МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ПРИОРИТЕТНЫЕ НАЦИОНАЛЬНЫЕ ПРОЕКТЫ «Образование»

ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ

Н.Ф. Гусарова, Ю.В. Дорогов, Р.В. Иванов, А.В. Маятин

ИЗДАТЕЛЬСКИЕ СИСТЕМЫ. КОМПЬЮТЕРНАЯ ИЗДАТЕЛЬСКАЯ ГРАФИКА

Часть 1

Учебное пособие

ИТМО
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Санкт-Петербург
2007

Издательские системы. Компьютерная издательская графика. Часть 1: Учебное пособие / Н.Ф. Гусарова, Ю.В. Дорогов, Р.В. Иванов, А.В. Маятин. - СПб: СПб ГУИТМО, 2007. - 48 с.

В пособии рассматриваются основные модели, подходы и методики для решения задач подготовки полиграфической продукции с использованием настольных издательских систем. Первая часть пособия состоит из трех разделов: подготовка текстовых материалов, подготовка векторной графики и многополосная верстка. По каждому разделу предлагается цикл лабораторных работ, позволяющий на практике познакомиться с основными методиками перевода, преобразования, структурирования, стилевого оформления и корректуры текстовой информации; моделями создания и преобразования объектов векторной графики; подходами к реализации многополосной верстки. Циклы демонстрируют не только отдельные этапы подготовки издательской продукции, но и взаимосвязь между ними. Каждая лабораторная работа предваряется кратким теоретическим материалом, более полно раскрываемым в курсе лекций.

Для студентов специальности 230201 - «Информационные системы и технологии» и специальности 230202 - «Информационные технологии в образовании»

Рекомендовано к печати Ученым советом естественнонаучного факультета от 08.05.07, протокол №8.

Рекомендовано к печати Ученым советом факультета информационных технологий и программирования от 27.05.07, протокол №10.

В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ. КОМПЬЮТЕРНЫЕ ИЗДАТЕЛЬСКИЕ ТЕХНОЛОГИИ И СПЕЦИФИКА ИХ ПРИМЕНЕНИЯ В ИТ

Издание – произведение печати, самостоятельно оформленное, прошедшее редакционно-издательскую подготовку, имеющее установленные выходные сведения и предназначенное для передачи содержащейся в нем информации [2].

Издательский процесс – одна из самых «старых» технологий в истории человечества. Основной его целью всегда было тиражирование (многократное воспроизведение) однажды созданных текстов. Поэтому основные задачи издательского процесса, по существу, остаются неизменными:

1. создание текста и графических материалов к нему;
2. перевод их в форму, удобную для тиражирования;
3. собственно тиражирование.

Для решения этих задач изобретались все новые технологии – переписывание от руки, полиграфические процессы (вырезание зеркальной копии будущей страницы на деревянной доске, вытравливание ее на металлической пластине, набор из отдельных литер) и т.д., но сами эти задачи до появления компьютера были разделены во времени и пространстве. Появление компьютера не только внесло изменения в процессы решения каждой задачи по отдельности, но и позволило зачастую решать их на одном рабочем месте. Благодаря развитию компьютерных технологий возникла возможность коренным образом изменить весь издательский процесс – сделать его менее трудоемким, более оперативным и дешевым.

Около 15 лет назад возникло понятие «компьютерные издательские технологии» (КИТ). В самом широком смысле КИТ – подготовка изданий средствами настольных издательских систем.

Издательская система – комплекс, состоящий из персональных компьютеров, сканирующих, выводных и фотовыводных устройств, программного и сетевого обеспечения, используемый для набора и редактирования текста, создания и обработки изображений, верстки и изготовления оригинал-макетов, фотоформ, цветопроб.

Хотя компьютеры сегодня находят применение во всем издательско-полиграфическом процессе, в том числе и непосредственно в тиражировании, понятие КИТ связывается, главным образом, с редакционным и допечатными процессами.

Редакционный процесс включает в себя креативную часть издательского процесса – замысел издания, создание, подбор и редактирование текста и иллюстративных материалов, проект оформления (разработка сетки издания).

Допечатные процессы – все стадии полиграфической технологии, связанные с подготовкой издания к печати (набор, цветоделение, обработка текста и изображения, верстка полос издания, монтаж и раскладка полос на печатном листе), до изготовления печатной формы включительно.

Исходно понятие КИТ распространялось только на книги и другую полиграфическую продукцию. Сейчас наблюдается постепенное расширение сферы распространения КИТ до полной информационной поддержки бизнеса:

- по номенклатуре продукции – флайерсы, плакаты, буклеты, отчеты, брошюры, книги, рекламные материалы, периодические издания, техническая документация по ПО – вплоть до концепции фирменного стиля целиком;
- по средам публикации – твердые копии, сетевые издания, Интернет-ресурсы.

Более того, в КИТ оформилось понятие «контент» – материалы, представляемые в различной форме (текстовые, графические, мультимедиа), которые являются основой для публикации.

Таким образом, от специалиста в области ИТ, особенно работающего в небольшой компании, сегодня «по умолчанию» ожидается не только владение достаточным набором КИТ, но и умение реализовать весь технологический цикл информационной поддержки бизнеса – от разработки и оформления контента до сопровождения полиграфического процесса и/или Интернет-публикации. Именно таким образом построен курс «Компьютерные издательские технологии».

В соответствии с концепцией и программой курс ставит своей целью:

- показать источники формирования контента и способы их использования для информационной поддержки бизнеса и/или создания технической документации;
- познакомить студентов с правилами, требованиями и рекомендациями к подготовке текстов, иллюстраций, оформлению печатных изданий;
- показать возможности применения современных пакетов компьютерных программ в информационной поддержке бизнеса и дать необходимые практические навыки их применения;
- дать необходимые представления о композиции печатного издания, об основных способах и приемах его оформления средствами компьютерной графики;
- научить создавать графические элементы оформления – рисунки, схемы, рекламные блоки, а также производить необходимую подготовку фотографического иллюстративного материала для дальнейшего использования в публикации;
- дать необходимые практические навыки по компьютерной верстке различных периодических и разовых изданий;
- поддерживать интерес у студентов к овладению новыми компьютерными технологиями, привить навыки к самостоятельному овладению новыми программами.

В настоящем учебном пособии рассматриваются такие разделы курса, как подготовка текстовых материалов, подготовка векторной графики и верстка печатных изданий. По каждому разделу предлагается цикл лабораторных работ (ЛР), предваряемых кратким теоретическим материалом. Литературные и Интернет-источники, использованные при составлении конкретных ЛР, приводятся в каждой из них. Кроме того, в конце пособия приводится список литературы по курсу в целом.

РАЗДЕЛ 1. ПОДГОТОВКА ТЕКСТОВЫХ МАТЕРИАЛОВ

Текст является основой любого контента, который приходится подготавливать специалисту по информационным технологиям (ИТ). Это может быть техническая документация по разрабатываемому программному обеспечению, отчет, коммерческое предложение, флайерс, буклет фирмы, брошюра, рекламные материалы, периодическое издание, наполнение страницы сайта и т.д.

Реалии современного бизнеса, в особенности среднего и мелкого, таковы, что для написания текстов достаточно редко приглашается профессионал – технический писатель, журналист, редактор и т.п. ИТ-специалисту часто приходится не только обрабатывать уже готовый текст, подготавливая его к печати, но и составлять текст самому.

Проблемами составления текста для успешной коммуникации занимается специальная наука – риторика. Классическая риторика, основы которой заложил Цицерон, была направлена на подготовку устного публичного выступления и включала пять основных этапов подготовки: изобретение мыслей, расположение, словесное выражение, запоминание, произнесение. К разработке письменного текста относятся первые три этапа:

- изобретение – первый этап, на котором в общих чертах составляется замысел текста, общая логика изложения;
- расположение – следующая стадия, на которой автор подробно прорабатывает план будущего текста и распределяет по нему, «располагает» материал.
- словесное исполнение – написание самого текста, подбор конкретных слов и выражений, шлифовка стиля.

Такая «трехчленная» методика подготовки текста используется столетиями и вполне доказала свою продуктивность.

Эта схема положена в основу структуры лабораторных работ, составляющих первую часть курса. Так, ЛР1 посвящена разработке общей логики изложения, а ЛР2 – рационального расположения информации в тексте. Эти вопросы актуальны для формирования технического текста на любом языке. ЛР3 и ЛР4 связаны с проблемами стиля технической документации, поэтому здесь много рекомендаций, имеющих отношение к русскому языку, специфичным для него возможностям и ограничениям, а также к использованию англоязычных источников при составлении русскоязычного контента. Наконец, ЛР5 нацелена на окончательную шлифовку получившегося текста – его корректуру.

