

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Пономарева Светлана Викторовна
Должность: Проректор по УР и НО
Дата подписания: 20.09.2023 20:50:37
Уникальный программный ключ:
bb52f959411e64617366ef2977b97e87139b1a2d



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)
АВИАЦИОННЫЙ КОЛЛЕДЖ**

УТВЕРЖДАЮ

Директор колледжа

А.И. Азарова

личная подпись

инициалы, фамилия

« ___ » _____ 2021 г.

**Методические рекомендации по выполнению
практических работ по дисциплине**

ОП.16 Пакеты прикладных программ

для студентов специальности

09.02.07 Информационные системы и программирование

Ростов-на-Дону
2021 г.

Практическая работа № 1

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Изучение интерфейса пакетов компьютерной графики.

Цель изучения темы: Изучение интерфейса приложения GIMP.

Выполнение работы:

1. Теоретическое обоснование

Основные панели

Основные функции и пункты меню переведены на русский язык. В GIMP реализованы стандартные функции создания, открытия, сохранения файлов, печати документов, вырезания и копирования в буфер и вставки из него и т.д.

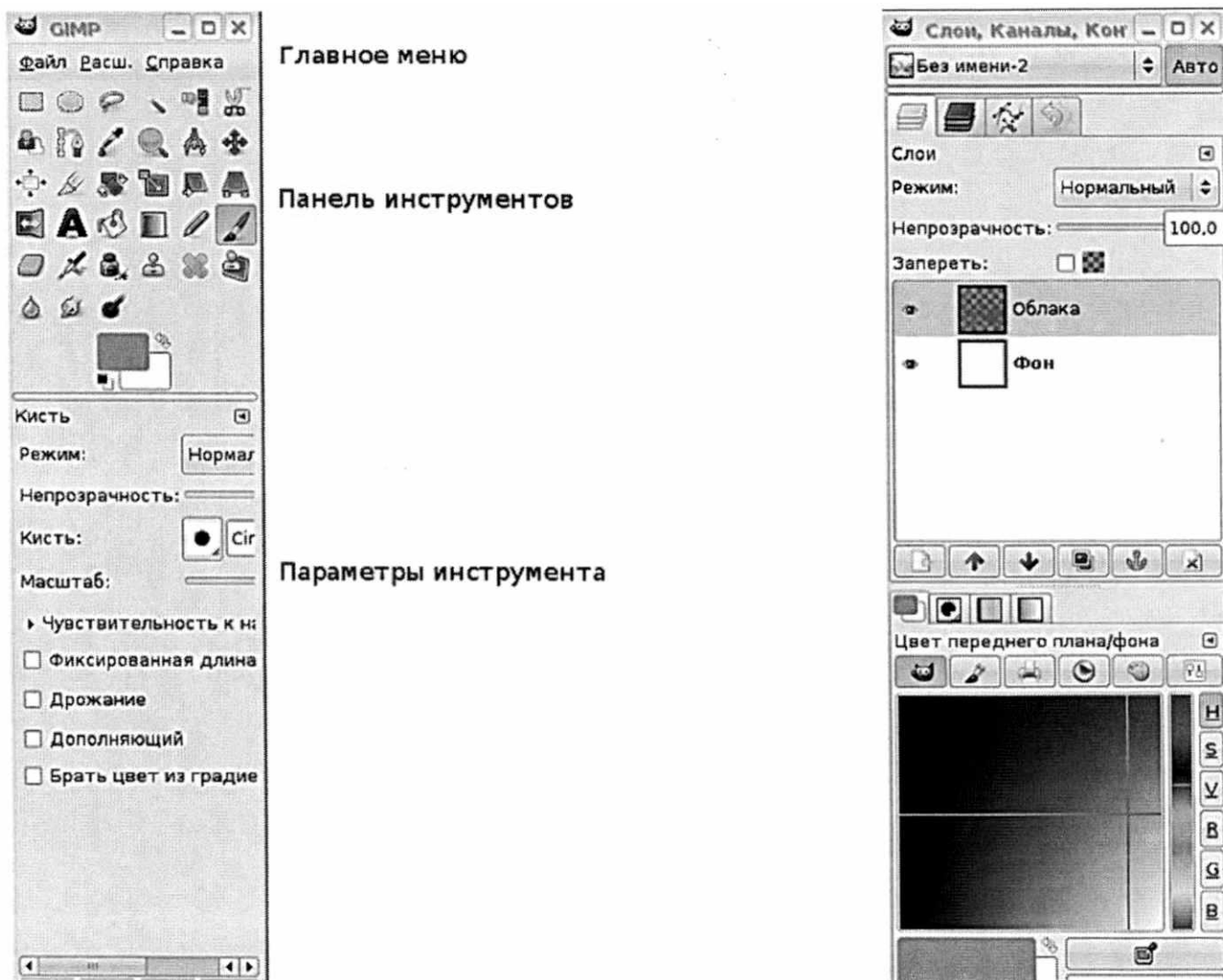


Рис. 1. Основные панели.

В основном окне программы представлены только три пункта меню, отвечающие за открытие и создание нового файла, помощь и расширенные функции программы. При соответствующей настройке кроме этого окна открываются и дополнительные плавающие палитры (выбор цвета, история, навигация, выбор слоя и др.). Для того чтобы выбрать какой-либо инструмент, достаточно щелкнуть мышью по его пиктограмме.

Каждое дочернее окно GIMP содержит следующие пункты меню.

- **Файл** - команды, имеющие отношение к операциям создания, открытия, сохранения

файлов, отправки файлов по электронной почте, открытия изображения на новый слой активного файла, восстановления, а также команды закрытия одного или нескольких окон, выхода из GIMP и список последних открытых документов.

- **Правка** (во многих других программах принято также название «редактирование») - команды, предназначенные для отмены последнего действия, работы с буфером обмена, заливки выделенной области, настройки клавиш быстрого доступа и единиц измерения, используемых при работе с изображением.
- **Выделение** - пункт содержит команды для работы с выделенными областями: выделение всей картинки, снятия и возврата выделения, инвертирования, модификации, растушёвки выделенной области и др.
- **Вид** - команды этого пункта меню предназначены для управления масштабом, предварительного просмотра редактируемого изображения в различных режимах, установки внешнего вида дочернего окна: отображение линеек, меню, полос прокрутки, строки состояния.
- **Изображение** - содержит команды, предназначенные для настройки размера и разрешения изображения, размеров холста, поворота изображения на 90 и 180 градусов. Также в этом пункте меню содержатся функции для кадрирования изображения, объединению видимых слоев и настройке сетки изображения.
- **Слой** - команды, предназначенные для работы со слоями: создание, дублирование слоев, наложение эффектов, группировка и выравнивание слоев, операции по их слиянию.
- **Цвет** - в этот пункт меню вынесены все функции для цветовой и тоновой коррекции изображения.
- **Инструменты** - данный пункт меню дублирует находящиеся на панели инструментов кнопки, а также содержит ссылки на наиболее часто используемые инструменты цвет коррекции. Это очень удобно в том случае, когда открыто много файлов, и найти панель инструментов достаточно проблематично. Также инструменты в данном меню разбиты на группы, чего нет на панели инструментов, и, таким образом, в них легче разобраться.
- **Фильтры** - стандартные фильтры, поставляющиеся вместе с программой, а также те, которые пользователь дополнительно установил на свой компьютер.

Создание нового изображения

Для того чтобы создать новое изображение, окно команды File → New (Новый), в котором задаётся цветовая модель изображения, фон (белый или прозрачный), а также размеры изображения (рис. 2).

