.п.);

Callout - средства выносных пояснений: «баллоны», ярлыки, оттиски печатей, окна, кнопки, «вспышки»;

Smart - элементы диаграмм (столбчатых, объемных, секторных) и графиков, в частности координатные оси, экспоненциальные и «нормальные» кривые, вспомогательные ярлыки;

Clipart - разнообразные мини-картинки: люди, средства связи, вычислительная техника, транспортные средства и др.;

Connectors - расширенный набор соединительных элементов;

Electrical - инверторы, схемы И, ИЛИ, диод, триод, «земля», батарея, резистор, конденсатор, сигнал типа, меандра, часы;

Flowchart (два набора) - элементы схем алгоритмов (все стандартизованные обозначения внутри- и межстраничные соединители, средства разветвления и соединения потоков, базы данных);

Forms - рамки для текста, шаблоны для таблиц и графиков (в частности, заготовки для логарифмической сетки), текстовые блоки для набора шрифтом Arial в 8,10 и 18 pt;

Maps - обширная группа наборов географических карт, изображающих контуры США, стран, регионов и континентов, а также обозначений на картах и схемах, и знаков дорожного движения;

Network - элементы сетей связи и компьютерных сетей устройства ЭВМ; малые и большие ЭВМ разных типов, включая Gateway, типичные сетевые структуры; антенна спутниковой связи и сам спутник;

Office Layout - обозначения офисного оборудования;

Project Timeline - средства сетевого планирования (PERT-диаграммы и графики Ганта);

Symbols - разнообразные денежные, дорожные, предостерегающие и информационные знаки.

На рис. 2.2 представлено меню выбора трафаретов.

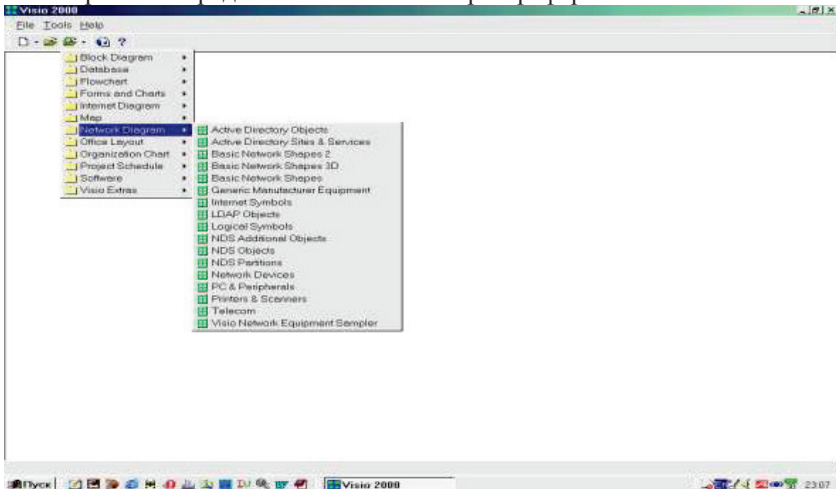
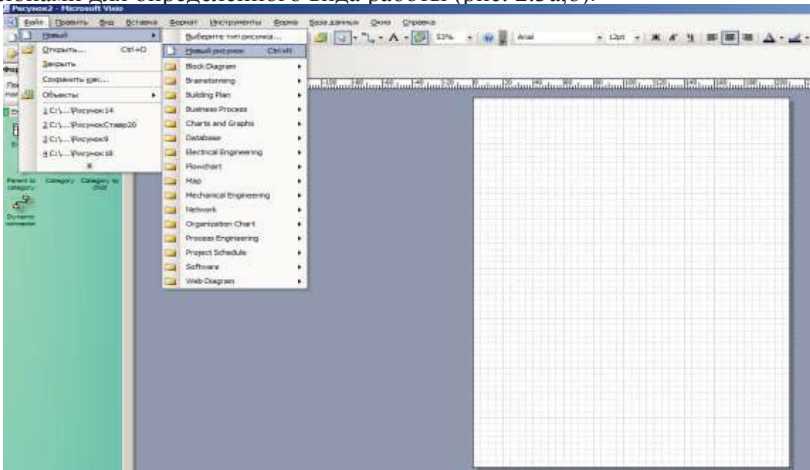


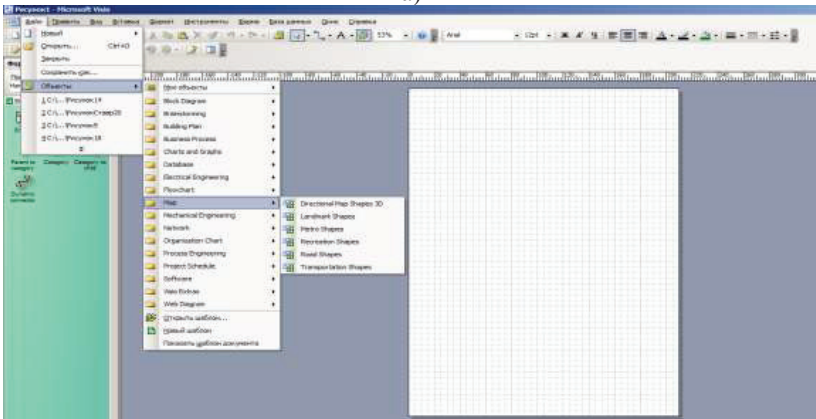
Рисунок 2.2 - Меню выбора трафаретов

### 2.4.3 Создание нового документа

Создать новый файл можно через меню в самой программе *File\New\Drawing* (сочетание клавиш CTRL-N) или воспользоваться готовыми шаблонами для определенного вида работы (рис. 2.3а,б).



а)



б)

Рисунок 2.3 - Пример создания файла с использованием шаблонов

В редактор включены новые мастера, которые помогают создавать или редактировать сложные диаграммы типа сетевых графиков проектов (рис 2.4). В редактируемую работу можно добавить и другие шаблоны через меню *File/Stencils* (*Выбрать необходимый шаблон*). Ввод информации осуществляется с помощью перетаскивания “образа” (Shape) из шаблона в рабочую зону, после чего размер “образа” можно изменить. Соединять разные “образы” в одну цепь можно с помощью Интеллектуальных соединительных

линий (иконка *Connector Tool* в панели инструментов). Интеллектуальные соединительные линии делают все что могут, чтобы избежать пересечений с другими объектами, а если это оказывается невозможным, перепрыгивают через пересекаемые линии. При нажатии правой кнопки мыши на “образ” можно задать дополнительные параметры.