Когда все тексты уже есть, начинается редактирование.

Важный момент: классический редактор – это не только тот человек, который просто прочитывает текст и пытается привести его в «приличный» вид, это еще и тот человек, который дает рекомендации верстальщику по использованию шрифтов, заголовков, курсивов, по подписям к рисункам, он задает довольно много аспектов, верстки издания. Сейчас, к сожалению, очень мало пользуются ГОСТами на издательскую продукцию.

Следующие два этапа зависят от того, какое издание формируется – это создание таблиц и создание формул.

В зависимости от того, какая технология будет использоваться потом для верстки издания, и таблицы, и формулы необходимо представлять в различных форматах и по-разному подходить к их разработке. «Перебросить» таблицу из формата в формат очень тяжело.

В какой-то мере это касается и формул. Если Word и Pagemaker поддерживают вставку объектов Microsoft Equation, то до сих пор многие издательства при верстке технической литературы используют TEX или LaTEX – специально созданные редакторы, которые позволяют очень хорошо работать с формулами. Преобразование Equation – LaTEX невозможно. Поэтому по формату формул необходимо согласование с другими участниками технологического процесса.

Когда все это сделано, можно говорить о корректуре. Процесс корректуры сложен, и не имеет стабильного места в издательском процессе. Как правило используется две или даже три корректуры.

1-я корректура делается непосредственно после того, как все текстовые материалы набраны, прошли через руки редактора и сложены в некую последовательность, на этом этапе корректор вычитывает ошибки. 2-я корректура происходит после верстки. Здесь выявляются не только ошибки по тексту, но и ошибки верстки. Наконец, 3-я корректура производится в крупных издательствах на уровне вывода пленок, перед печатью, так как и на этом этапе возможно возникновение ошибок. Некоторые пленки при этом выводятся повторно.

Лабораторная работа 1. Преобразование материалов текстового форума в линейный текст

Цель работы: научиться выстраивать общую логику изложения материала.

Интернет-форум, где происходит общение профессионалов, часто становится источником таких сведений, которые невозможно найти в учебной или справочной литературе. Поэтому профессиональные форумы – очень ценный источник информации для подготовки контента технической направленности. Но эту информацию нужно надлежащим образом реструктурировать.

От автора (или составителя) технического текста ожидается не самовыражение (как это часто бывает на форуме), а, в первую очередь, взаимодействие с читателем, то есть потенциальным адресатом текста. Соответственно, до начала работы над текстом нужно ответить на три вопроса:

- кто будет использовать текст,
- для чего будет использоваться текст,
- как будет использоваться текст.

С каждым вопросом связывается свой аспект проблемы создания технического текста.

Аспект 1. Чтобы содержащаяся в тексте информация была воспринята адресатом, нужно, чтобы она встроилась в его «картину мира».

В случае технического текста адресат, как правило, определяется через целевые группы. Возможны различные целевые группы. Например, для текстов по тематике «ИТ и программирование» можно выделить такие группы:

- малоквалифицированные пользователи / продвинутые пользователи /профессионалы…;
- дети / подростки / взрослые /пожилые…;
- группы со специальными интересами (хакеры, геймеры, администраторы, узкие специалисты в определенной области ИТ…);
- контролирующие лица (члены ГАК, представители рекрутингового агентства, потенциальные работодатели…)
- другие группы.

Нужно учитывать, что потенциальный адресат текста может одновременно принадлежать к нескольким целевым группам.

Вариант решения – «говорить с адресатом на его языке», то есть использовать при создании текста его (адресата) логику, стиль и языковые средства.

Аспект 2. Цели написания и прочтения текста не совпадают.

Цели использования текста можно разделить по двум основаниям:

- цели автора / читателя,
- цели когнитивные (познавательные, информационные) / аффективные (эмоциональные).

Автор и читатель текста, как правило, имеют разные задачи, решаемые при чтении текста.

Примеры целей автора:

- убедить адресата в положительных / отрицательных качествах описываемого объекта;
- передать ему свое понимание структуры объекта;
- другие цели.

Примеры целей читателя:

- получить недостающую (для него, то есть в его собственном структурировании) информацию об объекте;
- оценить объект в ряду аналогичных;
- другие цели.

Вариант решения – начать текст с постановки цели, актуальной для адресата, а далее в тексте постоянно подчеркивать полезность (применимость, эффективность) описываемого объекта для достижения этой цели.

Когнитивные и аффективные цели читателя часто конкурируют друг с другом.

Примеры аффективных целей читателя:

- облегчить выполнение своей практической задачи (заверстать красивый отчет, найти недостающую функцию…);
- освоить новую, интересную или престижную сферу деятельности (новую игру, модную «примочку»…);
- получить подтверждение своей значимости («я в состоянии это освоить, а другой – нет»)
- другие цели.

Примеры когнитивных целей читателя:

- получить информацию об устройстве и принципах работы объекта;
- связать ее с уже имеющейся информацией;
- другие цели.

Чего хочет читатель технического текста? На сознательном уровне – получить новую (полезную для себя) информацию, а на подсознательном уровне – получить положительные эмоции. К сожалению, эти цели во многом противоречат друг другу:

- получение из текста новой информации требует познавательных усилий, что у большинства читателей сопровождается негативными эмоциями (достаточно вспомнить собственное состояние при подготовке к экзамену);
- если читаемый текст не содержит новой для читателя информации, он также вызывает отрицательные эмоции (скука, ощущение потерянного времени и т.д.);
- попытка снижения когнитивных целей переводит текст из технического в рекламный;
- читатель, даже самый мотивированный, практически не влияет на свое подсознание. Если у читателя сформировался негативный эмоциональный образ текста, то его крайне трудно перекрыть даже самой высокой информативностью.

Варианты решения:

- использовать ожидаемую читателем структуру текста (см. аспект 3);
- любой раздел текста «заключать в аффективную рамку», то есть стараться привязать начало и конец раздела к аффективным целям читателя;
- по возможности выносить узкоспециальную информацию – например, описание внутренней функциональной структуры программного продукта, программистских решений и т п. – из линейного текста (в приложение, во врезки и т.д.);
- максимально использовать графический материал (схемы, иллюстрации и т.п.).

Аспект 3. Читатель использует текст не так, как планировал автор.

Существуют различные сценарии поиска информации и, соответственно, использования технического текста. Например, при разработке технической документации на программный продукт, как правило, предусматриваются следующие сценарии ее использования:

- «быстрый старт», то есть поиск сведений об установке, настройке, запуске и начале работы с программным продуктом;
- ознакомительное чтение, дающее возможность освоить работу с продуктом в целом и способы решения наиболее типичных практических задач;
- целевой поиск путей решения той или иной конкретной практической задачи с помощью продукта;
- запрос и получение исчерпывающей справочной информации по избранному режиму(ам), элементу(ам) интерфейса, функции(ям) и т.д.;
- детальное знакомство с программным продуктом и нюансами его работы.

Как правило, реализовать одновременно различные типы сценариев удается только на достаточно большом тексте. Поэтому современная техническая документация – это либо набор документов, либо набор разделов, соответствующих различным сценариям, с частичным дублированием информации. Облегчить одновременную реализацию нескольких сценариев помогает гипертекстовая структура документа, а также одновременное использование нескольких информационных каналов (устный текст + раздаточный материал, устный текст + презентация, письменный текст + CD и т.д.).

Вариант решения для небольшого текста: для целевой группы спрогнозировать типовой сценарий и строить логику изложения материала по нему.

Приняв решение по каждому из рассмотренных аспектов, можно составить план будущего текста. Примеры таких планов приводятся ниже.

Вариант 1 (типичен для статьи).

- Концептуализация – описание проблемы и способа ее решения на естественном языке.
- Формализация – описание проблемы и способа ее решения на уровне модельного представления (например, в виде математической модели, архитектуры программного продукта, структурной схемы и т.п.).
- Реализация – описание конкретного решения проблемы (например, в виде конкретных процедур, фрагментов программы и т.п.).

Вариант 2 (типичен для отчета или пояснительной записки).

- Введение – в чем состоит проблема, почему и для каких потребителей важно ее решение.
- Аналоги – какие варианты решения проблемы уже существуют, в чем их недостатки.
- Постановка задачи – что конкретно будет сделано для решения проблемы в данной работе.
- Основная идея решения – обобщенное описание предлагаемого в работе способа решения, как правило, в виде структурной схемы с пояснениями.
- Характерные особенности решения – детальное описание наиболее интересных и/или нестандартных частных решений.
- Примеры применения – описание реальных результатов опробования на различных уровнях (технического тестирования, тестирования юзабилити в фокус-группах, полномасштабного внедрения и т.д.).
- Заключение – перспективы использования предлагаемого решения.