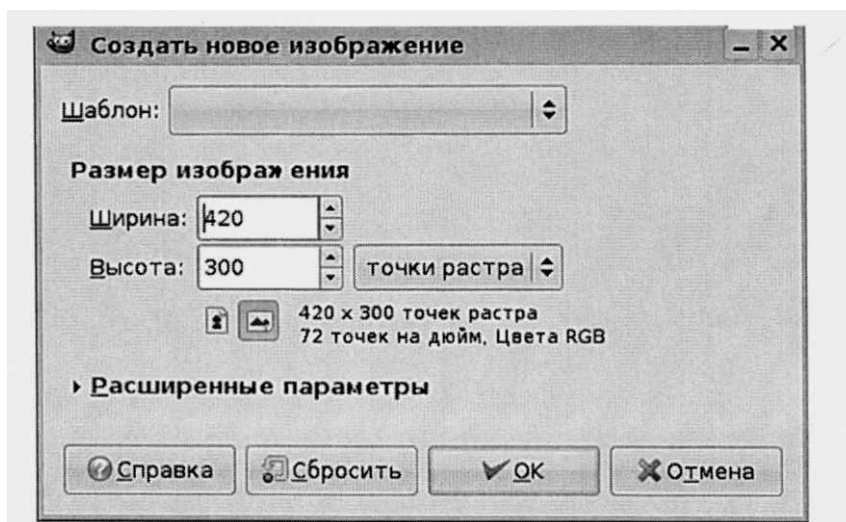










Рис. 2. Окно создания нового файла.

Инструменты в GIMP и их свойства

GIMP использует несколько различных типов кистей:

- простые - большинство устанавливаемых в GIMP кистей; отображаются в диалоге выбора кисти как черно-белые пиксельные изображения и помечены, синим треугольником в правом нижнем углу;
- цветные - отображаются как цветные изображения, кисть использует только те цвета, которые видны на изображении кисти в диалоге;
- анимированные - позволяют создать более одного типа отпечатков на изображении, отмечены в диалоге красным треугольником в правом нижнем углу;
- параметрические - создаются с помощью редактора кистей.

GIMP включает в себя набор из 10 инструментов рисования, которые предоставляют не только операции, непосредственно связанные с рисованием.

(кисть , карандаш , аэрограф , перо , ластик , затемнение и осветление изображения , Размывание и увеличение резкости , а также инструмент «Палец» ).

2. Алгоритм выполнения работы

- 1) Произведите настройку параметров рабочего окна редактора, удобную для пользователя: размер панели инструментов, отображаемые дополнительные палитры и т.д.
- 2) Уточните и запишите горячие клавиши, используемые для вызова тех или иных команд.
- 3) Создайте файл стандартных размеров.
- 4) Выполните рисунок на тему «Времена года», используя максимальный набор инструментов.
- 5) Сохраните работу в собственной папке на рабочем столе.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как настроить положение основных панелей? Включить/выключить дополнительные?
- 2) Какие инструменты выделения представлены, какие задачи они решают?
- 3) Какой тип кистей наиболее удобен в работе?
- 4) Какие фильтры установлены на ваш компьютер? Что можно выполнить с их помощью?
- 5) Как работает инструмент «Ластик» по сравнению с программой Paint?

Практическая работа № 2

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Инструменты выделения и трансформации GIMP

Цель изучения темы: Изучение инструментов для выделения и трансформирования части изображения.

Выполнение работы:

1. Теоретическое обоснование

Инструменты выделения

Для выделения простых областей (прямоугольных и круглых) предназначены специальные инструменты, активизировать которые можно верхней левой кнопкой на панели инструментов. Если установить на эту кнопку указатель мыши и удерживать кнопку мыши нажатой, будет предоставлен выбор из четырех инструментов выделения.

- **Прямоугольная выделенная область** (Rectangular Marquee Tool) - инструмент используется для прямоугольных выделенных областей.
- **Эллиптическая выделенная область** (Elliptical Marquee Tool) - инструмент применяется для создания эллиптической (или круглой) выделенной области.
- **Лассо** (Lasso Tool) - используется для создания выделенной области произвольной формы с помощью мыши. Выделенная область должна быть замкнута.
- **Выделение по цвету** (Select Color) - этот инструмент позволяет облегчить процесс выделения за счёт автоматического определения линий наибольшего цветового контраста и привязки к ним границы области выделения.
- **Волшебная палочка** (Magic Wand Tool). Чтобы использовать этот инструмент достаточно щёлкнуть мышью на желаемой области, и она окажется выделенной. Программа автоматически определяет пиксели с одинаковыми или близкими значениями яркости, расположенные рядом.

Инструменты трансформации

- **Вращение** - инструмент, который позволяет поворачивать на заданный угол часть изображения, контур или выделение.
- **Масштаб** - масштабирование слоев, выделенных областей или контуров.
- **Искажение** - инструмент используется для перемещения одной части изображения, слоя, выделения или контура в одну сторону, а другой части - в противоположном.
- **Перспектива** - инструмент трансформации или искажения изображения, с ее помощью удается передать на плоской фотографии ощущение объема изображения.
- **Зеркало** - инструмент, отражающий зеркально часть изображения, выделения, контура горизонтально или вертикально.

2. Алгоритм выполнения работы

- 1) Открыть произвольную фотографию, на которой изображён человек, либо животное. Выделить фигуру, используя различные инструменты GIMP.
- 2) Произвести трансформацию выделенной части изображения, используя инструменты трансформации.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как вырезать область правильной формы?
- 2) Какой инструмент лучше использовать, когда необходимо аккуратно вырезать маленькую область?
- 3) Если фотография достаточно четкая и контрастна, чем выделять ее области?
- 4) Уточните разницу между инструментами «Перспектива» и «Искажение».
- 5) Какие проблемы возникают при масштабировании растрового изображения?

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Компьютерная ретушь фотоизображений

Цель изучения темы: Изучение инструментов ретуширования и редактирования части изображения.

Выполнение работы:

1. Теоретическое обоснование

Для ретуширования фотографий, устранения крупных и мелких дефектов имеется набор инструментов: штамп, штамп по перспективе, лечебная кисть, палец, резкость или размывание.

- **Штамп** - он позволяет своеобразно копировать разные регионы изображения. Наведите курсор на ту часть изображения, которую хотите копировать. Нажмите на клавиатуре кнопку Ctrl и щелкните кнопкой мыши в определенной точке копируемого региона: в ней появится небольшой крестик. Теперь переведите курсор в ту часть изображения, куда вы хотите вставлять копируемый элемент. Нажмите кнопку мыши и не отпуская ее, начинайте рисовать, как если бы рисовали кистью.
- **Штамп по перспективе** - инструмент также позволяет копировать отдельные регионы изображения, но с учетом искажения перспективы. Перед применением этого инструмента необходимо с помощью специальной рамки задать перспективу имеющегося изображения - указать, какая его часть находится ближе к зрителю, а какая - дальше.
- **Лечебная кисть** - этот инструмент удобен в случаях, когда копируемый элемент вставляется на неоднородный фон. В такой ситуации общие оттенки копируемого элемента и фона будут приведены в соответствие, что позволит более аккуратно совместить эти элементы.
- **Палец** - инструмент позволяет смазывать определенные участки изображения, подтягивать края, размывать краски, как если бы по еще не высохшей картине провели пальцем.
- **Резкость или размывание** - инструмент позволяет настроить резкость или, наоборот, размывать изображение в отдельных его частях, указываемых при помощи специальной кисти.

2. Алгоритм выполнения работы

- 1) Открыть файл Пара-jpg, на которой изображены два человека с дефектами сканирования.
- 2) Используя набор инструментов выполнить: ретуширование фотографии, косметический макияж на лицах моделей, устранить дефекты сканирования.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как восстановить поврежденные участки фотографии?
- 2) Как нарисовать отсутствующие части изображения?
- 3) Как удалить мелкие недостатки с лиц моделей?
- 4) Как устранить дефекты сканирования изображения?
- 5) Зачем размывать часть изображения?

Практическая работа № 4

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Декорация фотографий

Цель изучения темы: Изучение фильтров, установленных в программе GIMP.

Выполнение работы:

1. Теоретическое обоснование

С помощью эффекта «Снимок в снимке» внимание привлекается к части фотографии, оставляя большую часть затемненной.