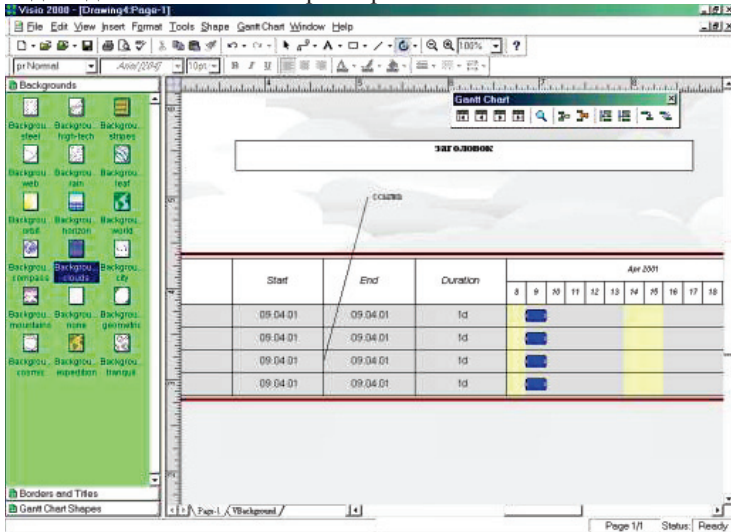


Рисунок 2.4 - Пример редактирования файла с использованием шаблонов

В программе имеется стандартный набор для ввода текста, изменения шрифта, закраски. Для того чтобы ввести текст, необходимо в панели инструментов нажать на иконку *Text Tool* и указать курсором редактируемое место. В поле *Size* устанавливается размер шрифта.

В программе можно вставлять графические объекты как из AutoCAD (выбрать команду *AutoCAD Drawing* в меню *Insert*), так и из стандартного набора Windows ClipArt (выбрать команду *ClipArt* в меню *Insert*). С помощью команды *Protect Document* в меню *Tools* можно защитить свою работу паролем.

В редакторе Visio предусмотрена возможность сохранения документа. Сохранить документ можно через меню *File>Save as*, при этом указав имя файла и нужный формат файла.

#### 2.4.4 Создание пользовательских фигур

В Visio под "фигурой" подразумевается любой геометрический объект, который может быть выделен одним щелчком левой кнопки мыши. Это может быть элементарная фигура - линия, дуга, сплайн (несколько объединенных дуг), или сложная замкнутая фигура, состоящая из любого числа последовательно соединенных простых элементов.

В Visio различается три типа фигур: одномерные (1-D shapes), двумерные (2-D shapes) и псевдотрехмерные (3-D shapes). Трехмерные получаются путем

добавления тени к обычным двумерным фигурам, при этом явное представление третьей координаты в Visio отсутствует - все это позволяет говорить о псевдотрехмерности. Главный признак фигуры - наличие маркеров при ее выделении (рис.2.5).

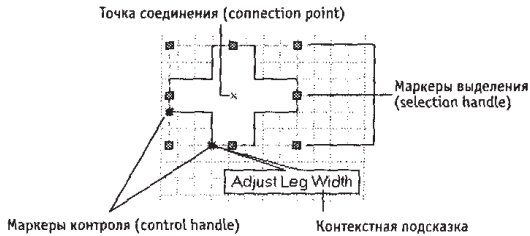


Рис. 2.5 - Типы маркеров

Основными являются маркеры выделения (selection handle). Каждая двумерная фигура имеет восемь маркеров выделения, которые образуют прямоугольник, ограничивающий фигуру. В дальнейшем, используя термин «маркер», будем подразумевать маркер выделения. Кроме визуального ограничения, маркеры играют важную роль в изменении размеров фигуры. Вместе с маркерами выделения в некоторых сложных фигурах могут использоваться маркеры контроля (control handle), которые предназначены для изменения внутренних размеров фигуры. Например, у представленной на рис.2.6 фигуры может быть изменена не только ее высота и ширина, но и ширина.

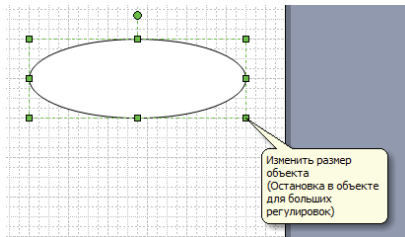


Рисунок 2.6 – Пример маркера контроля

Для более четкого различия маркеров выделения и маркеров контроля последние имеют контекстные подсказки, которые содержат информацию о способе их применения (рис.2.7). Несмотря на различное обозначение, маркеры начала и конца линии тождественны. С помощью этих маркеров можно изменить длину линии или ее наклон. Центральный маркер предназначен для решения других задач. Например, вращение линии с помощью специальных инструментов или формул всегда выполняется относительно центра, то есть центрального маркера.

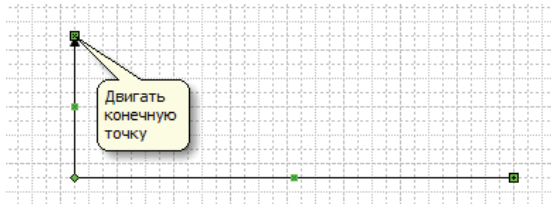


Рисунок 2.7 – Пример контекстной подсказки

Поскольку линия является частным случаем прямоугольника, то центральный маркер можно «растянуть», установив на него указатель мыши, который превратится в двунаправленную стрелку, и, нажав левую кнопку мыши, потянуть в его сторону. При этом линия сместится вниз или вверх на половину «растянутого» расстояния, а остальные ее параметры останутся неизменными. У линии нет явного обозначения точек соединения, поскольку ими являются ее начало и конец (рис. 2.8).

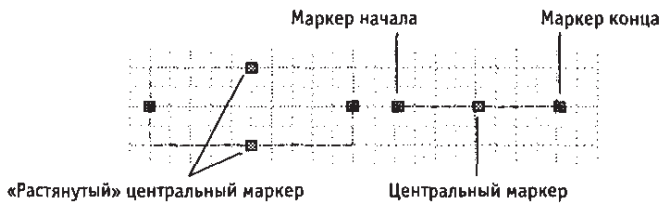


Рисунок 2.8 - Маркеры линий

**Одномерная фигура** представляет собой отрезок прямой (линию), который определяется только одним параметром - длиной. При выделении линия определяется двумя конечными маркерами (endpoint): в начале (обозначается крестиком X) и в конце (обозначается знаком плюс +), и маркером, обозначающим центр.

Другим примером одномерной фигуры является стрелка (рис. 2.9).



Рисунок 2.9 - Пример одномерной фигуры- стрелка

Как видно из рис.2.9, все стрелки имеют только два маркера выделения. Центральный маркер, определяющий толщину, у некоторых стрелок преобразован в несколько маркеров управления, которые позволяют устанавливать пользовательские размеры для ширины и углов наклона указателя и оперения. Таким образом, явным признаком одномерной фигуры является наличие маркеров начала и конца.