Вариант 3 (типичен для полного описания программного продукта).

- Вводные главы – краткое описание основных задач, решаемых продуктом, и его отличий от аналогичных продуктов.
- «Быстрый старт» – описание установки и начала работы с продуктом.
- Руководство пользователя – описание сценариев решения наиболее типичных задач.
- Справочник пользователя – детальное разъяснение составных частей, функций, элементов интерфейса программного продукта.
- Специальные главы – описание узкоспециализированных или вспомогательных процедур (например, экспорт, импорт, поиск файлов и т.п.).

В целом, независимо от конкретного плана, логика организации текста должна основываться на двух базовых принципах:

- от поставленной цели – к способу ее достижения.
- от более общего уровня описания – к более частному и детализованному.

Задание лабораторной работы №1:

1. ***Выбрать законченный фрагмент текстового форума по тематике ИТ.***
2. ***Выбрать целевую аудиторию, для которой будет составляться текст.***
3. ***Определить цели использования текста.***
4. ***Определить сценарий использования текста.***
5. ***Составить план текста.***
6. ***Составить текст.***
   Объем исходного форума не ограничивается, объем составленного текста – 2000–3000 знаков.

Представляемые к защите материалы

1. ***Исходный фрагмент форума (в распечатанном виде).***
2. ***Ответы на пп. 2–5 (в распечатанном виде).***
3. ***Составленный текст (в распечатанном виде).***

Лабораторная работа 2. Структурирование текста

Цель работы: научиться организовывать рубрикатор и выделения в тексте.

Любой текст представляется более понятным, если он разбит на разделы разного уровня, что позволяет локализовать любую информацию в общем ее массиве. С этой целью используется рубрикация текста.

Рубрикация – это деление текста на разделы, части, главы или иные более мелкие составные элементы, имеющие заглавную надпись – рубрику. Под рубрикацией понимается также такая система заголовков, в которой заголовки разной значимости занимают разные уровни (от высшего через промежуточные к низшему), а заголовки одной значимости стоят на одном уровне. Количество таких уровней называется глубиной рубрикации.

Сегодня для текстов характерны две системы рубрикации:

- иерархическая,
- гипертекстовая.

В первом случае текст организуется в виде строгого иерархического дерева, во втором имеет структуру, схожую с реляционной базой данных.

Исторически первой (еще в средние века) появилась иерархическая рубрикация. Она лучше соответствует возможностям линейного расположения текста, а также привычна европейской культурной традиции. Гипертекстовая рубрикация завоевала признание с появлением сетевых носителей. Она применима тогда, когда пользователь четко знает, что он ищет, и ему понятна и удобна система

атрибутов, по которым он проводит поиск. На практике оба подхода приходится сочетать. Тем не менее, для классических научно-технических текстов до сих пор характерна иерархическая рубрикация, которая лучше отражает логику исследования и хода мысли автора.

Однако читатель не всегда хочет следовать за логикой изложения по всему пути ее развертывания. Логическая последовательность весьма авторитарна и навязывает читателю совершенно определенную модель поведения, требует от него внимания, усидчивости и большого временного ресурса. Реальный же читатель (здесь мы не говорим о художественной литературе) хочет получить от текста только то, что считает необходимым. Он любит пропускать большие и не слишком интересные для него разделы, забегать вперед, читать текст в удобном лично для него порядке, использовать его как справочник и т.д. Поэтому любой текст, размеры которого превышают возможности одномоментной обозримости, должен обязательно сопровождаться рубрикатором. В зависимости от выбранной системы рубрикации это может быть содержание или карта сайта.

В научно-технических текстах приняты следующие средства рубрикации:

- абзац,
- нумерация,
- система заголовков.

Абзац (отступ вправо в начале первой строки каждой части текста) является простейшей рубрикой. Абзац указывает на переход от одной мысли или темы к другой.

Нумерация – числовое или буквенное обозначение последовательности расположения частей текста. Возможны два типа нумерации:

Часть первая.	1.
Раздел А.	1.1.
Глава I.	1.1.1.
§ 1.	1.1.1.1.
...	...

Система заголовков – наиболее информативный способ рубрикации. Заглавие – это структурный элемент текста. Оно позволяет в предельно краткой форме отразить тематику научного произведения, нередко и его основную мысль. Информативный заголовок – это предельно краткий реферат содержания раздела, главы, параграфа.

В русских технических текстах используются следующие типы заголовков:

- назывное предложение («Файлы с расширением .doc»);
- предложное сочетание («О файлах с расширением .doc»);
- вопрос («Что такое файлы с расширением .doc», «Как записать файлы с расширением .doc»);

Назывные заголовки наиболее употребительны. Главное слово должно обозначать основную тему раздела (например, «Файл настройки» или «Редактирование файла настройки»). Употребление таких заголовков практически не имеет ограничений, но иногда порождает слишком длинные конструкции (например,

«Разработка системы автоматизированного тестирования для software development kit, предназначенного для работы с графической информацией»).

Предложное сочетание употребляется, как правило, в качестве заголовка обзорного или вводного раздела, то есть раздела, не охватывающего всю тему, а лишь сообщающего некоторые сведения о ней.

Заголовок-вопрос можно употреблять либо для описания конкретной процедуры («Как отформатировать дискету»), либо если этот вопрос является часто задаваемым («Что делать при отсутствии соединения»). Такой заголовок соответствует «популярному» стилю изложения и плохо подходит для серьезных или официальных текстов.

Ниже перечислены основные требования к рубрикации текстов при иерархической системе.

- Заголовок должен описывать все то и только то, что содержится в данном разделе.
- Заголовки одного уровня должны быть однотипными.
- Структура разделов должна полностью соответствовать требованиям, предъявляемым к сбалансированному иерархическому дереву.
- Каждый раздел, кроме самых нижних, должен иметь не менее двух подразделов.

Заголовки каждого уровня должны быть оформлены единообразно, а заголовки разных уровней должны различаться. Для этого можно использовать средства, которые перечислены ниже в порядке убывания предпочтительности:

- шрифтовые различия (кегль, начертание и гарнитура шрифта):
- подчеркивающие линейки разного начертания;
- наборные или рисованные изображения, элементы убранства;
- цветовые различия;
- трансформация шрифта.

Количество уровней рубрикации текста определяется несколькими факторами.

- Чем больше и сложнее текст, тем более уровней рубрикации требуется.
- В текстах, которые служат для изучения и справок, всегда больше оснований для более дробного членения. В особенности это важно, если текст будет использоваться в гипертекстовом режиме.
- Объем отдельной рубрикационной единицы не должен превышать возможностей единовременного восприятия текста.
- Читателю трудно удержать в памяти «иерархическую лестницу» заголовков глубиной более 3–4.
- Чем слабее подготовка читателей, тем число ступеней должно быть меньше.

Помимо рубрикации, в тексте могут быть по желанию составителя акцентированы сведения определенного типа:

- выделенные сообщения (определения, инструкции, советы, замечания и т.п.)
- выделенные элементы (вновь вводимые термины, названия фирм, горячие клавиши и т.д.).

Выделенные элементы могут входить в состав выделенных сообщений.

Ниже перечислены основные требования к оформлению выделяемых элементов.

- Параметр выделения должен быть формальным и понятным читателю (например, вновь определяемые термины). Недопустимо выделение оформлением (например, курсивом) отдельных фрагментов текста по принципу их важности для автора.
- Выделение должно быть единообразным по всему тексту (например, выделяются все вновь вводимые термины).
- Выделения по разным параметрам должны быть оформлены по-разному.

Некоторые порции информации в тексте могут быть оформлены в виде перечислений – списков.

Ниже перечислены основные требования к составлению и оформлению списков.

- Список является выделенным элементом. Поэтому каждый список должен предваряться фразой, вводящей его в основной линейный текст (например, «ниже приводится последовательность действий...»).
- Недопустимо вводить список в основной текст с помощью назывного предложения и, тем более, строить текст как набор списков.
- Нумерация позиций в списке должна быть логически обоснована. Если все позиции равнозначны, нужно использовать маркированный список.
- Все позиции списка должны быть организованы однотипно.

Задание лабораторной работы №2:

1. ***Выбрать в сети Интернет текст по тематике «ИТ» с отсутствующим рубрикатором, согласовать выбор с преподавателем.***
2. ***Составить рубрикатор текста глубиной не менее 2.***
3. ***Отредактировать текст в соответствии с составленным рубрикатором***

   Объем отредактированного текста – 8000–9000 знаков.

Представляемые к защите материалы

1. ***Исходный текст (в распечатанном виде).***
2. ***Отредактированный текст (в распечатанном виде).***

Лабораторная работа 3. Преобразование стиля текста

Цель работы: научиться поддерживать стиль текста в соответствии с ожиданиями целевой группы.