2. Алгоритм выполнения работы

- 1) Открыть файл Para1.jpg, выделите ту часть фотографии, которая должна находиться в фокусе. Поверните на произвольный угол, скопируйте и расположите копию на отдельном слое.
- 2) Создайте новый прозрачный слой, поместите его между двумя существующими, примените к выделению **фильтр «Свет и Тень»** - «Отбрасываемая тень».
- 3) Затемните фоновый слой и осветлите фрагмент фотографии.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как работает инструмент Осветление/Затемнение?
- 2) Какие настройки оказались оптимальными для фильтра «Отбрасываемая тень»?
- 3) Как перемещать часть изображения на новый слой?
- 4) Какие еще фильтры можно применить к данной работе?
- 5) Как, используя инструмент прямоугольное выделение, нарисовать рамку к фотографии?

Практическая работа № 5

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Анимация движения, формы объекта.

Цель изучения темы: Изучение возможностей создания анимации графическим редактором GIMP.

Выполнение работы:

1. Теоретическое обоснование

Анимация способна оживить часть изображения в развлекательных целях, привлечь внимание к важным вопросам или показать процессы, объекты, явления с образовательной точки зрения. Графический редактор GIMP позволяет создавать простейшую, по кадрам нарисованную анимацию или применить к уже готовым объектам специальный фильтр.

2. Алгоритм выполнения работы

- 1) Открыть файл 15154.jpg. Рассмотрите фотографию ежа, на него необходимо наложить три анимированных действия - подмигивание, шевеление носом и лапами.
- 2) Выделите и перенесите на отдельные слои части изображения, которым будет придаваться движение, восстановите штампом фоновый слой.
- 3) Сделайте несколько копий носа, лап, глаза. На каждом слое, согласно рисунка движения трансформируйте и переместите элементы изображения.
- 4) Сделайте копии фонового слоя и расположите все слои в нужном порядке. Не забывайте объединять нужные фрагменты изображения в единый слой.
- 5) Запустите фильтр «Анимация», убедитесь, что все слои собраны в нужном порядке, сохраните файл с расширением GIF как анимацию.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как создается анимированное изображение?
- 2) Какие манипуляции необходимо провести с «шевелиющимися» частями?

- 3) Почему необходимо восстановить фоновый слой?
- 4) Как объединять слои в единое изображение?
- 5) Как просмотреть готовую работу? С какими настройками необходимо сохранить файл?

Практическая работа № 6

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Создание gif-анимации для Web-страниц.

Цель изучения темы: Изучение возможностей создания анимации для Web-страниц.

Выполнение работы:

1. Теоретическое обоснование

Умелое использование анимации значительно оживляет Web-страницы. Традиционно для рекламных целей в сети используют баннеры - небольшие анимированные графические изображения.

2. Алгоритм выполнения работы

- 1) Откройте новый документ размером 240x400 пикселей с белым фоном. Создадим баннер для фирмы «barDUCK» - чай и кофе на вынос.
- 2) Выбираем основной цвет #ebbb7d и фоновый #93643b, а затем - инструмент Градиент (L). В настройках градиентной заливки выбираем тип градиента: от основного к фоновому и заливаем фоновый слой баннера.
- 3) Вставляем логотип, размещаем его на отдельном слое с названием «Логотип». Выбираем для размещения правый верхний угол.
- 4) Создаем новый прозрачный слой и рисуем круг с помощью Эллиптического выделения, инструментом Плоская заливка закрашиваем цветом #ffe7b1, добавляем текст «Начни свой день правильно».
- 5) На новом слое нарисуем стакан, на передней части изображения разместим надпись «Возьми кофе с СОБОЙ».
- 6) Сведем в один слой Фон, Стакан и логотип. Должны получиться три готовых к работе слоя. Настраиваем анимацию.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как создается анимированный объект - баннер?
- 2) Какие фильтры можно использовать для декорирования баннера?
- 3) Как использовать направляющие и линейку?
- 4) Как объединять слои в единое изображение?
- 5) Как просмотреть готовую работу? С какими настройками необходимо сохранить файл?

Практическая работа № 7

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Интерфейс приложения Inkscape. Создание простых объектов.

Цель изучения темы: Создание простых объектов в среде векторного редактора Inkscape.

Выполнение работы:

1. Теоретическое обоснование

Панель в левой части окна содержит основные инструменты для рисования и редактирования объектов. Верхняя панель (под меню) содержит элементы, специфичные для каждого конкретного инструмента. На полосу состояния внизу окна выводятся полезные подсказки и рабочие сообщения (рис. 3).

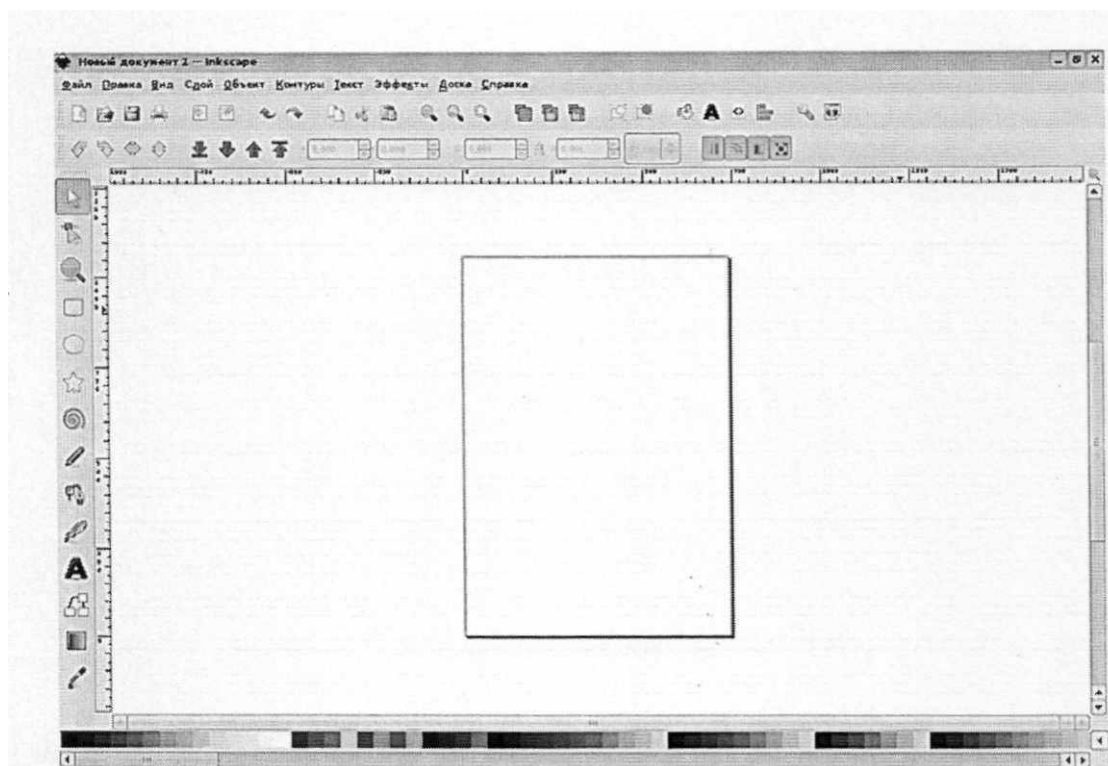


Рис. 3. Интерфейс окна программы Inkscape

Для создания нового документа используйте меню Файл - Создать или клавиши Ctrl+N. Inkscape открывает отдельное окно для каждого документа. Переключаться между окнами можно командой менеджера окон (обычно это Alt+Tab) или командой Inkscape Ctrl+Tab, которая циклически переключает рабочие окна.

Создание и редактирование фигур

Приведенные на рис. 4 инструменты называются инструментами создания фигур. Каждая вновь созданная фигура имеет белые маркеры в некоторых точках контура. С помощью этих маркеров можно изменять размеры и пропорции фигур.