**Двумерные фигуры** представляют собой основу фигур Visio, т.е. все остальные фигуры являются их следствием. Например, линия, так же как и куб,

является частным случаем прямоугольника - двумерной фигуры. В отличие от трехмерных фигур, двумерные могут быть не замкнуты. Простейшей двумерной фигурой является фигура, образованная путем объединения двух линий. Несмотря на кажущуюся простоту, эта фигура является полноправным представителем своего типа, так как для ее описания уже недостаточно одного параметра - длины одной из линий, необходим еще один параметр - длина другой линии.

Выделенная двумерная фигура всегда обрамляется восемью маркерами, с помощью которых можно осуществлять управление ее размерами. Боковые маркеры позволяют изменить соответствующие линейные размеры - высоту или ширину, а с помощью угловых маркеров все размеры изменяются пропорционально.

**Трехмерные фигуры**, представленные в трафаретах Visio - это фигуры, созданные на основе двумерных путем добавления к ним тени. Это единственное явное отличие двумерной фигуры от трехмерной. В отличие от простого наложения тени, в трехмерных фигурах имеется маркер управления (рис. 2.10), с помощью которого можно установить угол наклона и размер тени. Тень в трехмерных фигурах очерчивается контурными линиями.



Рисунок 2.10 - Маркер управления

Для упрощения работы с фигурами разных типов в Visio предусмотрена возможность **преобразования фигур** в фигуры других типов. Для преобразования одномерной фигуры в двумерную или наоборот необходимо выполнить следующую операцию:

- Выделить фигуру, тип которой нужно преобразовать.
- Выбрать команду Format/Behavior и в открывшемся окне диалога перейти на вкладку Behavior.
- В зависимости от задачи в разделе Interaction Style установить переключатель в нужное положение: Line (1-dimensional) или Box (2-dimensional).
- Нажать кнопку ОК.

После преобразования изменятся свойства фигуры и вид маркеров выделения. Например, стрелка, которая относится к одномерному типу, может быть преобразована в фигуру, относящуюся к двумерной. После этого число ограничивающих ее маркеров станет равным восьми.

При преобразовании прямоугольника в одномерный тип у него останется два маркера выделения - маркеры начала и конца, и пара маркеров, определяющих ширину фигуры. Теперь с помощью любого из

маркеров выделения прямоугольник можно повернуть на произвольный угол относительно противоположного маркера (нужно заметить, что в обычных условиях поворот двумерной фигуры осуществляется с помощью специальных инструментов).

Для **создания тени** используется специальный инструмент *Shadow Color*, который находится на панели инструментов *Format Shape*. Этот инструмент позволяет назначить фигуре тень, которая будет иметь указанный цвет. Для выбора цвета тени в раскрывающемся списке *Color* или *Pattern* укажите необходимые пункты.

#### **2.4.5. Макет**

Основной концепцией *Visio* является возможность точной привязки объектов. Графический редактор *Visio* создан специально для разработки высокоточных рисунков и изображений. В *Visio* имеются специальные элементы, позволяющие максимально упростить работу по выстраиванию и привязке фигур друг к другу и определенным точкам листа.

##### **2.4.5.1 Линейка**

Линейка (*ruler*) - это стандартный элемент интерфейса, который используется в большинстве графических приложений. В *Visio* линейки располагаются слева и вверху окна редактирования. Шкала линейки, используемая в текущий момент, зависит от масштаба изображения и от единиц размерности. По умолчанию нулевая отметка (точка отсчета) каждой из линеек соответствует левому нижнему углу листа.

При перемещении фигур по листу на линейках появляются три метки, указывающие на положение сторон и центра прямоугольника, который описывает перемещаемую фигуру или группу.

Для перемещения точки отсчета, необходимо настроить данный параметр в специальном окне диалога *Ruler and Grid*. Для его открытия выберите команду *Tools/Ruler and Grid*. Окно включает два раздела: *Ruler* - настройка линеек, и *Grid*- настройка координатной сетки. Для изменения положения точки начала отсчета установите необходимые параметры в полях *Ruler zero*: горизонтальное смещение относительно левой границы листа в поле *Horizontal* и вертикальное смещение относительно нижнего края - в поле *Vertical*. Координаты точки отсчета всегда указываются относительно левого нижнего угла листа.

##### **2.4.5.2 Сетка**

Сеткой (*grid*) называются непечатаемые горизонтальные и вертикальные штриховые линии, нанесенные на лист и отображаемые через равные промежутки. Сетка позволяет визуально оценить размер фигур, вставленных в лист, и расстояние между ними.

Точка пересечения вертикальной и горизонтальной сетки называется узлом сетки, или просто узлом. Сетка настроена так, чтобы при увеличении или уменьшении масштаба частота линий сетки изменялась автоматически. Кроме видимых узлов, существуют располагающиеся между ними невидимые. Число невидимых узлов зависит от ширины шага между видимыми узлами и от текущего



масштаба и, как правило, равно двум или четырем. Лист Visio имеет уникальное строение. В отличие от других редакторов, в которых рабочее пространство является непрерывным, лист Visio является дискретным, т.е. все важные точки фигуры всегда находятся в узлах сетки. Это можно заметить при выполнении небольших смещений фигуры - она "скачет" от одного узла к другому. Это кажущееся неудобство легко преобразуется в очень удобный инструмент привязки: достаточно увеличить масштаб, и те позиции, которые прежде "проскакивали", станут доступными. Это значительно упрощается визуальное соединение фигур. Например, достаточно подвести одну вершину фигуры к другой, как она автоматически будет расположена в том же узле, что и первая.

В зависимости от стоящей задачи параметры настройки могут быть изменены пользователем. Для этого в Visio имеется окно диалога *Ruler and Grid*, которое становится доступным при выборе команды *Tools /Ruler and Grid*.

В разделе Grid можно настроить три параметра: интервал *Grid Spacing*, минимальный интервал *Minimum Spacing* и точку начала координат сетки *Grid Origin*. Выбор интервала позволяет определить, как далеко друг от друга будут располагаться линии координатной сетки. Начало координатной сетки по умолчанию находится в левом нижнем углу листа.

#### **2.4.5.3 Точка привязки**

В Visio имеются специальные инструменты, определяющие привязку и позиционирование фигур на листе. К ним можно отнести непечатаемую *точку привязки (guide point)*, которая позволяет выделить один узел на листе. После выделения этот узел приобретет специфическое свойство - все вершины или точки соединения фигуры, попадающие в область его видимости, автоматически помещаются в узел. Точка привязки имеет синий контур, а в случае выделения закрашивается изумрудным цветом. Геометрический размер точки и ее области видимости всегда остается постоянным независимо от изменения масштаба.