Стиль изложения – одно из наиболее сложно формализуемых качеств текста. «Абсолютно хорошего» стиля не существует: основной критерий – это соответствие стиля ожиданиям целевой группы. В то же время любой представитель этой

целевой группы легко отличит «плохо написанный» текст от «хорошо написанного», и, соответственно, передаваемая ему информация получит негативную или позитивную эмоциональную поддержку.

Таким образом, в условиях рынка составитель текста решает следующие задачи:

- определить адресную целевую группу;
- угадать ее стилистические ожидания;
- составить (написать, преобразовать) текст в соответствии с этими ожиданиями.

Появление Интернета радикально изменило подходы к решению этих задач:

- диапазон социально приемлемых стилей существенно расширился (см. врезку 1);
- один и тот же человек может принадлежать к нескольким целевым группам (например, являться менеджером ИТ-компании и одновременно блоггером);
- несоответствие стилю конкретной целевой группы вызывает резкое неприятие (см. врезки 2–4);
- даже хорошо написанный текст, функционально полностью соответствующий своим задачам, может просто не понравиться представителю заказчика, вызвать у него неприятие на уровне личного вкуса.

Некоторые цитаты из Интернет-форумов, иллюстрирующие ситуацию, приведены во врезках 1–4. В цитатах сохранены стиль и орфография авторов, однако удалены нецензурные выражения.

Врезка 1.

http://dbd.ru/~alex/20445

В этой заметке я буду перечислять то, что я ненавижу всем сердцем. Искренне и отчаянно. Естественно, заметка будет обновляться.

...

*6. Терпеть не могу, когда пишут с ошибками! Особенно, когда путают «тся» и «ться» (это полный ***). Ведь то, как пишет человек, наглядно иллюстрирует то, что творится у него в голове. С неграмотными людьми иметь дело опасаюсь.*

Сей гнев не касается заведомо грамотных людей, которые дурачяцца.

...

*8. Жывой журнал. Бесит неизъяснимо. Просто ***. (И еще много всего по поводу жж – censored. Не скажу. А то среди нас жыжыстов тут просто *** ешь, так разобидятся еще. Ведь они такие мнительные и ранимые.)*

Комментарий:

Пункт 8 и "которые дурачяцца" – опять же, имхо, уже пережило себя. Уже не стильно, а "попсово".

Пункт 6 – поддерживаю, это говорит о человеке не в его пользу, но ненависти не вызывает.

Врезка 2.

forums.overclockers.ru

Имею сказать следущее:

стиль изложения как то кисловат, особенно не понравилось

“главный Вывод: к разгону системы (именно системы, а не только процессора) необходимо подходить как к решению достаточно серьёзной технической задачи.”

напоминает предисловие к учебнику, где автор считает себя на порядок “мудрее”

Врезка 3.

Конференция iXBT.com: «Компьютерный гипермаркет САНРАЙЗ-ПРО »

... жуткая смесь русского и английского в прайс-листе не всегда удобна. Очень многие товары можно описать на нормальном русском языке, не используя английские термины.

А вообще, лично меня, например, сильно раздражает использование слова «опционально». Ну нет такого слова в русском языке, и по моему опыту, достаточно многие люди не понимают, что оно означает. А ведь можно очень просто написать: «дополнительные элементы, приобретаемые отдельно» или «в поставку не включено» или как-то еще. Хотя некоторые компьютерные специалисты так погрузились в жаргон, что уже стали забывать русский язык.

Можно приводить и другие примеры, но я лучше посоветую, как, на мой взгляд, следует решать проблему.

Необходимо нанять редактора, который проверит и исправит все описания, причем возможно, потребуется два редактора – по техническим вопросам и по вопросам русского языка. Те, кто составляют описания, не могут находить ошибки, это совершенно другая работа. И посетители форумов не могут, так как смотрят только интересующие их описания. Проверкой и исправлением нужно заняться профессионально. Кроме того, редактор может не только исправить ошибки, но и выработать единый стиль изложения, чтобы описания разных товаров различных производителей было легче читать и сравнивать.

Если же оставить все как есть, через некоторое время прайс будет немного получше, чем сейчас, но все равно в нем регулярно будут находиться ошибки и постоянно будут появляться возмущенные покупатели. А главное, из-за отсутствия информации значительная часть покупателей откажется от покупки в Санрайзе (они просто не узнают, что там есть именно то, что им нужно).

А стиль изложения – это вообще отдельная песня. Люблю, когда всё красиво, а стилистические и, в особенности, грамматические ошибки, очень бросаются в глаза и не делают чести такой серьёзной фирме.

Врезка 4.

http://www.kinoafisha.msk.ru/forum

crazy:

Я вот не понимаю тех, кто всё время придирается к человеку, не обидев его какой ни будь грубостью, а именно придирается к ошибкам! И таких людей, к со-

жалению, очень много на форуме, очень жаль. Мне вот в принципе всё равно, мне плевать на грамматические ошибки пользователей (коих встречается не мало, в том числе и я, может, и вы уважаемый читатель), но я считаю просто тупостью придираться. Да и вообще, какое право человек имеет делать ему замечание по поводу его образованности?!

К сожалению, есть такой тип людей, может, им мешает что-то жить, и они готовы оторваться на таких, как я, но сколько можно. Тут форум, а не общество людей, состоящих из «батанов», или общество, которое именуется «А давайте научимся писать правильно?!».

Бывают, конечно, допущены и ошибки и пр., я это прекрасно вижу, но вот кто-то, которого я вообще не встречала на форуме никогда, человек, который ни разу не написал, ни один пост, заявляет такое!!! Какое вы имеете право?!!! Я сама знаю, какая я, и не надо меня учить!

darthkim:

Когда людям ответить нечего, то они отходят от темы и пытаются всяческими извращенными способами показать свое превосходство. Ни и пускай лесом себе идут такие люди. И нечего заморачиваться на идиотов, нервы целее будут.

Нина:

Действительно, человеку дана врожденная грамотность. И как он пишет – это его личное дело. Мы уже все не в первом классе, чтобы тыкать нас носом. Тем более – это свободный форум, каждый пишет о кино то, что он думает и как он хочет (оговорюсь, что в рамках правил форума). Обсуждаем кино, а не посты и стиль друг друга! Свободу попугаям!

Кстати, сейчас вообще модно так писать, до неузнаваемости искажая и сокращая слова. Американский сленг – там вообще слова составляют из первых слогов словосочетаний и понять их практически невозможно! Так что у нас очень культурный и образованный форум!

zayka:

Надо слать всех "учителей русского" подальше! Crazy, не расстраивайся, дружок мой!

Ночь:

Главное не то, как человек пишет (правильно или нет), главное – что он пишет !!!

Sanchez:

Цитата (darthkim, 04/05/2005, 22:30:13):

Когда людям ответить нечего, то они отходят от темы и пытаются всяческими извращенными способами показать свое превосходство. Ни и пускай лесом себе идут такие люди. И нечего заморачиваться на идиотов, нервы целее будут.

Дартким, перестань писать правильно! Я, пока не увидел твое имя, думал, что это другой человек писал – ни одной ошибки и мысль хорошо мне понятна!

Это здорово, конечно, но твой уникальный стиль изложения – это все же ТВОЙ стиль!

ВЕРНИ ЕГО НА МЕСТО!

Принято считать, что «серьезная», «научная» информация должна обязательно оформляться в «серьезном», академическом стиле. Однако это не так (см. врезку 5). Более того, подача одной и той же информации одновременно в различных стилях может служить удачным маркетинговым приемом.

Врезка 5.

http://lib.metromir.ru/book27574

«Автор считает неконструктивным деление научных работ на академические (трудно читаемые) и популярные (легковесные). Любую сложную проблему можно изложить живым и ясным языком, не снижая научной значимости».

Л.Н. Гумилёв

Для формализации отношений заказчиков и составителей технических текстов и предотвращения споров о качестве текста документации разработан стандарт «Стиль изложения документации пользователя программного средства. Общие требования» [14].

Пример преобразования технического текста из стиля «Интернет-форум» в «академический» стиль приведен в табл. 3.1.