Рис.4. Инструменты создания фигур

Верхняя панель содержит поля ввода для точной настройки параметров фигуры. Они (так же как и маркеры) изменяют ту фигуру, которая в данный момент выделена, а также определяют параметры, с которыми создаются новые фигуры.

- **Прямоугольник** - рисует прямоугольник, с нажатой клавишей Ctrl - квадраты.
- **Рисовать параллелепипеды** - рисует объемный объект.
- **Эллипс** - рисует дуги, замкнутые контуры - эллипсы, с нажатой клавишей Ctrl - круги.

- **Рисовать звёзды и многоугольники** - по умолчанию рисуется звезда с указанным количеством углов, если нужен многоугольник с таким количеством углов, необходимо поставить галочку перед словом «Многоугольник».
- **Рисовать спирали** - используется для рисования спиралей, параметры рисования (количество витков, нелинейность, внутренний радиус) варьируются на панели свойств инструмента.
- **Перо** - позволяет рисовать произвольные контуры, прямые и ломаные линии.
- **Кривая Безье** - кривая, позволяющая создавать рисунок из точек и соединяющих их прямых линий или кривых.
- **Рисовать каллиграфическим пером** - возможность создания каллиграфических надписей.
- **Создавать и править текстовые объекты** - используется для ввода и редактирования текста.

2. Алгоритм выполнения работы

- 1) Создайте новый файл в среде векторного редактора Inkscape.
- 2) Выполните работу «Городской пейзаж».
- 3) Сохраните изображение в своей папке.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как создать и отредактировать объект?
- 2) Как залить объект цветом, градиентом, текстурой?
- 3) Как размещаются объекты на странице?
- 4) Как перемещаются объекты?
- 5) Как объединить/разъединить группу объектов?

Практическая работа № 8

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Работа с текстом в Inkscape.

Цель изучения темы: Научиться размещать текст вдоль заданного пути.

Выполнение работы:

1. Теоретическое обоснование

В некоторых случаях возникает необходимость написания текста по круговой траектории. Рассмотрим эту функцию в среде векторного редактора Inkscape.

2. Алгоритм выполнения работы

- 1) Берём инструмент «Эллипс» и рисуем круг. Выключаем «Заливку» объекта, устанавливаем толщину обводки 5 px, цвет - чёрный. Снова выделяем объект и задаем ширину и высоту объекта по 400 px.
- 2) Копируем и вставляем объект два раза, меняя ширину и высоту первого до 380 px, второго - 280 px. Соединяем три круга, совмещая точку центра.
- 3) Берём инструмент «Текст» и щёлкаем в любом месте. Пишем текст ЮБИЛЕЙНАЯ МОНЕТА и г. Ростов-на-Дону (Шрифт: Courier New, Кегль: 40, Полуужирный).
- 4) Возьмите «Указатель» и выделите написанное, нажмите клавишу Shift и кликните малый круг. Далее перейдите в верхнее меню «Текст» - «Текст вдоль пути» («Разместить по контуру»).
- 5) Выделите текст отдельно, используя верхнее меню, аккуратно распределите буквы по эллипсу.
- 6) Сохраните файл как Монета, svg для дальнейшей работы.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как установить толщину обводки объекта?
- 2) Как изменить размеры объекта?
- 3) Как размещаются объекты на странице?
- 4) Как перемещаются объекты?
- 5) Как настраивается выравнивание текста?

Практическая работа № 9

Разработка программных модулей программного обеспечения для компьютерных систем

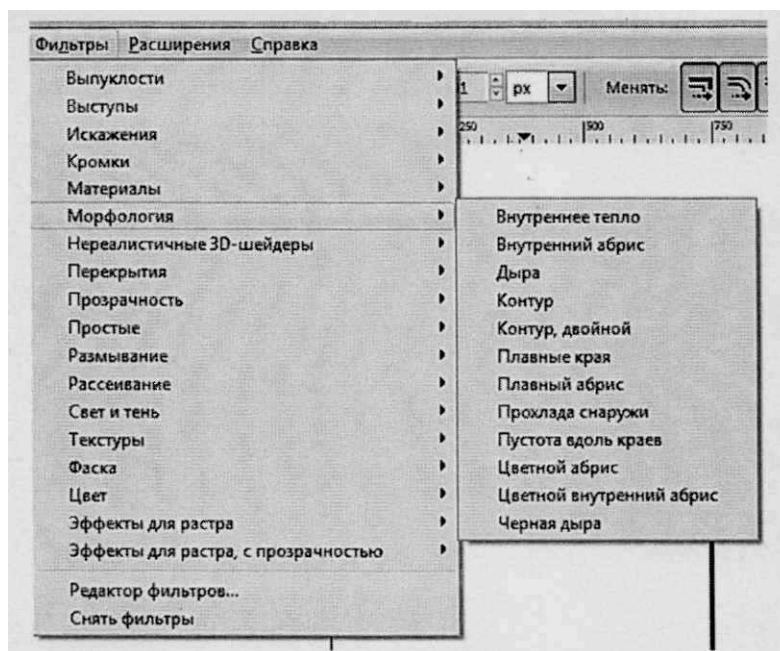
Тема: Специальные эффекты Inkscapе.

Цель изучения темы: Научиться работать с эффектами Inkscapе.

Выполнение работы:

1. Теоретическое обоснование

Векторный редактор Inkscapе имеет более 20 установленных и готовых к работе фильтров. Функция **Фильтр** вызывается на верхней панели инструментов. Каждый из предложенных пунктов меню имеет дополнительный набор спецэффектов (рис. 5).



2. Алгоритм работы

- 1) Откройте файл **Монета.svg**, выполненный на прошлом занятии. Заполните середину монеты новыми объектами, например,



Рис. 6. Монета с гербом.

герб Ростовской области (рис. 6).

2) Наложите на готовую монету эффект из меню «Фильтры»:

- текстура «Слой джема»
- морфология «Финифть»
- кромка «Металлизированная кромка»

4) Сохраните работу, экспортируйте файл в PNG.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Какие еще фильтры из раздела «Текстура» подойдут для работы?
- 2) Какие еще фильтры из раздела «Морфология» подойдут для работы?
- 3) Какие еще фильтры из раздела «Кромка» подойдут для работы?
- 4) Как импортировать изображение в готовый объект?
- 5) Как экспортировать готовый файл в растровое изображение?

Преподаватель

Н.С. Осмоловская

Практическая работа № 10

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Создание объемных объектов

Цель изучения темы: изучить способы создания объёма на плоскости.

Выполнение работы:

1. Теоретическое обоснование

Чтобы сделать изображение с 3D эффектом в InkScare, потребуется совсем немного инструментов - трансформация, сумма и разность контуров, и, что очень важно, градиент. В этом уроке техники применены к базовым контурам, тем не менее, полученный результат выглядит весьма эффектно.

2. Алгоритм выполнения работы

- 1) Рисуем в Inkscare эллипс «F5», для удобства задаем значение непрозрачности для него в 75 %. Дублируем полученный эллипс «Ctrl+D» и уменьшаем его размер при помощи инструмента трансформации «F1», удерживая клавиши «Shift+Ctrl».
- 2) Дублируем полученную группу из двух эллипсов «Ctrl+D» и на время перемещаем ее немного в сторону. Возвращаемся к первоначальной группе и снова дублируем ее. Зажав клавишу «Ctrl», перемещаем дубликат немного вверх.
- 3) Возвращаемся к группе из двух эллипсов, которую мы переместили в сторону, снова дублируем ее «Ctrl+D» и опять перемещаем в сторону. Затем выполняем операцию Контур→Разность «Ctrl-».
- 4) Выделяем два маленьких эллипса, полученных во 2-м шаге урока, дублируем их и в который раз перемещаем немного в сторону. С ними также выполняем операцию Контур→Разность «Ctrl-».
- 5) Выделяем два больших эллипса, полученных во 2-м шаге урока, дублируем их и перемещаем в сторону. Теперь нам нужно нарисовать прямоугольник. Его ширина будет равна ширине больших эллипсов, а высота - расстоянию между двумя эллипсами. Выделяем нижний эллипс и прямоугольник, после чего выбираем в меню Inkscare Контур→Сумма «Ctrl +». Перемещаем этот объект вниз. Выбираем оба этих объекта и выполняем Контур→Разность «Ctrl-».
- 6) Следующий этап - это собрать все 3 полученных объекта вместе. Сделать это несложно, увеличив масштаб отображения и используя диалог выравнивания «Shift+Ctrl+A».