Для вставки точки привязки на лист используется специальный инструмент *Guide Point*, который находится в левом верхнем углу в месте пересечения вертикальной и горизонтальной линеек. В случае если линейки скрыты, этот инструмент становится недоступным.

#### **2.3.5.4 Автоматическое управление фигурами**

Основные команды, управляющие положением фигуры, находятся в меню *Tools* и на панели инструментов *Action*. В Visio имеется два инструмента для автоматического выравнивания фигур: *Align Shapes* - позволяет выполнить вертикальное или горизонтальное выравнивание; *Distribute Shapes* - позволяет выровнять расстояния между фигурами. Окно диалога инструмента *Align Shapes* можно открыть, выбрав команду *Tools/Align Shapes*. В открывшемся окне диалога представлены три способа выравнивания по горизонтали и три по вертикали.

Процесс выравнивания состоит в следующем: выделить фигуры, которые нужно выровнять; открыть окно диалога *Align Shapes*; выбрать нужный способ выравнивания и нажать кнопку *OK*. Если установить флажок *Create guide and*

*glue shapes to it*, то после выравнивания на лист будет помещена линия или точка привязки, которые будут связаны с соответствующим способом выравнивания.

В Visio все фигуры располагаются по слоям, у фигуры помещенной первой на лист, он самый нижний. Для изменения порядка слоев имеются специальные инструменты, расположенные на панели *Action* и в меню *Shape: Bring to Front, Send to Back, Bring Forward, Send Backward*.

## 2.5 Порядок выполнения лабораторной работы

- 1) Для того чтобы создать новый файл, в программе входим в меню *File\New\Network Diagram\Basic Network*. На экране появляется два окна, одно из которых “рабочее поле”, а другое набор библиотек стандартного для создания диаграммы ЛВС оборудования.
- 2) Сначала на “рабочем поле” вставим образ локальной сети. Для этого в библиотеке щелчком мыши перейдем на закладку под названием *Basic Network Shapes 3D* и выберем мышкой иконку с названием *Ring Network* и перетащим на “рабочее поле”.
- 3) С помощью таких же процедур расставляем и образы компьютеров.
- 4) После того как образы компьютера и образы ЛВС расставлены их можно соединить друг с другом двумя способами:
  - а) Выделить объединяемые части и через меню *Tools* выполнить команду *Connect Shapes*.
  - б) Из строки меню выбрать команду *Connector Tool*, и вручную соединить каждый образ друг с другом.
- 5) Для включения хаба в диаграмму локальной сети, необходимо загрузить дополнительную библиотеку. Для это выполним команду *File\Stencils\Network Diagram\Cabletron* ММАС и вставим образы хабов в диаграмму. Соединяем образы друг с другом с помощью второго способа.
- 6) У соединительных линий можно поменять цвет закрашки и вид линии. Для этого надо выделить линию, и с помощью правой кнопки мыши выполняем команду *Format\Line*. Название образам можно давать двойным нажатием левой кнопки. Образ “Cisco Router” можно вставить из библиотеки сетевого оборудования Cisco 7000 Family.
- 7) Таким же образом вставляем в рабочее поле образы Антенны, Спутника и соединяем их друг с другом (рис.2.11).
- 8) После того как схема готова ее можно сохранить в любом из представленных форматов, например в графическом.

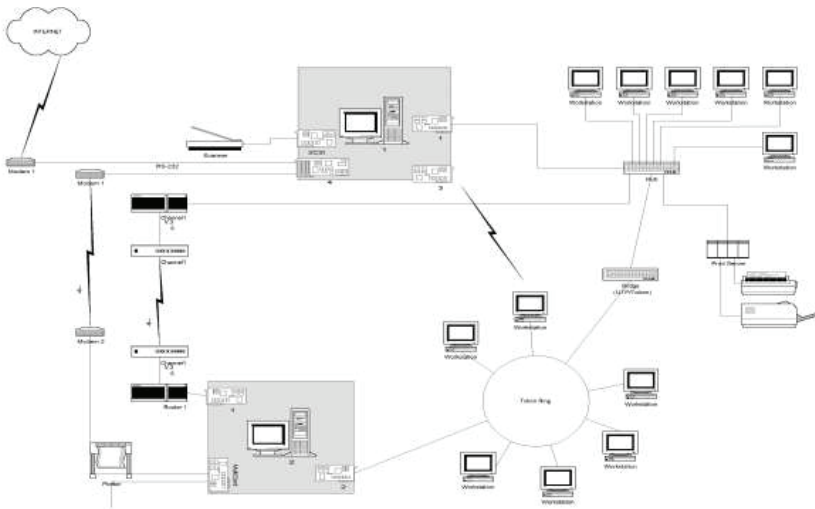


Рисунок 2.11 – Пример схемы

## 2.6. Вопросы к защите лабораторной работы

- 1) Назначение и основные приемы работы с Visio.
- 2) Приведите примеры названий трафаретов и перечислите содержащиеся в них объекты.
- 3) Перечислите основные операции при работе с документом: добавление новых шаблонов, вставка “образа”, графических объектов.
- 4) Как добиться точности в расположении объектов по отношению к листу и друг другу?
- 5) Какие для этого имеются инструменты?
- 6) Как открыть новый рисунок с требуемым трафаретом?
- 7) Опишите технологию копирования объектов Visio в рабочем поле.
- 8) Перечислите возможности редактирования изображений Visio и опишите, как они реализуются.
- 9) Опишите варианты комбинирования фигур Visio.
- 10) Как создается собственный стандартный объект Visio и как он помещается в трафарет?
- 11) Как обеспечить "непрозрачное" наложение надписи Visio на графику?
- 12) В чем смысл введения подложек и как они взаимодействуют с лицевыми страницами?
- 13) Как осуществить печать документа, и какие существуют режимы печати?

## 2.7. Список рекомендуемой литературы

- 1) Гелмерс Скотт А. Microsoft Visio 2013. Шаг за шагом.- ЭКОМ Паблишерз, 2014. - 612 с.

### **3. ЛАБОРАТОРНАЯ РАБОТА №8 СОЗДАНИЕ HTML-ДОКУМЕНТОВ**

#### **3.1. Цель работы**

Целью данной работы является выработка практических навыков создания HTML-документов (простых и составных), обработки встроенной графики и владения основными инструментами их создания, реализованных в редакторе FrontPage.

#### **3.2. Перечень компетенций, формируемых в ходе выполнения лабораторной работы**

Общепрофессиональная компетенция ОПК-2:

**ОПК-2** - Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.

**ИД-2<sub>ОПК-2</sub>** - современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности, в части:

ОПК-2.2.5 - умеет создать HTML-файла в соответствии с заданием;

ОПК-2.3.3 - владеет инструментами создания HTML-документа, реализованными в редакторе FrontPage.