Таблица 3.1

стиль «Интернет-форум»	**«академический» стиль**
Что в имени тебе моем?	*Установка и настройка DNS-сервера под Windows Server*
Во времена, когда Землю топтали динозавры, в Сети не было имен хостов и все пользовались 4-циферными IP-адресами. Однако когда число этих адресов перевалило за десять, динозавры стали задумываться о потенциальных объемах своей памяти. Прошло немного времени, и в 1982 году придумали такую замечательную штуку, как DNS...	Раньше в глобальной сети Интернет не было имен хостов, на начальном этапе существовали IP-адреса из четырех цифр. Однако со временем требования к памяти компьютеров росли, и таких ограничений стало не хватать – поэтому в 1982 году появилась система DNS.
Да-да, мы поговорим именно об использовании DNS, точнее, о настройке этого чуда под Windows Server. Еще точнее — не просто о настройке, а о настройке DNS-сервера, который позволит твоему «серверному лесу» и пользователям работать с полным комфортом.	В данной статье речь пойдет об использовании DNS, точнее, о настройке этой системы под Windows Server.

Для ясности уточним, что такое зона, домен и хост.	Уточним, что такое зона, домен и хост.
Домен – это некий контейнер, содержащий в себе зоны, хосты или другие домены. Домен, таким образом, представляет собой исключительно виртуальную структуру, не привязанную к компьютеру.	Домен – это некий объект-контейнер, содержащий в себе зоны, хосты или другие домены. Домен, таким образом, представляет собой исключительно виртуальную структуру, не привязанную к компьютеру.
Зона – это контейнер, содержащий в себе несколько доменов и хостов. Этот контейнер определяет общие разрешения управления своим содержимым. Различие между зонами и доменами состоит в том, что домен может поддерживать несколько зон, делегируя таким образом различные полномочия на поддомены и их хосты.	Зона – это контейнер, содержащий в себе несколько доменов и хостов. Этот контейнер определяет общие разрешения управления своим содержимым. Различие между зонами и доменами состоит в том, что домен может поддерживать несколько зон, делегируя таким образом различные полномочия на поддомены и их хосты.
Хост – это, по сути, один-единственный компьютер, содержащийся в домене. Стоит отметить, что имя хоста может совпадать с именем домена, как бы нивелируя виртуальность последнего.	Хост – это, по сути, один-единственный компьютер, содержащийся в домене. Стоит отметить, что имя хоста может совпадать с именем домена, как бы нивелируя виртуальность последнего.
Производитель ОС Windows рекомендует администраторам интегрировать DNS в Active Directory, что стало возможным начиная с ОС Windows 2000 Server. Такая интеграция существенно упрощает процесс администрирования сети.	Производитель ОС Windows рекомендует администраторам интегрировать систему DNS в службу каталогов Active Directory, что стало возможным, начиная с ОС Windows 2000 Server. Такая интеграция существенно упрощает процесс администрирования сети.
Произвести интеграцию очень легко. При создании Active Directory ставим галочку «Install and configure the DNS server on this computer and set this computer to use this DNS server as the preferred DNS server».	Для интеграции необходимо при создании службы каталогов Active Directory включить параметр «Install and configure the DNS server on this computer and set this computer to use this DNS server as the preferred DNS server».

Для имени домена лучше выбирать два слова, разделенных точкой. Например «skynet.ru», как проделал я в моем случае. В принципе возможно сделать домен с простым разделением точкой, но Microsoft не рекомендует это, иначе могут возникнуть проблемы в работе домена.	Для имени домена рекомендуется выбирать два слова, разделенных точкой, например «skynet.ru».
После запуска сервера и входа под учетной записью администратора ты увидишь привычную консоль Manage Your Server. Прокручиваем ее и находим ссылку для настройки DNS. Далее достаточно нажать кнопочку Manage this DNS server и получить результат — консоль управления DNS-сервером.	После запуска сервера и входа под учетной записью администратора загружается консоль «Manage Your Server», на которой имеется ссылка для настройки DNS, ведущая в консоль «Manage Your DNS-Server».
Начнем с самого важного — с Event Viewer. Эта часть консоли управления существенно поможет в борьбе с возможными проблемами. У меня, как видишь, проблем нет, и эти записи чисто информационного характера ;). Но в большой сети ты в любом случае не обойдешься без ошибок, так что используй эту возможность на все 100%!	Самый важный раздел – Event Viewer. Эта часть консоли управления предназначена для помощи при решении проблем. В большинстве случаев без ошибок не обходится, а они отмечаются именно здесь.
Я пишу это не к тому, что можно создать «кривую» зону :), а к тому, что она занимается прямым разрешением имен, то есть преобразует доменные имена в IP-адреса. Также существует «кривая» зона, которую принято называть обратной, она занимается разрешением IP-адресов в имена доменов. Если относиться ко всему этому строго, то для нормальной работы компьютера в сети для него должны существовать записи и в прямой зоне, и в обратной, причем желательно, чтобы они ссылались друг на друга.	Для нормальной работы компьютера в сети для него должны существовать записи и в прямой зоне, и в обратной, причем желательно, чтобы они ссылались друг на друга.

Задание лабораторной работы №3:

1. ***Выбрать в сети Интернет текст по тематике ИТ с ярко выраженной стилистикой специальной целевой группы (см. ЛР 1), согласовать выбор с преподавателем.***
2. ***Выбрать другую адресную целевую группу.***
3. ***Преобразовать текст для выбранной группы.***
   Объем отредактированного текста – 3000–4000 знаков.

Представляемые к защите материалы

1. ***Исходный текст (в распечатанном виде).***
2. ***Отредактированный текст (в распечатанном виде).***

Лабораторная работа 4. Перевод технической документации

Цель работы: научиться адекватно использовать материалы на иностранном языке для составления контента.

Перевод – это не механическое воспроизведение всей совокупности элементов подлинника, а сложный отбор различных возможностей их передачи. Существует несколько стратегий перевода. Для создания контента лучше всего подходит так называемая «стратегия приблизительного перевода», основная задача которой – передача смысла исходного текста, а не его художественных или стилевых особенностей.

Раньше было принято считать, что письменный перевод – процесс исключительно творческий, сродни написанию художественной книги. Однако сегодня при переводе, в первую очередь, важны точность передачи информации и оперативность исполнения. Как отмечает сотрудник бюро переводов, «...мы не ищем гениев – с нами очень тяжело работать. Гений начинает отстаивать свою правоту при переводе каких-то общеупотребимых терминов, но для нас, прежде всего, важны требования заказчика, которые мы проецируем на наших исполнителей».

Сегодня знание языка, профессиональный перевод и составление контента – это разные вещи. Наличие диплома переводчика давно не является обязательным условием при приеме на работу, связанную с переводом технических текстов. Работодатели часто делают выбор в пользу технических специалистов с хорошим знанием иностранного языка, которые лучше лингвистов владеют специальной терминологией и понимают смысл текста.

Таким образом, при составлении контента на базе технических англоязычных текстов требуется:

- точность передачи смысла исходного текста;
- унификация терминологии и стиля;
- соответствие терминологии и стиля ожиданиям потенциальных потребителей;
- максимальная оперативность перевода.

Для реализации этих требований целесообразно использовать такие ИТ технологии, как:

- системы машинного перевода;
- компьютерные словари;
- технологии Translation Memory.

Системы машинного перевода эффективны для составления первичного варианта перевода, который, как показывает практика, править намного быстрее, чем переводить заново «с нуля». Согласно некоторым оценкам, подстановка даже на 80% совпадающих сегментов из базы переводов может сократить время работы над переводом на 50–60%.

Принцип работы технологии Translation Memory состоит в следующем. В процессе перевода пары «исходный текст – конечный (переведенный) текст» накапливаются в базе (или базах) данных и затем используются для перевода новых документов. Единицей (сегментом) сравнения является, как правило, предложение, но могут быть и другие правила сегментации.

Программно реализованная система Translation Memory автоматически сравнивает все сегменты текста с уже имеющимися в базе. Если системе удается найти полностью или частично совпадающий сегмент, то его перевод отображается с указанием совпадения в процентах. Как правило, задается порог совпадений на уровне не ниже 75%. При меньшем проценте совпадения этот сегмент быстрее перевести вручную. Слова и фразы, которые отличаются от сохраненного текста, подсвечиваются.

При составлении контента следует максимально широко использовать эту технологию, хотя бы в ручном режиме или с помощью макросов.

Ниже приводятся некоторые рекомендации по переводу англоязычных текстов для составления контента.

- Не нужен дословный перевод, нужен гладкий и понятный русский текст. Перевод должен быть русским пересказом на заданную тему с сохранением технических подробностей. Для этого итоговый перевод может содержать гораздо больше слов и предложений, чем оригинал.
- В процессе работы можно использовать любую систему машинного перевода, но машинный перевод не должен появляться в итоговом тексте.
- Чтобы правильно передать смысл оригинала в переводе, нужно хорошо понять принципы работы того устройства или программы, о которых идет речь. По возможности нужно поставить у себя ту программу, о которой идет речь в оригинале, и сверять свой перевод с реальной практикой на своем компьютере.
- Для некоторых программ существуют официальные русские версии, в этом случае весь перевод должен идти с использованием перевода терминов по официальной, а не пиратской русской версии.
- Следует по возможности избегать англицизмов и жаргонной терминологии (аддоны, юзать и др.).
- Многие общепринятые названия (Microsoft SQL Server, Visual Basic for Application) не переводятся, а дословно переписываются из источника.