- 7) Следующее, что нужно сделать - это немного раскрасить наши объекты при помощи инструмента Градиент в InkScare. Для первого объекта можно оставить плоскую заливку. Для второго объекта нам нужно назначить линейный градиент темный-светлый-темный. Для третьего объекта нам нужно назначить линейный градиент светлый-темный-светлый. Значение непрозрачности для всех трех объектов - 100 %.
- 8) После всех вышеприведенных действий между объектами может наблюдаться небольшая щель. Чтобы исправить это, назначаем обводке объекта те же самые значения цвета и градиента, что и для заливки.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как получить сумму/разность двух объектов?
- 2) Как объединить группу объектов?
- 3) Как изменить уровень расположения объекта?
- 4) Как выделить, повернуть группу объектов?
- 5) Как работает инструмент «Градиент»?

Практическая работа № 11

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Обработка растровых изображений.

Цель изучения темы: Применение навыков работы в Inkscare

Выполнение работы:

1. Теоретическое обоснование

Разработка рекламы заключается в создании логотипа и яркого запоминающегося изображения, которое можно разместить на консервной банке кошачьего корма.

2. Алгоритм выполнения работы

- 1) Создайте новый файл. Нарисуйте консервную банку из двух овалов и одного прямоугольника. Наложите градиент, чтобы придать банке визуальный объем.
- 2) Разрабатываем логотип «Кошачьей радости». Открываем файл «КОТ.jpg».
- 3) Выделяем фотографию кота, в меню «Контур» выбираем «Векторизовать растр». Нам нужен режим «Квантования цветов», множественное сканирование в цвете. Получаем цветное векторное изображение кота, которое и разместим на банке.
- 4) Удаляем ненужные части изображения, оставляя лишь морду животного.
- 5) Создаем этикетку на банке. Копируем основную часть банки, перекрашиваем в желтый цвет: dad318ff, немного сжимаем и равномерно распределяем по высоте банки.
- 6) Собираем все части изображения.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Как происходит векторизация растрового изображения?
- 2) Из скольких слоев состоит векторное изображение?
- 3) Как удалить часть векторного изображения?
- 4) Как редактировать группу объектов?
- 5) Как сохранить готовое изображение?

Практическая работа № 12

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Дизайн и планирование стартовой страницы

Цель изучения темы: Планирование стартовой страницы

Выполнение работы:

1. Теоретическое обоснование

Гипертекстовые документы в WWW представляются в виде Web-страниц (или, другими словами, Web-документов). Web-страница обычно содержит разнородную информацию: текст, графику, звуки и даже видео. Ссылки на Web-страницах отображаются выделенными словами или графическими изображениями. Если навести на ссылку указатель мыши, то он обычно превращается в изображение руки с поднятым указательным пальцем. Любая ссылка - это вход в другой документ. Web-страницы обычно организуются в так называемые сайты. Web-сайт (или Web-узел) - это совокупность Web-страниц (нескольких десятков, сотен или даже тысяч), объединенных одной общей темой и помещенных, как правило, на одном узлом компьютере.

Web-сайт принадлежит конкретному лицу или организации, которая его поддерживает (или финансирует его поддержку). Адреса Web-страниц, составляющих сайт, имеют общий префикс - это URL данного Web-узла

Гипертекстовые документы в World Wide Web размещаются на Web-серверах. Пользователь, получающий с помощью браузера доступ к ресурсам Интернета, называется клиентом (слово «клиент» относится, строго говоря, к программе Web-браузера, которая формирует запросы компьютера). Когда компьютер клиента стремится получить тот или иной файл из Web, его запрос отправляется на Web-сервер, который отвечает за данный файл. Сервер отыскивает затребованный файл в своих хранилищах и отправляет его копию на компьютер клиента.

Web-сервер - это компьютер, подключенный к Интернету, на котором выполняется специальная программа, обеспечивающая обработку запросов клиентов и выдающая клиентам различные страницы с информацией. Взаимодействие клиента с Web-сервером происходит согласно протоколу HTTP (HyperText Transport Protocol) - протокол передачи гипертекста. Чтобы отобразить на экране компьютера HTML-документы в том виде, как они задумывались разработчиками, необходимо иметь специальную программу, называемую браузером (по-русски - «обозревателем»).

Браузер - это программа просмотра информационных ресурсов WWW на компьютере, подключенном к Интернету. Современные программы браузеров работают в соответствии со спецификацией HTML, содержат интерпретаторы языков сценариев (VBScript, JavaScript), поддерживают динамическую объектную модель (Dynamic HTML) и т.д.

Браузер напрямую работает с объектной моделью, а не с предоставленными файлами. Используемая браузером модель называется Document Object Model. Она соединяет объекты, представляющие элементы, атрибуты и текст в ваших документах. В эту объектную модель внедрены все стили, значения вашего HTML-документа. Конкретная модель для данной Web-страницы называется DOM-деревом страницы.

Web-страница, загружаемая в браузер, представляет собой HTML-файл. Это текстовый ASCII-файл с расширением .htm, .html или .html1. Файлы .htm создаются для обработки в операционной системе DOS (где принято расширения обозначать тремя буквами). Для работы в Windows допустимо сохранять HTML-файлы с расширениями .htm, .html, причем более предпочтительным является .html. Если работа с файлами Web-документов будет выполняться в ОС Unix, можно использовать расширение .html1.

Документ, оформленный в соответствии со стандартом HTML, называется документом HTML (или HTML-документом). Такой документ состоит из содержимого и тегов, определяющих структуру и отображение документа. HTML-документ сохраняется в виде HTML-файла.

Поскольку документ Web представляет собой текстовый файл, для его редактирования можно использовать не только Блокнот, но и любой текстовый редактор.

В настоящее время существуют также более серьезные программы для подготовки Web-страниц, которые можно условно разделить на два типа:

- Визуальные редакторы HTML, то есть редакторы типа WYSIWIG (например, Microsoft FrontPage, Macromedia Dreamweaver, Netscape Navigator Gold и др.)- При работе в этих программах пользователь имеет дело с графическими образами элементов HTML, а не с кодом документа. Кстати, существуют конверторы текстовых документов, подготовленных в Word и Word Perfect, в формат HTML.

- Редакторы собственно HTML-текстов, позволяющие автоматизировать набор и редактирование кода (например, HomeSite, HotDog, Ken Nesbitt Web Editor и многие другие). Эти программы дают возможность пользователю быстро и легко вставлять в документ элементы HTML, проверять синтаксис команд, выполнять запуск и отладку страницы, не покидая окна редактора. Вообще работа в таких редакторах напоминает работу в интерактивной среде программирования типа Delphi или Visual Basic.

Кроме перечисленных двух категорий, существуют программы, объединяющие в себе черты текстовых и визуальных HTML-редакторов. К таким программам относится Hot-Metal разработки компании SoftQuad Software.

Разметка HTML-документов выполняется с помощью тегов, которые записываются с соблюдением определенных правил. Теговая модель предполагает разбиение документа на отдельные фрагменты, которые заключаются в теги или начинаются тегом.

В языке HTML для присвоения какому-либо элементу определенных свойств, таких как цвет, размер, положение на странице и т. п., приходится каждый раз задавать эти свойства для каждого объекта, даже если на одной странице должны располагаться десятки таких объектов, не отличающихся один от другого.