#### **3.3. Задание на выполнение работы**

1. Создать HTML-документ, содержащий: название страницы; заголовки нескольких уровней, выровненные по центру, левому или правому полю; параграфы с текстом (с использованием различных стилей); списки (нумерованные, маркированные, вложенные, определенных).

2. Создать HTML-документ, содержащий таблицу из двух строк и трех ячеек в каждой строке. Установить в таблице фоновый рисунок.

3. Выполнить привязку ссылки к фрагментам изображения.

4. Создать HTML-документ и разместить на нем текстовое поле, командную кнопку и переключатель. Использовать в качестве фона графический файл.

5. Добавить к любому из трех HTML-документов бегущую строку и фоновый звук, и другие динамические объекты.

6. Использовать две гиперссылки для связи какого-либо слова из первого документа со вторым документом.

7. Использовать гиперссылку для связи графического файла из третьего документа с первым документом.

8. Разместить на любом из документов горизонтальную линию.

9. Создать базовую страницу, на которой будут отображаться ранее созданные документы и рамка-фрейм, содержащая графическое изображение. Структура и размеры фреймов задаются в процентах.

10. Распечатать созданные HTML-файлы (коды и вид страниц).

### 3.4. Основные теоретические сведения

HTML-документ представляет собой обычный текстовый файл, который содержит как стандартный текст, так и особые коды форматирования или разметки. Эти коды указывают, каким образом браузер должен отображать данный документ. В HTML документы записываются в ASCII формате и поэтому могут быть созданы и отредактированы в любом текстовом редакторе или с помощью специализированных HTML-редакторов и конвертеров. Выбор редактора зависит исключительно от понятия удобства. Большинство традиционных средств создания документов имеют конвертеры, позволяющие преобразовывать документы к формату HTML. Однако в некотором смысле HTML ограничен, и тогда можно воспользоваться с помощью языка Java, который может привнести на страницу эффекты и функции, которые находятся за пределами возможностью HTML.

Название языка HTML является сокращением от HyperText Markup Language, т.е. "язык гипертекстовой разметки". *Гипертекстом* называется самый обычный текст, украшенный форматированием, графикой, различными мультимедиа-средствами и ссылками на другие документы. *Разметкой* - (*tag*) называется вставка в текст дополнительных служебных символов. Каждый служебный символ в HTML представляет собой команду, которая указывает браузеру, как следует отображать текст. Разметка может быть как очень простой, так и чрезвычайно сложной. В любом случае, размечаемый текст всегда присутствует на экране. Теги начинаются со знака "<" (левой угловой скобки), за которым следует ключевое слово, и заканчиваются символом ">" (правой угловой скобки). Как правило, существует *стартовый тег* (открывает действие некоторого эффекта) и *завершающий тег* (всегда выглядит как ключевое слово, перед которым стоит символ / (косая черта)). Например, теги могут выглядеть так: <Strong>, <TITLE>, </B>, <html>. Регистр в тегах не учитывается, поэтому, например, теги <Title>, <title>, <TITLE>, <titLE> считаются одинаковыми.

Как любой язык, HTML позволяет вставлять в тело документа комментарии, которые сохраняются при передаче документа по сети, но не отображаются браузером. Комментарии могут встречаться в документе где угодно и в любом количестве. Синтаксис комментария:

```
<!-- Это комментарий -->
```

#### 3.4.1. Структура документа

Любой HTML-документ похож на книгу и может быть разбит на отдельные структурные элементы: собственно документ, главы, параграфы, пункты, подпункты, абзацы. Когда web-браузер получает документ, он

определяет, как документ должен быть интерпретирован. Самый первый тег, который встречается в документе, должен быть тегом <HTML>. Данный тег сообщает web-браузеру, что ваш документ написан с использованием HTML. Минимальный HTML-документ будет выглядеть так:

**<HTML> ...тело документа... </HTML>**

Список базовых тегов HTML приведен в табл.3.1.

Таблица 3.1 – Базовые теги HTML

Стартовый	Завершающий	Описание
<HTML>	</HTML>	Обозначение HTML-документа
<HEAD>	</HEAD>	Заголовочная часть документа
<TITLE>	</TITLE>	Заголовок документа
<BODY>	</BODY>	Тело документа
<H1>	</H1>	Заголовок абзаца первого уровня
<H2>	</H2>	Заголовок абзаца второго уровня
<H3>	</H3>	Заголовок абзаца третьего уровня
<H4>	</H4>	Заголовок абзаца четвертого уровня
<H>	</H5>	Заголовок абзаца пятого уровня
<H6>	</H6>	Заголовок абзаца шестого уровня
<P>	</P>	Абзац
<PRE>	</PRE>	Форматированный текст
 		Перевод строки без конца абзаца
<BLOCKQUOTE>	</BLOCKQUOTE>	Цитата

### 3.4.2 Заголовки документов

Каждый HTML-документ должен иметь заголовок, он показывается отдельно и используется, прежде всего, для идентификации документа (например, при поиске). Тег заголовочной части документа должен быть использован сразу после тэга <HTML> и более нигде в теле документа. Данный тег представляет собой общее описание документа. Заголовок должен описывать цель документа и содержать не больше 5-6 слов. Практически во всех браузерах заголовок документа виден в верхней части экрана (окна). Для выделения заголовка *служат символы:*

**<HEAD><TITLE>Заголовок</TITLE></HEAD>**

Следует обратить внимание, технически, стартовые и завершающие тэги типа <HTML>, <HEAD> и <BODY> необязательны. Но настоятельно рекомендуется

их использовать, поскольку использование данных тегов позволяет WEB-браузеру уверенно разделить заголовочную часть документа и непосредственно смысловую часть.

### 3.4.3 Заголовки разделов документов

HTML имеет шесть уровней заголовков, имеющих номера с 1 по 6 (заголовок первого уровня является заголовком высшего уровня). По сравнению с нормальным текстом, заголовки выделяются шрифтом - размером и толщиной букв. Первый заголовок в каждом документе должен быть выделен <H1>. Синтаксис заголовков: <Hy> *Текст заголовка*</Hy> где y - число от 1 до 6, определяющее уровень заголовка.

### 3.4.4 Теги тела документа

Теги тела документа идентифицируют отображаемые в окне компоненты HTML-документа. Тело документа может содержать ссылки на другие документы, текст и другую форматированную информацию.

Тело документа должно находиться между тэгами <BODY> и </BODY>.

### 3.4.5 Гиперссылки

Главное преимущество HTML состоит в его способности связываться с другими документами. WEB-браузер выделяет (обычно цветом и/или подчеркиванием) ключевые слова, являющиеся гипертекстовыми ссылками (гиперссылками). Описывается ссылка на другой документ следующим образом:

<A HREF="имя файла"> *Текст, который будет служить как обращение к другому документу*</A>.