- Если вы назвали что-то определенным термином, то далее в тексте должны использовать именно его. Например, использовав слова «логотип» и «логин», нельзя далее использовать слова «эмблема» и «учетная запись».
- При написании имен, фамилий, названий книг рядом с русским переводом необходимо писать в скобках английский вариант, приведенный в источнике.
- Как показывает практика, книга по схожей тематике является хорошим подспорьем для перевода не только терминов, но и практических алгоритмов. Целесообразно выбрать хорошую переводную книгу, совпадающую по стилю с переводимым текстом (см. ЛР 3), и следовать ее стилю.
- Объявите, защитите и поддерживайте во всем тексте свой стиль перевода (от практически пословного до передачи смысла крупных кусков текста).

Последовательность работы над переводом технического текста проиллюстрирована в табл. 4.1.

Таблица 4.1

Исходный текст	Машинный перевод	Отредактированный перевод
Preface Once upon a time, a consultant made a visit to a development project. The consultant looked at some of the code that had been written; there was a class hierarchy at the center of the system. As he wandered through the hierarchy, the consultant saw that it was rather messy. The higherlevel classes made certain assumptions about how the classes would work, assumptions that were embodied in inherited code. That code didn't suit all the subclasses, however, and was overridden quite heavily. If the superclass had been modified a little, then much less overriding would have been necessary. In other places some of the intention of the superclass had not been properly understood, and behavior present in the superclass was duplicated. In yet other places several subclasses did the same thing with code that could clearly be moved up the hierarchy. The consultant recommended to the project management that the code be looked at and cleaned up, but the project management	**Предисловие** Когда-то, консультант сделал посещение проекта развития. Консультант смотрел на часть кодекса, который был написан; была иерархия класса в центре системы. Поскольку он блуждал через иерархию, консультант видел, что это было довольно грязно. higherlevel классы сделали определенные предположения о том, как классы будут работать, предположения, которые были воплощены в унаследованном кодексе. Тот кодекс не удовлетворял всем подклассам, однако, и был отвергнут весьма тяжело. Если бы суперкласс был изменен немного, то намного меньше отвергания было бы необходимо. В других местах не была должным образом понята часть намерения суперкласса, и подарок поведения в суперклассе был дублирован. Во все же других местах несколько подклассов сделали ту же самую вещь с кодексом, который мог ясно быть продвинут иерархия. Консультант рекомендовал к руководству проектом, что на кодекс смотрели и вымыться,	**Предисловие** Однажды консультант анализировал проект в стадии разработки. Посмотрев на код с иерархией классов, он посчитал ее довольно беспорядочной. Классы высшего уровня имели серьезные допущения на предмет своего функционирования, которые были реализованы при наследовании. Этот код подходил не всем подклассам, и замена виртуальных методов проходила достаточно тяжело. Если бы основной класс был написан чуть иначе, то изменений в наследуемых классах было бы существенно меньше. В одних случаях цель основного класса была неправильно понята и его действия были продублированы, а в других случаях подклассы реализовывали операции более высокого уровня. Консультант порекомендовал руководителям пересмотреть код и привести его в порядок, но эта идея не вызвала энтузиазма – код казался

didn't seem enthusiastic. The code seemed to work and there were considerable schedule pressures. The managers said they would get around to it at some later point.
The consultant had also shown the programmers who had worked on the hierarchy what was going on. The programmers were keen and saw the problem. They knew that it wasn't really their fault; sometimes a new pair of eyes are needed to spot the problem. So the programmers spent a day or two cleaning up the hierarchy. When they were finished, the programmers had removed half the code in the hierarchy without reducing its functionality. They were pleased with the result and found that it became quicker and easier both to add new classes to the hierarchy and to use the classes in the rest of the system.
The project management was not pleased. Schedules were tight and there was a lot of work to do. These two programmers had spent two days doing work that had done nothing to add the many features the system had to deliver in a few months time. The old code had worked just fine.
So the design was a bit more "pure" a bit more "clean." The project had to ship code that worked, not code that would please an academic. The consultant suggested that this cleaning up be done on other central parts of the system. Such an activity might halt the project for a week or two. All this activity was devoted to making the code look better, not to making it do anything that it didn't already do.
How do you feel about this story? Do you think the consultant was right to suggest further clean up? Or do you follow that old engineering adage, "if it works, don't fix it"?

но руководство проектом не казалось восторженным. Кодекс, казалось, работал и были значительные давления списка. Менеджеры сказали, что они найдут время для этого в некотором более позднем пункте.
Консультант также показал программистам, которые воздействовали на иерархию, что продолжалось.
Программисты сильно желали и видели проблему. Они знали, что это не была действительно их ошибка; иногда новая пара глаз необходима, чтобы определить проблему. Таким образом программисты провели а день или два мытья иерархии. Когда они были закончены, программисты удалили половину кодекса в иерархии, не уменьшая ее функциональные возможности. Они были рады результату и найдены, что это стало более быстрым и более легким и добавить новые классы к иерархии и использовать классы в остальной части системы.
Руководство проектом не было радо. Списки были напряженны и было много работы, чтобы сделать. Эти два программиста провели два дня, делая работу, которая не сделала ничего, чтобы добавить много особенностей, которые система должна была поставить через несколько месяцев. Старый кодекс работал только прекрасный.
Таким образом проект был "немного более чист" "немного более чистый". Проект должен был отправить кодекс, который работал, не кодекс, который будет понравиться академику. Консультант предложил, что это мытье было сделано на других центральных частях системы. Такая деятельность могла бы остановить проект в течение недели или два.
Вся эта деятельность была

работоспособным, к тому же были временные ограничения. Руководство заявило, что вернется к этому позже.
Консультант также объяснил происходящее программистам, работавшим над иерархией классов, и они поняли проблему. Они знали, что это была не их вина, так как иногда проблема видна только свежим взглядом.
Программисты за день или два привели иерархию классов в порядок. По окончании работ была удалена половина программного кода без ущерба для функциональности. Они были довольны результатами и обнаружили, что стало легче как добавлять новые классы, так и использовать существующие
Руководство, однако, не было довольно – сроки были очень сжатыми, а впереди было еще много работы. Эти два программиста провели два дня за работой, которая никак не влияла на ту функциональность, которую следовало реализовать уже через несколько месяцев. Старый код и так работал отлично. Разработка стала немного строже и яснее – и только. Однако результатом проекта должен быть работающий (а не академически правильный) код. Консультант же предлагал подобную оптимизацию кода в других частях системы, что могло приостановить работу над проектом на неделю или две – только для того, чтобы код просто выглядел лучше, но при этом не делал ничего нового.
Что Вы думаете об этой истории? Был ли консультант прав, предлагая дальнейшую оптимизацию?

I must admit to some bias here. I was that consultant. Six months later the project failed, in large part because the code was too complex to debug or to tune to acceptable performance.
The consultant Kent Beck was brought in to restart the project, an exercise that involved rewriting almost the whole system from scratch. He did several things differently, but one of the most important was to insist on continuous cleaning up of the code using refactoring. The success of this project, and role refactoring played in this success, is what inspired me to write this book, so
that I could pass on the knowledge that Kent and others have learned in using refactoring to improve the quality of software.

What Is Refactoring?

Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure. It is a disciplined way to clean up code that minimizes the chances of introducing bugs. In essence when you refactor you are improving the design of the code after it has been written.
"Improving the design after it has been written." That's an odd turn of phrase. In our current understanding of software development we believe that we design and then we code.
A good design comes first, and the coding comes second. Over time the code will be modified, and the integrity of the system, its structure according to that design, gradually fades. The code slowly sinks from engineering to hacking.
Refactoring is the opposite of this practice. With refactoring you can take a bad design, chaos

посвящена созданию кодового лучше взгляда, не тому, чтобы заставлять это сделать что - нибудь, что это уже не делало.
Как Вы чувствуете
об этой истории? Вы думаете консультант, право состояло в том, чтобы предложить, далее моются? Или Вы следуете за той старой технической пословицей, ", если это работает, не устанавливайте это"?
Я должен признаться
в некотором уклоне здесь.
Я был тем консультантом.
Шесть месяцев спустя проект терпел неудачу, в значительной степени потому что кодекс был слишком сложен, чтобы отладить или настроиться к приемлемой работе.
Консультант Кент Бек был введен, чтобы повторно начать проект, осуществление, которое вовлекало переписывание почти целая система на пустом месте. Он сделал несколько вещей по-другому, но один из самых важных должен был настоять на непрерывном мытье перефакторинга использования кодекса.
Успех этого проекта, и роли, повторно выступающей в качестве фактора играемый в этом успехе, является тем, что вдохновило меня писать эту книгу, таким образом
то, что я мог передать знание, что Кент и другие узнали в использовании перефакторинга, чтобы улучшить качество программного обеспечения.