Эту проблему можно обойти с помощью каскадных таблиц стилей (CSS - Cascading Style Sheets), которые действуют другим, более удобным и экономичным способом. Основным понятием CSS является стиль, то есть набор правил оформления и форматирования, который может быть применен к различным элементам страницы. Для присвоения какому-либо элементу определенных характеристик вы должны один раз описать элемент и определить это описание как стиль, а в дальнейшем просто указывать, что элемент, который вы хотите оформить соответствующим образом, должен принять свойства описанного стиля.

Более того, вы можете сохранить описание стиля не в коде вашей web-страницы, а в отдельном файле. Это позволит использовать описание стиля для любого количества web-страниц. И еще одно преимущество - возможность изменить оформление любого количества страниц, исправив лишь описание стиля в одном файле.

Применить стиль можно, выбрав название в меню Text» CSS Styles (Текст * Стили CSS) или в соответствующем контекстном меню объекта. Но в этих случаях вы не увидите, как выглядит стиль.

Созданный стиль можно применить к любому текстовому фрагменту на текущей web-странице. Если бы мы создали внешнюю таблицу стилей, сохранив в файле формата CSS, то стиль можно было бы использовать в любом документе.

2. Алгоритм выполнения работы

- 1) Проработать материал лекции. Выбрать термины, определения.
- 2) Ответить на контрольные вопросы в письменном виде.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Что такое Гипертекст?
- 2) Где размещаются в сети Интернет гипертекстовые документы?

- 3) Что такое браузер? Какие браузеры вы знаете?
- 4) Как придать определённым элементам Web-страницы цвет, размер шрифта и т.д.?
- 5) Где и как используются каскадные таблицы стилей?

Практическая работа № 13

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Разработка простейшей Web-страницы.

Цель изучения темы: Разработать простейшую Web-страницу.

Выполнение работы:

1. Теоретическое обоснование

Основой Всемирной «паутины» World Wide Web является язык гипертекстовой разметки HTML. Создатель этого языка Тим Бернерс-Ли.

HTML не относится к языкам программирования высокого уровня, таким как C++, Java или Visual Basic. HTML это, прежде всего, язык разметки, и код, написанный на нем, исполняется на компьютере клиента в приложении Web-браузера.

В случае Web-страниц пользователь имеет дело с электронными документами, полученными -через Internet. Принцип отображения документа средствами форматирования родительского приложения здесь неприемлем. Слишком много приложений или всевозможных конверторов пользователю пришлось бы иметь на своем компьютере, чтобы эффективно работать с множеством возможных форматов документов. Идея решения проблемы обмена документами между различными компьютерами и приложениями через Internet основана на языке разметки гипертекста HTML (HyperText Markup Language).

Разметка HTML-документов выполняется с помощью тегов, которые записываются с соблюдением определенных правил. Теговая модель предполагает разбиение документа на отдельные фрагменты, которые заключаются в теги или начинаются тегом.

Тег (tag - указатель, метка) - это код, идентифицирующий определенный элемент документа HTML, например, абзац, ссылку, таблицу и т.д. Теги заключаются в угловые скобки < >.

Теги HTML являются подмножеством тегов языка SGML. Если какие-либо из тегов не понятны браузеру, они при анализе документа просто игнорируются.

```
<!DOCTYPE PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>Пример Web-документа</TITLE>
</HEAD> <BODY>
<P> Web-страница </P>
</BODY> </HTML>
```

Все теги начинаются с открывающейся угловой скобки <, за которой следует текст, определяющий содержание тега, например, TITLE или BODY. Оканчивается тег закрывающейся угловой скобкой >. Содержанием тега может быть просто имя тега либо имя и набор атрибутов тега.

Большинство тегов являются парными, то есть для каждого начального тега <Имя> есть конечный тег </Имя>, в котором к имени тега добавляется косая черта "/" (слэш).

Кроме парных, возможны одиночные теги, т.е. теги, в которых имеется только открывающий тег. В соответствии с инструкциями одиночных тегов браузер выполняет определенные действия. Например, согласно тегу
 выполняется разрыв текстовой строки, а в соответствии с тегом <P> формируется новый абзац. При наличии тегов <HR>, или <EMBED> производится вставка горизонтальной линии, изображения или звукового файла соответственно.

Имена всех тегов стандартизированы и удобны для запоминания. Например, имя тега <BODY> (в переводе «тело») обозначает основную часть документа, а тег <I> (сокращение от *italic* - курсив) задает начертание курсивом.

Имена в парных тегах должны быть одинаковыми. Отклонение от этого правила будет считаться ошибкой.

Пара тегов, состоящая из начального и конечного тегов, называется контейнером. Контейнеры обозначаются по имени начального тега и записываются в угловых скобках. Два выражения «пара тегов <TITLE></TITLE>» и «контейнер <TITLE>» обозначают одно и то же.

Документ HTML включает в себя элементы, которые представляют абзацы, заголовки, гиперссылки, списки, таблицы, рисунки и проч. Вообще весь документ можно рассматривать состоящим из определенных элементов.

Элемент - это пара тегов и символьные данные (текст или код), заключенные между ними. То есть элемент состоит из трех компонент: начального тега, содержимого и конечного тега. В некоторых элементах конечный тег может быть опущен (в случае одиночных тегов).

Элемент называется обычно по имени тега (без угловых скобок):

• <TITLE> Пример Web-документа </TITLE> - это элемент TITLE, обозначающий название документа;

- <P> Web-страница - элемент P, определяющий абзац.

Некоторые элементы могут не иметь содержимого, например, элемент разрыва строки BR. В таких элементах отсутствует конечный тег.

Описывая синтаксис элементов HTML, необходимо указать, что они не чувствительны к регистру символов, то есть браузер, одинаково воспринимает теги <TITLE>, <Title> или <title>.

Следует различать термины «элемент» и «тег». Элемент обязательно включает в себя хотя бы один тег и, возможно, содержимое, в то время как у тега содержимое отсутствует. Например, HTML-код <TITLE> Пример Web-документа </TITLE> представляет собой элемент title, а <title> - тег.

Все элементы, предусмотренные в HTML, можно условно разбить на несколько категорий:

• структурные - это элементы, которые обязательны для документа, соответствующего стандарту HTML (например, элементы HTML, HEAD, BODY и TITLE);

• блоковые - элементы, которые предназначены для форматирования целых текстовых блоков (например, элементы BLOCKQUOTE, DIV, H1, H2, H3, H4, H5, H6, P, PRE); часто блоковые элементы отделяются переводом строки от остального содержания документа;

• текстовые - элементы, которые задают разметку текста (EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ABBR, ACRONYM), разметку шрифта (I, B, U, TT, BIG, SMALL, SUB, SUP);

• специальные - элементы пустой строки (BR, HR, NOBR), якорный элемент A, вложенные элементы (EMBED, IMG, BGSOUND, OBJECT, MAP), элементы формы (INPUT, SELECT, TEXTAREA), элементы таблицы (TABLE) и др.

В документе HTML обязательно присутствуют вложенные элементы, то есть элементы, включенные в состав других элементов. Так, в примере из раздела «Ваша Web-страница» элемент TITLE вложен в элемент HEAD, а элемент P - в элемент BODY. Элементы HEAD и BODY, в свою очередь, являются вложенными в элемент HTML.

Последовательность, в которой допустимо размещать HTML-теги, то есть правила вложения элементов HTML, зависят от их категории. При составлении кода страницы соблюдайте следующие правила:

Структурные элементы могут включать в себя элементы других категорий. Например, элемент <HTML> является внешним по отношению ко всем остальным элементам;

Блоковые элементы могут включать в себя другие блоковые или текстовые элементы. Так, элемент TABLE, задающий таблицу, может содержать элементы заголовка H1 и разметки шрифта I, B;

Текстовые элементы также могут быть вложенными, но они не могут включать блоковые элементы. Например, недопустимо, чтобы элементы верхних индексов SUP содержали элементы абзаца или списков UL;

При составлении HTML-кода нужно следить за правильным написанием имен вложенных парных тегов. Для закрытия тегов лучше придерживаться последовательности: от последнего к первому. Понятно, что подобных проблем не возникает с одиночными тегами.