Приведем пример такой гипертекстовой ссылки:

<A HREF="mainfile.html">Пример HTML-текста</A>

Здесь ключевые слова «Пример HTML-текста» являются гиперссылкой на файл mainfile.html, который лежит в той же директории, что и текущий документ. Вы можете ссылаться на документ, лежащий в любой директории, описав к нему полный путь. Так, например, ссылку на файл Proba.html, лежащий в поддиректории LABORA можно описать как:

<A HREF="LABORA/Proba.html">New Jersey</A>

Это так называемые *относительные ссылки*. Вы также можете использовать абсолютное имя файла (полный путь). В общем случае, использование ссылки по абсолютному имени файла более предпочтительно.

### 3.4.6 Обращение к определенным разделам других документов

Гиперссылки могут также использоваться для соединения с определенными разделами документов. Предположим, мы хотим соединить документ А с первой главой документа В, для чего нам необходимо создать *именованную гиперссылку* в документе В:

Здесь вы можете увидеть <A NAME = "Глава 1">Главу 1</a>

Текст первой главы.

Теперь, описывая ссылку в документе А, надо включить не только имя файла, например, "documentB.html", но также и имя гиперссылки, отделяемое символом (#):

Здесь вы можете увидеть текст `<A HREF= "documentB.html#Глава1">Главы 1 </A>` документа В.

Теперь щелкнув мышью на слове "Главы 1" в документе А, вы переходите непосредственно в Главу 1 документа В.

### 3.4.7 Соединения с разделами текущего документа

Техника соединения аналогична описанной выше, только опускается имя файла, например:

Это `<A HREF = "#Глава1">Глава 1</A>` текущего документа.

### 3.4.8 Списки

Списки делают информацию более наглядной и удобной для восприятия. Более того, внутри списков (точнее, для произвольного отображения элементов списка) тоже можно использовать различные теги. При создании веб-страницы можно использовать:

- 1) Menu list (список меню - `<menu>`);
- 2) Directory list (список директорий - `<dir>`);
- 3) Ordered list (нумерованный список - `<ol>`);
- 4) Unordered list (маркированный список или нумерованный - `<ul>`);
- 5) Definition list (список определений - `<dl>`).

*Ненумерованные списки:* `<UL><LI>список пунктов</UL>`, например:

```
<LI> информатика
<LI> программирование
</UL>
```

что дает на экране:

```
информатика
программирование
```

*Нумерованные списки:* `<OL><LI>список пунктов</OL>`,

например:

```
</OL>
<LI> экзамен
<LI> зачет
<LI> лекция
</OL>
```

Браузер автоматически нумерует элементы такого списка:

1. экзамен
2. зачет
3. лекция

*Списки определений* - Список определений обычно состоит из чередования термина (DT) и определения (DD). Обычно Web- браузер определения



располагает на новой строке. Приведем пример списка определений:

```
<DL>
<DT> ASCII
<DD> ASCII (American Standard Code for Information Interchange).
<DT> TTF
<DD> TTF (True Type Font).
</DL>
```

что дает:

```
ASCII
ASCII (American Standard Code for Information Interchange)
TTF
TTF (True Type Font).
```

### 3.4.9 Стили

Можно описывать специальными стилями отдельные слова или предложения. Имеются два типа стилей: логический и физический. *Логические стили* определяют текст согласно заданному значению, в то время как *физические стили* определяют некоторые участки текста.

Логические стили:

```
<DFN> - служит для описания определений;
<EM > - служит для выделения слов;
<CITE> - служит для выделения заголовков книг, фильмов, цитат;
<CODE> - служит для выделения программных кодов, текстов программ
и т.п. Изображается шрифтом фиксированной ширины;
<KBD> - используется для ввода с клавиатуры пользователя. Может быть
изображено жирным шрифтом (но в большинстве браузеров изображается
специальным шрифтом);
<SAMP> - используется для машинных сообщений. Изображается
шрифтом фиксированной ширины;
<STRONG> - служит для ОСОБОГО выделения слов. Обычно
выделяется жирным шрифтом;
<VAR> - используется для символьных переменных.
```

Существуют физические способы выделения - автор задает стиль написания текста, описывая шрифт в исходном HTML-документе. Можно задать:

```
<B>, </B> - жирный шрифт (это жирный шрифт);
<I>, </I> - наклонный шрифт (это наклонный шрифт);
<TT>, </TT> - фиксированный шрифт (шрифт заданной ширины) (это
fixed шрифт).
```

### 3.4.10 Встроенные графические изображения

Большинство Web-документов содержат встроенную графику. Делается это при помощи тега <IMG...ISMAP>. Использование данного тега позволяет значительно улучшить внешний вид и функциональность документов. Чтобы включить рисунок, надо описать гиперссылку на него:

`<IMG SRC="URL">`

где "URL" - имя файла, содержащего рисунок. Автоматически рисунок выравнивается по нижнему краю сопровождающего его текста, но Вы можете задавать взаимное расположение рисунка и текста. Форматирование положения рисунка задается включением в гиперссылку пункта "ALIGN = ". Синтаксис тега:

`<IMG SRC="URL" ALT="text" HEIGHT=n1 WIDTH=n2  
ALIGN=top\middle\bottom>`,

где ALT="text" - данный необязательный элемент задает текст, который будет отображен браузером, не поддерживающим отображение графики или с отключенной подкачкой изображений. Обычно, это короткое описание изображения, которое пользователь мог бы или сможет увидеть на экране. Если данный параметр отсутствует, то на месте рисунка большинство браузеров выводит пиктограмму (иконку), активизировав которую, пользователь может увидеть изображение. Тег ALT рекомендуется, если ваши пользователи используют браузер, не поддерживающий графический режим, например Lynx;

HEIGHT=n1 - данный необязательный параметр используется для указания высоты рисунка в пикселях. Если данный параметр не указан, то используется оригинальная высота рисунка. Этот параметр позволяет сжимать или растягивать изображение по вертикали, что позволяет более четко определять внешний вид документа;

WIDTH=n2 - параметр также необязателен, как и предыдущий. Позволяет задать абсолютную ширину рисунка в пикселях;

ALIGN - данный параметр используется для выравнивания текста по верхней границе (top), по центру (middle), по нижней границе (bottom).

#### **3.4.11 Основные теги таблицы**

Основные теги, описывающие таблицу: `<TABLE>...</TABLE>`.