Что Повторно выступает в качестве фактора?

Перефакторинг - процесс изменения системы программного обеспечения таким способом, что это не изменяется, внешнее поведение кодекса все же улучшает его внутреннюю структуру. Это - дисциплинированный способ вымыть кодекс, который минимизирует возможности

Или Вы придерживаетесь мнения, что не следует чинить то, что работает?
Я должен кое в чем признаться: я был этим консультантом. Шесть месяцев спустя проект провалился, во многом из-за того, что код был слишком сложным для отладки или настройки под приемлемую производительность.
Консультант Кент Бек был привлечен для перезапуска проекта, который включал написание всей системы с нуля. Некоторые вещи он сделал по-другому, но наиболее важным было то, что он настоял на оптимизации кода с использованием рефакторинга. Успех этого проекта (и роль рефакторинга в этом успехе) – вот что побудило меня написать эту книгу, чтобы я мог передать знания, которые Кент и другие получили при использовании рефакторинга как средства повышения качества программного обеспечения.

Что такое рефакторинг?

Рефакторинг – это процесс изменения программного кода таким образом, что его внешнее поведение не изменяется, но улучшается его внутренняя структура. Это строгий способ приведения кода в порядок, который уменьшает вероятность ошибок. В сущности, если вы занимаетесь рефакторингом, то это означает, что вы улучшаете структуру программного кода уже после того, как он написан.
«Улучшаете структуру программного кода уже после того, как он написан» - довольно странная фраза.
В нашем текущем понимании, мы сначала проектируем

even, and rework it into well-designed code. Each step is simple, even simplistic. You move a field from one class to another, pull some code out of a method to make into its own method, and push some code up or down a hierarchy. Yet the cumulative effect of these small changes can radically improve the design. It is the exact reverse of the normal notion of software decay. With refactoring you find the balance of work changes. You find that design, rather than occurring all up front, occurs continuously during development. You learn from building the system how to improve the design. The resulting interaction leads to a program with a design that stays good as development continues.	на представление ошибок. В основном, когда Вы повторно выступаете в качестве фактора, Вы улучшаете проект кодекса после того, как это было написано. “Улучшая проект после того, как это было написано.” Это - нечетный поворот фразы. В нашем текущем понимании разработки программного обеспечения мы полагаем, что мы проектируем, и затем мы кодируем. Хороший проект прибывает сначала, и кодирование прибывает секунда. В течение долгого времени кодекс будет изменен, и целостность системы, ее структура согласно тому проекту, постепенно исчезает. Кодекс медленно снижается от разработки до рубки. Перефакторинг - противоположность этой практики. С перефакторингом Вас может взять плохой проект, хаос даже, и переделать это в хорошо - разработанный кодекс. Каждый шаг прост, даже упрощен. Вы перемещаете область от одного класса до другого, вытаскиваете некоторый кодекс из метода, чтобы превратить в его собственный метод, и увеличить некоторый кодекс или вниз иерархия. Все же совокупный эффект этих маленьких изменений может радикально улучшить проект. Это - точная перемена нормального понятия распада программного обеспечения. С перефакторингом Вас находят баланс изменений работы. Вы находите, что проект, вместо того, чтобы произойти все фронт, происходит непрерывно в течение развития. Ю учится из построения системы, как улучшить проект. Получающееся взаимодействие приводит к программе с проектом, который остается хорошим, поскольку развитие продолжается.	структуру, а потом ее реализуем в программном коде. Сначала идет проектирование, а потом программирование. Со временем код будет меняться, и целостность системы, ее ранее разработанная структура постепенно исчезают. Программист становится не инженером, а хакером. Рефакторинг – противовес такой практике. С рефакторингом Вы можете взять плохую структуру проекта и переработать ее в хорошо структурированный код. Каждый шаг очень прост. Вы перемещаете поля одного класса в другой, делаете часть одного метода другим методом, перемещаете программный код по иерархии классов и т.д., пока количество этих небольших изменений не приведет к качественному улучшению структуры проекта. Это перемена стандартного взгляда на размытие структуры программного обеспечения при разработке. С рефакторингом Вы находите противовес вынужденным переменам и понимаете, что проектирование осуществляется не только в начале проекта, но и на всем его протяжении. Построение системы может помочь Вам улучшить ее структуру. Как результат, Вы получаете программу с хорошей структурой даже во время разработки.

Задание лабораторной работы №4:

1. ***Выбрать в сети Интернет англоязычный текст по тематике ИТ, не имеющий русскоязычного зеркала. Согласовать выбор с преподавателем.***
2. ***Выбрать адресную целевую группу.***
3. ***Перевести текст для выбранной группы.***
   Объем отредактированного текста – 3000–4000 знаков.

Представляемые к защите материалы

1. ***Исходный текст (в распечатанном виде).***
2. ***Отредактированный текст (в распечатанном виде).***

Лабораторная работа 5. Корректура контента

Цель работы: ознакомиться с основными правилами и приемами корректуры, научиться держать корректуру технического текста по специальности.

Корректура – это исправление оттисков с набора. Цель корректуры – устранить в подготовленном документе ошибки всех видов (орфографические, стилистические, смысловые, технические и т.д.).

В профессиональном издательстве корректуру держат автор, редактор, корректор, художественный и технический редакторы. При однократной подготовке технического документа эти функции, как правило, совмещает один человек.

Требования стандарта к качеству набора и верстки выражаются следующими цифрами: не более 15 ошибок всех видов на 1 учетно-издательский лист (40 000 знаков, то есть 16 страниц) в гранках, не более одной – после выходной корректуры. К сожалению, в настоящее время эти требования часто не выдерживают даже специализированные издательства, имеющие профессиональный штат редакторов и корректоров.

Тем не менее, как показывает опыт защиты выпускных квалификационных работ (ВКР), наличие корректурных ошибок в тексте пояснительной записки и, тем более, в презентации к ВКР снижает результирующую оценку не менее чем на 0,5 балла.

От специалиста в области ИТ не ожидается полномасштабное владение методикой корректуры, которую использует профессиональный корректор. Ниже приводятся некоторые приемы, позволяющие представить подготовленный текстовый материал на уровне, адекватном требованиям к защите ВКР.

В противовес широко распространенному мнению, научиться грамотно писать по-русски (практически независимо от исходного уровня) можно очень быстро – за 8 дней. Такие курсы существуют, в частности, в Санкт-Петербурге.

Если поднять свою грамотность до приемлемого уровня не удается, то для ответственных текстов необходимо привлекать корректора.

От автора текста, как минимум, ожидается выполнение следующих требований:

- с удвоенным вниманием проверять заголовки – в них вероятность не заметить ошибку увеличивается;
- не ориентироваться на программы автоматической проверки правописания (например, Word) – они часто дают ошибки;
- пользоваться эталоном – взять книгу, подготовленную солидным издательством, лучше до 1990 г. выпуска, и следовать принятым в ней правилам верстки;
- проблемы правописания решать с помощью справочников (например, справочно-информационного портала «Грамота.ру», http://spravka.gramota.ru/pravila.html)

Задание лабораторной работы №5:

1. ***Составить пояснительную записку по теме курсового проекта, выполненного в 9 семестре, или предполагаемой теме выпускной квалификационной работы. Текст должен иметь объем 15 000–20 000 знаков (6–8 страниц), содержать иллюстрации и/или другие нетекстовые элементы (формулы, фрагменты кода, таблицы и т.п.). Согласовать составленный документ с преподавателем.***
2. ***Заверстать подготовленный текст в соответствии с требованиями ЛР 13 (тем самым выполнив эту ЛР).***
3. ***Выполнить корректуру сверстанного текста. Требования к готовому тексту – не более 2 корректурных ошибок на 2000 знаков текста.***

Представляемые к защите материалы

1. ***Готовый текст (в распечатанном виде).***

РАЗДЕЛ 2. ПОДГОТОВКА ВЕКТОРНОЙ ГРАФИКИ

Основная идея векторной графики – представить графическое изображение как совокупность замкнутых и разомкнутых контуров с заданными параметрами заливки и обводки. Каждый контур состоит их одного или нескольких сегментов. Начало и конец каждого сегмента называют точками привязки или опорными точками. Координаты этих точек определяют положения каждого сегмента (привязывают его к определенной позиции в документе).