Элементы должны вкладываться друг в друга таким образом, чтобы каждый внутренний элемент располагался внутри одного и того же внешнего элемента.

Кроме перечисленных выше разновидностей элементов, в HTML предусмотрены пустые элементы. Такие элементы не включают в себя какие-либо текстовые фрагменты или другие элементы. Пустой элемент состоит только из одиночного тега, например, тег <HR> является в то же время элементом горизонтальной линии HR.

Часто теги, помимо имени, содержат дополнительные элементы, которые называются атрибутами, например, `id= "zagl">` Это заголовок, идентифицируемый как элемент "zagl" благодаря наличию атрибута id

Атрибут записывается после имени тега перед закрывающейся скобкой > и состоит, как правило, из пары «имя атрибута - значение». Однако возможны атрибуты, в которых значение отсутствует, например,

```
<TD NOWRAP>
```

Атрибуты - это компоненты тега, содержащие указания о том, как браузер должен воспринять и обработать тег. В объектных моделях атрибуты отвечают свойствам объектов.

В теге может быть несколько атрибутов. Тогда атрибуты отделяются друг от друга пробелами, символами табуляции или возврата к началу строки. Очередность записи атрибутов в теге значения не имеет. Атрибуты записываются только в начальных тегах и отсутствуют в конечных тегах.

Значение атрибута указывается после имени атрибута через знак равенства. Допускается применение пробела до и после знака =, например, записи `COLOR = "blue"`, `COLOR= "blue"` и `COLOR="blue"` будут восприняты браузером одинаково.

Все значения атрибутов по умолчанию должны заключаться в двойные (") или одинарные (') кавычки. В определенных случаях при записи значений можно обходиться и без кавычек, однако в руководствах W3C рекомендуется кавычки все же ставить.

В заключение рассмотрения синтаксиса HTML перечислим основные правила записи кода HTML-документов:

- каждый тег должен начинаться с открывающейся угловой скобки, а заканчиваться закрывающейся угловой скобкой;
- браузеры игнорируют нестандартные теги и атрибуты, поэтому имена тегов и их атрибутов должны соответствовать перечню допустимых в HTML имен;
- при записи имен тегов можно пользоваться как верхним, так и нижним регистром, но нельзя ставить пробелы;
- если значение атрибута содержит пробелы, оно обязательно должно быть заключено в кавычки;
- в содержимом документа браузеры игнорируют несколько пробелов, следующих подряд, и сжимают их до одного пробела;
- элемент, включающий в себя начальный тег другого элемента, должен содержать и конечный тег этого элемента (за исключением одиночных тегов);
- допускается включать в HTML-документы комментарии.

2. Алгоритм выполнения работы

- 1) Проработать материал лекции. Выбрать термины, определения.
- 2) Ответить на контрольные вопросы в письменном виде.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) К каким языкам относится HTML?
- 2) Как выполняется разметка HTML-документов?
- 3) Что такое тэг? Чем тэг отличается от атрибута?
- 4) Как записываются тэги?
- 5) Сколько атрибутов может быть у тэга? В каком порядке они записываются?

Практическая работа № 14

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Оформление web-страницы.

Цель изучения темы: Правила оформления web-страницы, добавление изображений.

Выполнение работы:

1. Теоретическое обоснование

Использование изображений в Web-страницах, безусловно, улучшает восприятие документов, однако чрезмерное увлечение графикой имеет отрицательную сторону. Связано это в первую очередь с тем, что графические изображения довольно медленно загружаются. Пользователь, обладающий не очень хорошим модемом или подключенный к Интернету через плохую линию связи, может ждать окончания загрузки изображений несколько минут. Понятно, что медленный вывод изображений на экран будет его раздражать, и вряд ли он захочет повторно посетить такой Web-сайт. Поэтому при использовании графики руководствуйтесь следующими рекомендациями.

Количество графической информации, размещенной на Web-страницах, нужно сводить к минимуму. Это касается как количества изображений, так и их размера.

Необходимо использовать возможности по ускорению открытия страницы. Имеется в виду предварительная загрузка уменьшенного варианта изображения, чересстрочная развертка, а также альтернативный текст на случай отключения вывода изображений в браузере пользователя.

При размещении большого количества изображений, придерживайтесь общего правила: крупные изображения, должны находиться на страницах низшего уровня, посвященных специальным и узким вопросам. На главной странице сайта должны присутствовать только небольшие изображения.

Данные об изображениях хранятся в файлах графических форматов. Существует достаточно много форматов, обеспечивающих организацию графических данных (например, BMP, EPS, GIF, JPEG, PCX, TIFF и множество других). Но посетитель вашей страницы увидит графическое изображение лишь в том случае, если его браузер поддерживает данный графический формат. Поэтому изображения создаются в тех форматах, которые понимают большинство браузеров. К таким форматам относятся следующие:

- GIF (сокращение от Graphics Interchange Format- графический формат обмена), имеет расширение .gif;
- JPEG (сокращение от Joint Photographic Experts Group - объединенная группа экспертов фотографии), имеет расширение jpg или jpeg;
- PNG (сокращение от Portable Network Graphics - переносимая сетевая графика), имеет расширение .png.

Формат GIF. Является стандартом для Web-графики, причем без потери качества достигается высокая степень сжатия. GIF обладает замечательными свойствами, которые позволяют создавать прозрачные области на изображении и назначать воспроизведение изображений в режиме чересстрочной развертки. С помощью этого формата в одном файле можно хранить несколько изображений и создавать простые анимации. Восьмибитовая схема кодирования цвета, принятая в GIF, ограничивает глубину цвета (всего $2^8=256$ цветов). Поэтому в этом формате обычно сохраняются простые рисунки с небольшим количеством цветов. Однако имеются некоторые

формальные ограничения на использование GIF, поскольку чрезвычайно эффективный алгоритм сжатия данных, примененный в этом формате, защищен патентом.

Формат JPEG. Подобно формату GIF считается стандартом для Web-графики и обеспечивает высокую степень сжатия. Поскольку в этом формате принята 24-битовая схема кодирования цветов, в нем можно сохранять почти семнадцать миллионов цветов ($2^{24} = 16777216$) и получать высокое качество проработки деталей изображения. Этот формат позволяет достичь компромисса между степенью сжатия и качеством изображения (с увеличением качества уменьшается степень сжатия и наоборот). JPEG часто применяется для изображений с высоким качеством проработки деталей (например, фотографии или графика с большим количеством цветов).

Формат PNG. Этот формат обладает всеми возможностями формата GIF, а также рядом дополнительных достоинств. Во-первых, PNG поддерживает очень «глубокий» цвет (глубина цвета - 48 бит). Во-вторых, файлы PNG более компактны: в них выполняется сжатие изображения по вертикали и по горизонтали. Наконец, данный формат не имеет тех юридических сложностей, что формат GIF. Единственным недостатком PNG является то, что этот формат пока поддерживается не всеми браузерами.

Описание цветов в HTML

В HTML для описания цветов используются числовой и символьный форматы цветовой модели RGB. Цветовая модель - это способ описания цвета. Например, цвет может быть задан количественным значением базовых цветов в его разложении. Для описания цветов, <воспроизводимых на мониторе, применяется цветовая модель RGB (от названий основных цветов: Red (красный), Green (зеленый), Blue (синий)). В полиграфии принята модель CMYK с компонентами Cyan (голубой), Magenta (пурпурный), Yellow (желтый) и Black (черный). Существует также цветовая модель HSV, основанная на восприятии цвета человеком. В ней для количественного описания приняты цветовой тон (Hue), насыщенность (Saturation) и яркость (Value).