Все элементы таблицы должны находиться внутри этих двух тегов. По умолчанию таблица не имеет обрамления и разделителей. Обрамление добавляется атрибутом BORDER. Количество строк таблицы определяется количеством встречающихся пар тегов `<TR>..</TR>`. Строки могут иметь атрибуты ALIGN и VALIGN, которые описывают визуальное положение содержимого строк в таблице. Теги `<TD>...</TD>`

описывают стандартную ячейку таблицы. Ячейка таблицы может быть описана только внутри строки таблицы. Каждая ячейка должна быть пронумерована номером колонки, для которой она описывается. Если в строке отсутствует одна или несколько ячеек для некоторых колонок, то браузер отображает пустую ячейку. Расположение данных в ячейке по умолчанию определяется атрибутами ALIGN=left и VALIGN=middle.

Основные атрибуты таблицы:

**BORDER**- данный атрибут используется в теге TABLE. Если данный атрибут присутствует, граница таблицы прорисовывается для всех ячеек и для

таблицы в целом. BORDER может принимать числовое значение, определяющее ширину границы, например BORDER=1;

*ALIGN*- если атрибут ALIGN присутствует внутри тегов <CAPTION> и </CAPTION>, то он определяет положение подписи для таблицы (сверху или снизу). По умолчанию ALIGN=top. Если атрибут ALIGN встречается внутри <TR>, <TH> или <TD>, он управляет положением данных в ячейках по горизонтали. Может принимать значения left (слева), right (справа) или center (по центру);

*VALIGN* - данный атрибут встречается внутри тегов <TR>, <TH> и <TD>. Он определяет вертикальное размещение данных в ячейках. Может принимать значения top (вверху), bottom (внизу), middle (по середине) и baseline (все ячейки строки прижаты кверху);

*NOWRAP*- данный атрибут говорит о том, что данные в ячейке не могут логически разбиваться на несколько строк и должны быть представлены одной строкой;

*COLSPAN*- указывает, какое количество ячеек будет объединено по горизонтали для указанной ячейки (количество столбцов). По умолчанию значение равно -1;

*ROWSPAN*- указывает, какое количество ячеек будет объединено по вертикали для указанной ячейки (количество строк). По умолчанию - 1;

*COLSPEC*- данный параметр позволяет задавать фиксированную ширину колонок либо в символах, либо в процентах, например COLSPEC="20%".

### 3.4.12 Фреймы

Формат документа, использующего фреймы, внешне напоминает формат обычного документа, только вместо тега BODY используется контейнер FRAMESET, содержащий описание внутренних HTML-документов и собственно информацию, размещаемую во фреймах.

```
<HTML>
<HEAD>... </HEAD>
<FRAMESET>... </FRAMESET>
</HTML>
```

### 3.4.13 Image Map

В HTML Image Map - это технология, которая позволяет привязывать ссылки к фрагментам изображения. Щелкая мышью на отдельных частях изображения, пользователь может переходить по той или иной ссылке на разные web-страницы.

### 3.4.14 White-space

White-space - это свойство, позволяющее менять количество пробелов между словами при отображении веб-страницы. По умолчанию любое количество пробелов в HTML-коде браузер интерпретирует как один пробел. Главное преимущество white-space заключается в том, что при написании HTML-кода разработчику не нужно беспокоиться о лишних пробелах, которые он мог ввести: браузер при отображении веб-страницы уберёт все лишние

пробелы. Разработчик может добавлять в код пробелы не только случайно, но и специально: чтобы код стал более читабельным.

### 3.4.15 Знак *copyright* на веб-странице

Знак *copyright* относится к спецсимволам. Чтобы разместить его на веб-странице, нужно в соответствующем фрагменте HTML-кода набрать на клавиатуре последовательность *&copy* или *&#169*.

## 3.5. Основные приемы работы

### 3.5.1. Microsoft FrontPage

Microsoft FrontPage – одно из наиболее мощных средств создания Web-страниц. Основное назначение такого рода редакторов – это избавление пользователя от довольно трудоемкого прямого написания HTML-кода. Все что требуется от пользователя – это разместить на форме объекты: командные кнопки, поля, ссылки с неких объектов и т.д., а сам код пропишет редактор без вашего вмешательства. Рассмотрим основные объекты:

1) *Web-Компонент* - «динамические объекты» на web-странице, которые запускаются в то время, когда разработчик сохраняет страницу на web-сервере, или когда пользователь просматривает страницу. В комплект FrontPage включены web-компоненты, позволяющие помещать метки времени, а также средства вставки и поиска на Web -страницы. При выборе данного пункта открывается следующее окно (рис.3.1):

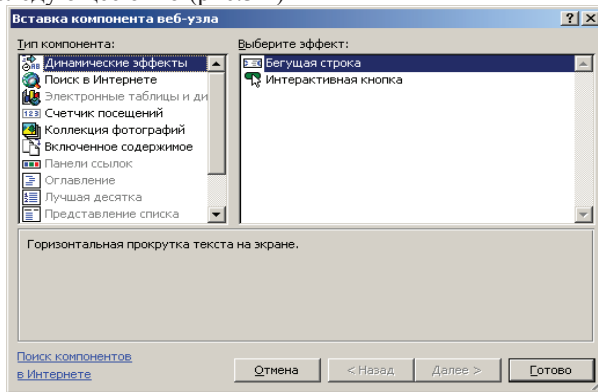


Рисунок 3.1 - Окно Вставка компонентов веб-узла

2) *Бегающая строка* - размещение на форме бегущей строки. После выбора этого элемента на экране появляется меню «Свойства бегущей строки». В поле «Текст» вводится необходимый текст, затем выбирается направление движения текста (вправо или влево), скорость перемещения, можно изменить цвет букв и т.д. (рис.3.2).

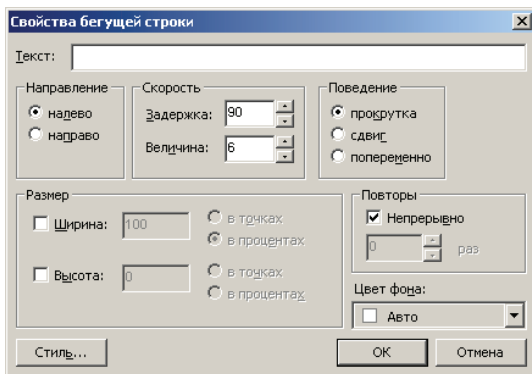


Рисунок 3.2 - Окно Свойства бегущей строки

3) *Разметка HTML* - используется для установки меток;

4) *Сценарий* - существуют несколько разновидностей так называемых языков сценариев: VBScript, JavaScript, Perl и т.д. С помощью них значительно расширяются возможности HTML. Например, при разработке страницы для торговой фирмы, на странице оформления заказов можно поместить, написанный на языке сценариев, калькулятор и т.д. После выбора этого пункта меню на экране появится дополнительное меню, в котором необходимо указать тот язык сценариев, который используется в HTML документе;

5) *Гиперссылка* - служит, например, для связи какого-либо объекта одного документа (текста, картинки, кнопки и т.д.) с другой страницей, расположенной на локальном диске или в сети. В результате при щелчке мышью на этом объекте загружается страница, на которую указывает ссылка. После выбора этого пункта меню на экране появляется меню (рис.3.3):

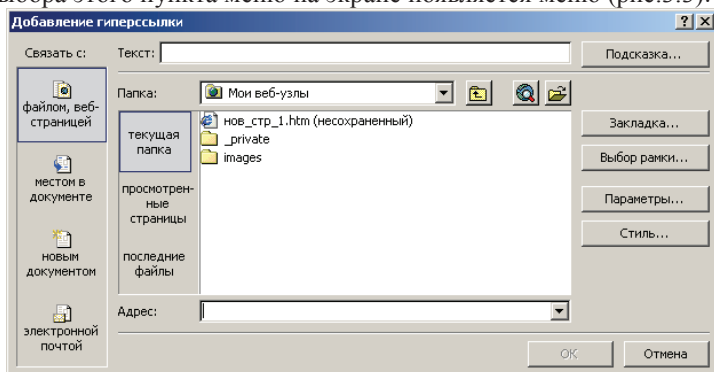


Рисунок 3.3 - Окно Добавление гиперссылки

В поле «Адрес (URL)» указывается адрес, на который будет делаться гиперссылка. Например, необходимо связать какое-либо слово из одного документа с другим документом. Для этого необходимо в первом документе

выделить "мышью" требуемое слово и нажать на гиперссылку, в появившемся окне в поле «Адрес (URL)» необходимо ввести адрес расположения второго документа. В результате мы установили ссылку. Если запустить страницу и щелкнуть "мышью" на слове в первом документе, то будет загружен второй документ.

### 3.5.2 Последовательность действий

Для создания страницы, приведенной на рис.3.4:

1. Нажмите кнопку «Выравнивание по центру» и введите текст, изображенный вверху страницы, например, «Главная страница».
2. Выделите введенный текст и щелкните мышкой на кнопке для изменения цвета текста.
3. В меню «Вставка» выберите пункт «Горизонтальная линия».

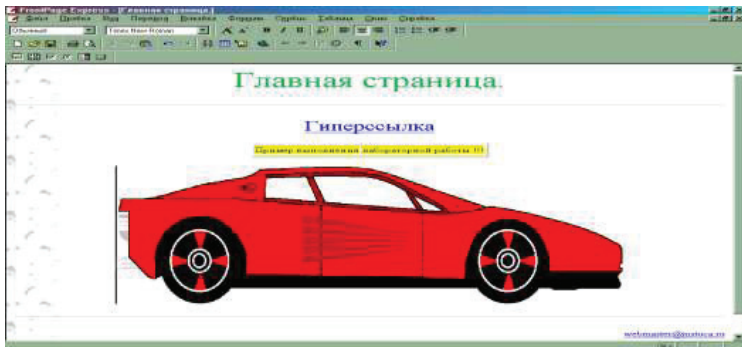


Рисунок 3.4 – Пример web-страницы

4. Теперь создадим гиперссылку. Наберите любое слово и затем выделите его мышкой. Нажмите на значок гиперссылки. В появившемся меню укажите путь к другому html-документу.
5. В меню «Таблица» выберите пункт «Вставить». В появившейся вкладке установите количество столбцов и строк. В данном примере одна строка и два столбца.
6. Щелкните мышкой внутри ячеек таблицы и введите текст.
7. Щелкните правой кнопкой мыши на левой ячейке таблицы, в появившемся меню выберите вкладку «Свойства ячейки» и измените цвет фона.
8. Выполните аналогичные действия для правой ячейки.
9. Нажмите кнопку «Вставить изображение» и выберите какой-либо графический файл.
10. Повторите пункт 3.
11. Выберите в меню «Формат» пункт «Фон» и в строке «Фоновое изображение» укажите путь к файлу, который будет использоваться в качестве фона.

Ниже приведен HTML- код создаваемого документа.

```

<html>
<head>
<title>Главная страница</title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
</head>
<body>
<p align="right"><strong><font face="Lucida Sans">Главная страница
</font></strong></p>
<hr>
<p align="center"><a
href=" ../Mou%20документы/demonstra/r.avi">Гиперссылка</a></p>
<table border="1" width="100%">
<tr>
<td width="50%"><p align="center">пример выполнения</td>
<td width="50%">лабораторной работы!!</td>
</tr>
</table>
<p></p>
<hr>
<p align="right"><a href=" ../Mou%20документы/demonstra/Af.arj"> webmaster @
mstusa.ru</a></p>
</body>
</html>

```

На рис. 3.5 приведена страница, содержащая формы для ввода данных и радиокнопки



Рисунок 3.5 – Пример web-страницы с динамическими элементами

Ниже приведен фрагмент HTML- код создаваемого документа.

```

<html>
<head>
<title> Задание </title>
</head>
<body bgcolor="#F1CAF3">

```

```

<p><textarea name="comment"></textarea></p>
<p><input type="submit"></p>
<p style="text-align: center"><button>Кнопка с текстом</button>
<a href="Task1.html"><button> Перейти на 1
страницу</button></p></a>
<input type="button" name="Kn" value="Кнопка" title="Просто кнопка"
<br><br>
<input type="checkbox" name="Pr" value="Переключатель">Положение 1
<input type="checkbox" name="Pr" value="Переключатель">Положение 2
<input type="checkbox" name="Pr" value="Переключатель">Положение 3
<br><br>
<p><input name="perek" type="radio" value="punkt1">Пункт 1</p>
<p><input name="perek" type="radio" value="punkt2">Пункт 2</p>
<p><input name="perek" type="radio" value="punkt3" checked>Пункт 3</p>
<p><input type="submit" value="Выбрать"></p>
</body>
</html>

```

### 3.5. Вопросы к защите лабораторной работы

1. Что такое HTML?
2. Из чего состоит форматированный текст HTML?
3. Что является разметкой в HTML?
4. Назовите и объясните синтаксис основных тегов HTML.
5. Что такое гиперссылка и для чего она применяется?
6. Какие основные виды списков применяют при создании веб-страницы
7. Как обеспечатывается возможность включения ссылок на графические и иные типы данных в HTML-документ?
8. Как разместить бегущую строку и изменить скорость ее движения?
9. Как поместить изображение на документ?
10. Как изменить цвет фона и толщину бордюра таблицы?
11. Как использовать на странице фоновый звук?
12. Как изменить свойства ячеек таблицы?
13. Что такое Image Map?
14. Что такое white-space и его основное преимущество?

### 3.6. Список рекомендуемой литературы

- 1) Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд.- СПб Питер, 2018 - 768 с.
- 2) Дронов В.А. HTML и CSS. 25 уроков для начинающих - СПб.:ВНУ-Санкт-Петербург, 2020. -400 с.