В RGB-модели цвет можно кодировать с помощью трех пар шестнадцатеричных чисел. Каждая пара определяет «количество» основного цвета в интервале от 00 до FF. Первое число определяет «долю» красного цвета, второе число - зеленого цвета, а последнее - задает синюю компонент. Например, темно-фиолетовый цвет задается кодом - 800080. Значение цвета начинается символом #, а запись шестнадцатеричных чисел не зависит от регистра букв (например, браузер одинаково отобразит белый цвет, установленный СИМВОЛАМИ #FFFFFF и #ffffff).

Цвет фона всего документа определяется атрибутом bgcolor тега <BODY>. Например, тег <BODY bgcolor = "#ffffff"> назначает одинаковую, максимально возможную насыщенность красного, зеленого и синего, в результате чего браузер устанавливает белый фон страницы.

Для определения цвета шрифта в HTML-документе вы можете воспользоваться одним из трех способов:

- назначение цвета текстовых символов всего документа осуществляется атрибутом text тега <BODY>; например, синий цвет текста устанавливается тегом <BODY text="blue">
 - если в HTML-документе определен основной шрифт с помощью элемента BASEFONT, то можно дополнить этот элемент атрибутом color, например:
 - <BASEFONT size=5 color="red">
 - для текстовых фрагментов цвет шрифта задается с помощью атрибута color в тегах , например, отображение текста красным шрифтом можно задать одной из следующих инструкций:
 - ТЕКСТ или ТЕКСТ
- Введение атрибута color в тег FONT отменяет цвета, заданные атрибутом text в элементе BODY.

Вставка изображения на Web-страницу

Процедура вставки изображения на Web-страницу достаточно проста и выполняется одиночным тегом , который может быть размещен в любом месте элемента body.

При этом URL-адрес изображения задается значением атрибута src (название происходит от слова source - источник). Данный атрибут относится в числу обязательных в теге .

Допустим, вы хотите разместить на странице картинку puzzle.gif с изображением «пазлов». Для этого сохраните файл с изображением в определенной папке (например, в той же папке, что и сам HTML-документ), а в документ введите тег .

```
<HTML>
<HEAD><TITLE>Пазлы</TITLE></HEAD>
<BODY>
<H2>Вставка изображения</H2>
<IMG src="puzzle.gif" >
</BODY>
</HTML>
```

2. Алгоритм выполнения работы

- 1) Проработать материал лекции. Выбрать термины, определения.
- 2) Ответить на контрольные вопросы в письменном виде.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Какие форматы графических файлов можно размещать на Web-страницах?
- 2) Каковы преимущества и недостатки этих форматов?
- 3) Как вставить изображение в HTML-страницу?
- 4) Как изменить цвет шрифта на странице?
- 5) Где взять цвет в RGB-модели?

Практическая работа № 15

Разработка программных модулей программного обеспечения для компьютерных систем

Тема: Использование шаблонов для страниц web-сайта.

Цель изучения темы: Использование шаблонов для страниц web-сайта

Выполнение работы:

1. Теоретическое обоснование

Существует множество вариантов использования табличного дизайна. Ведь язык разметки HTML имеет много ограничений. А таблицы нередко помогают обходить эти ограничения. Создадим каркас документа, который в дальнейшем с некоторыми дополнениями можно будет использовать в качестве шаблона web-страниц.

Наиболее часто web-страница состоит из следующих блоков: заголовка, блока гиперссылок, блока основного содержимого страницы, блока новостей и строки с авторскими правами. Для указанных блоков мы и создадим каркас.

Создадим таблицу из 3 строк и 3 столбцов. Установим для таблицы относительную ширину 100%. Рамку таблицы зададим толщиной 3 пикселя. Интервал между ячейками также установим равным 3 пикселям.

Для блока заголовка объединим ячейки первой строки.

В ячейках второй строки будут расположены слева направо: блок гиперссылок, блок основного содержимого страницы и блок новостей. Чтобы ширина блоков гиперссылок и новостей не зависела от ширины окна обозревателя Интернета, мы сделаем эти размеры фиксированными.

Когда мы указываем числовое значение в поле ввода W (Ширина), ширина выделенной ячейки становится фиксированной. Таким образом, у первой и третьей ячеек второй строки таблицы ширина зафиксирована. А ширина второй ячейки, в которой будет расположено основное содержимое страницы, будет устанавливаться обозревателем Интернета в зависимости от ширины окна так, чтобы таблица заполняла все окно.

Создавая web-страницы на основе табличного дизайна, следует иметь в виду, что стандартная объемная рамка таблицы будет смотреться не очень хорошо. Вы можете вообще отказаться от рамки, а можете сделать рамку таблицы необъемной.

Ячейки таблиц могут содержать вложенные таблицы, графические изображения и другие объекты. Достоинство табличного дизайна в том, что все элементы находятся в четко определенных позициях, независимо от размеров окна обозревателя Интернета. Например, если колонка новостей должна быть справа, то при уменьшении ширины окна обозревателя Интернета эта колонка ни при каких обстоятельствах не окажется снизу.

Основное содержимое web-страницы можно ввести во вторую ячейку второй строки точно так же, как если бы вы работали на пустой странице. То есть вы можете создавать в ячейке таблицы, списки, гиперссылки, вставлять рисунки и другие объекты.

Таблицы имеют и недостатки. Главной проблемой таблиц является запутанность и громоздкость кода HTML. Другой, не менее важной проблемой, может стать медленная обработка таблиц обозревателем Интернета. Например, программа Internet Explorer не отображает таблицу до завершения загрузки из Интернета на компьютер пользователя, а многие пользователи пока еще используют медленное соединение с Интернетом посредством модема. В случае объемной таблицы таким пользователям придется ждать загрузки сайта со сложными таблицами очень долго. Поэтому не следует создавать на web-странице множество таблиц или очень громоздкие и объемные таблицы. Кроме того, старайтесь устанавливать значения ширины или высоты таблиц и ячеек фиксированными, чтобы избавить обозревателя Интернета от большого объема вычислений, связанных с подгонкой ширины ячеек под содержимое.

Вы, вероятно, обратили внимание на то, что в Интернете web-страницы больших сайтов оформлены в едином стиле. Но это не значит, что, добавляя новую web-страницу, разработчику приходится каждый раз создавать одни и те же повторяющиеся элементы, например набор гиперссылок или заголовков с названием сайта и логотипом. Конечно, если необходимо добавить всего лишь одну страницу, то ничего затруднительного в этом нет. Но если необходимо добавить несколько десятков страниц, то большая часть времени будет затрачена на создание одинаковых для всех страниц элементов.

Предположим, мы хотим использовать в качестве шаблона созданный нами на основе таблицы каркас web-страницы. Для этого необходимо добавить в документ неизменяемое содержимое, то есть те элементы, которые будут одинаковыми для всех web-страниц, использующих данный шаблон. В нашем примере такими неизменяемыми элементами будут заголовок сайта, блок гиперссылок и блок сведений об авторских правах. Эти блоки следует заполнить фактической информацией.

Теперь вы можете добавить на новую web-страницу содержимое. Обратите внимание, что редактировать можно только данные в изменяемых областях.

Использование шаблонов позволяет быстро создавать и редактировать web-страницы сайта. Благодаря шаблонам вы легко сможете обновлять дизайн своих web-страниц, не давая информации устареть, а оформлению наскучить.

2. Алгоритм выполнения работы

- 1) Проработать материал лекции. Выбрать термины, определения.
- 2) Ответить на контрольные вопросы в письменном виде.

3. Анализ полученных результатов

Совместно с преподавателем проанализируйте полученные результаты. Отметьте критерии, по которым проекты являются удачными или неудачными.

4. Контрольные вопросы

- 1) Для чего используют шаблоны на Web-страницах?
- 2) Из каких блоков чаще всего состоит на Web-страница?
- 3) Можно ли вставить изображение в готовый шаблон? Использовать в качестве шаблона?
- 4) Какие недостатки имеют шаблонные таблицы?
- 5) Как можно использовать каркас-шаблон, созданный на основе таблицы Web-страницы?