Существует три типа контуров:

- открытые – линии, имеющие две отдельные концевые точки, между которыми может быть расположено произвольное количество опорных точек;
- замкнутые – непрерывные линии, не имеющие концевых точек, без начала и без конца;
- составные – линии, состоящие из нескольких замкнутых или открытых контуров

Математик и инженер Пьер Безье разработал метод описания кривых с помощью четырех точек для каждого криволинейного сегмента. Две из них лежат на концах сегмента – это уже упоминавшиеся опорные точки, а две других отвечают за форму кривой и в общем случае лежат за пределам кривой. Такие точки называются управляющими и представляют собой как бы магниты, к которым притягивается кривая. Если опорная точка имеет управляющую точку, то следующий сегмент будет криволинейным. Если обе управляющие точки сегмента находятся на одной прямой с опорными – соответствующий сегмент будет прямолинейным.

Управляющие точки связаны с опорными точками с помощью управляющих линий, которые являются касательными к кривой в опорной точке. Если управляющие точки на каждом конце сегмента находятся по разные стороны от криволинейного сегмента, то этот сегмент будет иметь S-образную форму, а если по одну сторону от сегмента – то U-образную.

Типы опорных точек:

- гладкие точки – при соединении в этой точке двух сегментов траектория остается гладкой. Как правило, если контур не выделен и местоположение опорных точек не видно, распознать, где находиться гладкая точка, практически не возможно. Гладкие точки отвечают за недопущение резкого изменения направления линии. Для каждой гладкой точки существует две управляющих точки, расположенных на одной прямой;
- угловые точки – в этих точках происходит резкое изменение направления. Существует три вида угловых точек:
    - Прямолинейная угловая точка – в ней соединяются под определенным углом два прямолинейных сегмента. Для такой опорной точки не существует управляющих точек.
    - Криволинейная угловая точка – в ней соединяются два криволинейных сегмента, и направление контура резко меняется. Для такой опорной точки существует две независимых (не лежащих на одной прямой) управляющих точки.

- Комбинированная угловая точка – точка соединения прямолинейного и криволинейного сегментов. Для нее существует только одна управляющая точка.

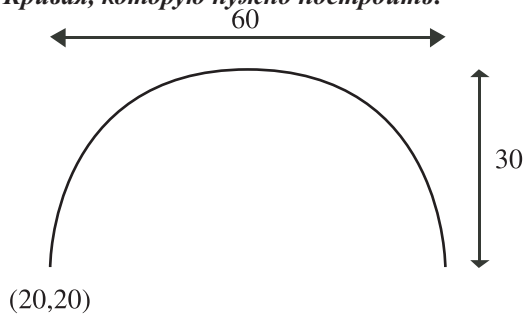
Лабораторная работа №6. Кривые Безье

Цель работы: приобрести навыки построения кривых Безье в заданных координатах.

Среда выполнения: Adobe Illustrator (любая версия) или Corel Draw (любая версия).

Пример создания кривой Безье (Adobe Illustrator)

Кривая, которую нужно построить:

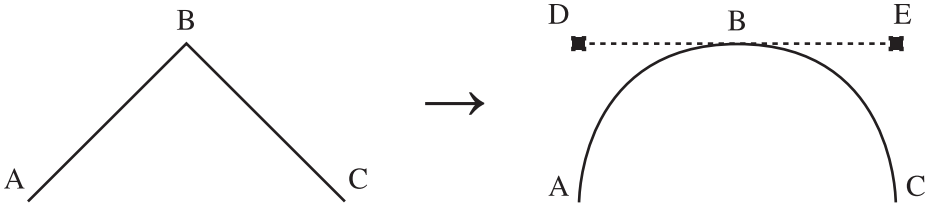


Подготовительные операции:

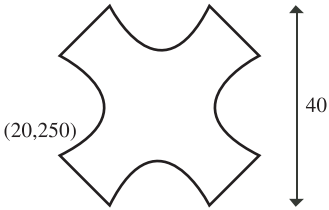
1. *Создать новый документ (File - New; параметры документа – по умолчанию);*
2. *Включить сетку (View - Show Grid) и привязку к сетке (View - Snap to Grid). При необходимости – изменить параметры сетки (Edit - Preferences - Guides & Grid).*
3. *Включить окно информации для отображения координат курсора (Window - Info).*

Основные операции:

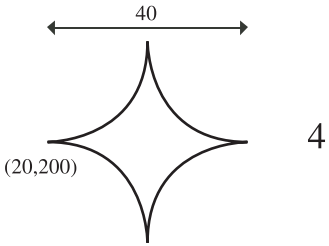
1. *Ставим опорные точки A, B и C с помощью инструмента «перо» (панель инструментов - Pen) в точках (20,20), (50,50) и (80,20).*
2. *Преобразуем точку B в точку искривления с помощью инструмента «уголок» (панель управления - Convert Anchor Point Tool) и растаскиваем появившиеся управляющие точки D и E в позиции (20,50) и (80,50).*



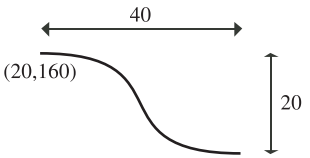
Задание лабораторной работы №6: построить следующие кривые в заданных координатах используя минимальное количество опорных точек.



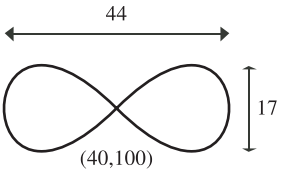
5



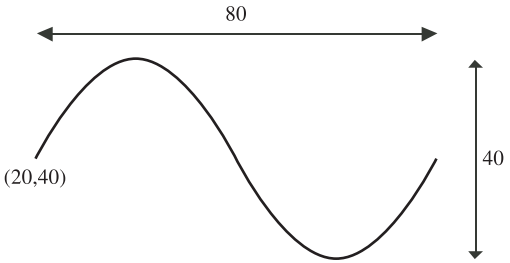
4



3



2



1

Примечания:

1. Применение инструмента «уголок» к опорной точке последовательно преобразует ее из прямоугольной угловой в гладкую точку и обратно. Применение инструмента «уголок» к управляющей точке позволяет преобразовать гладкую точку в криволинейную угловую.
2. В Corel Draw последовательность действий остается такой же, только для установки опорных точек используется инструмент FreeHand Tool, а все операции с управляющими точками (преобразование сегмента из прямолинейного в криволинейный и обратно, изменение типа угловой точки и т.д.) производятся инструментом Shape Tool с использованием контекстного меню.

Кроме непосредственного создания векторных контуров с помощью кривых Безье, можно получать их из графических примитивов. Во все векторные пакеты включены инструменты, воспроизводящие такие векторные примитивы, как прямоугольник (Rectangle Tool), в том числе со скругленными углами (Rounded Rectangle Tool), эллипс (Ellipse Tool), многоугольник (Polygon Tool), звезда (Star Tool) и др. При этом применение инструмента с клавишей Shift позволяет построить симметричную фигуру (круг, квадрат и т.п.), а применение с клавишей Alt – построить фигуру «от центра» (по умолчанию фигуры строятся от верхнего левого угла). Одиночный щелчок выбранным инструментом позволяет создать в указанном месте соответствующий объект, задав его параметры численно в открывшемся диалоговом окне.

Примечание:

В Corel Draw используются другие клавиши: для симметричной фигуры – Ctrl, а для рисования «от центра» – Alt.

Одним из ключевых преимуществ векторного представления графической информации является сохранение качества отображения границ объектов при их деформации. Это связано с тем, что при любых деформациях фактически просто пересчитываются координаты опорных и управляющих точек сегментов, а затем заново пересчитывается собственно граница сегмента. В результате точность воспроизведения линии в каждый момент времени ограничена только разрешающей способностью выводного устройства.

В большинстве векторных пакетов реализованы следующие инструменты для деформации объекта (названия взяты по Adobe Illustrator):

1. Scale Tool – масштабирование;
2. Rotate Tool – поворот;
3. Twirl Tool – перекручивание;
4. Reflect Tool – отражение:
5. Shear Tool – сдвиг.

Первые два инструмента не нуждаются в комментариях. Twirl Tool – скручивает изображение относительно выставленного центра внутри окружности или эллипса. Reflect Tool строит зеркальное отображение объекта относительно оси отражения (горизонтальной, вертикальной или расположенной под произвольным

углом). Shear Tool еще называют инструментом наклона. Его действие можно представить следующим образом: представим себе, что исходный объект был вписан в прямоугольник, а затем осуществим параллельный сдвиг любых двух противоположных сторон прямоугольника, превращая его в параллелограмм. Этот инструмент используется, в частности для, создания эффекта перспективы.

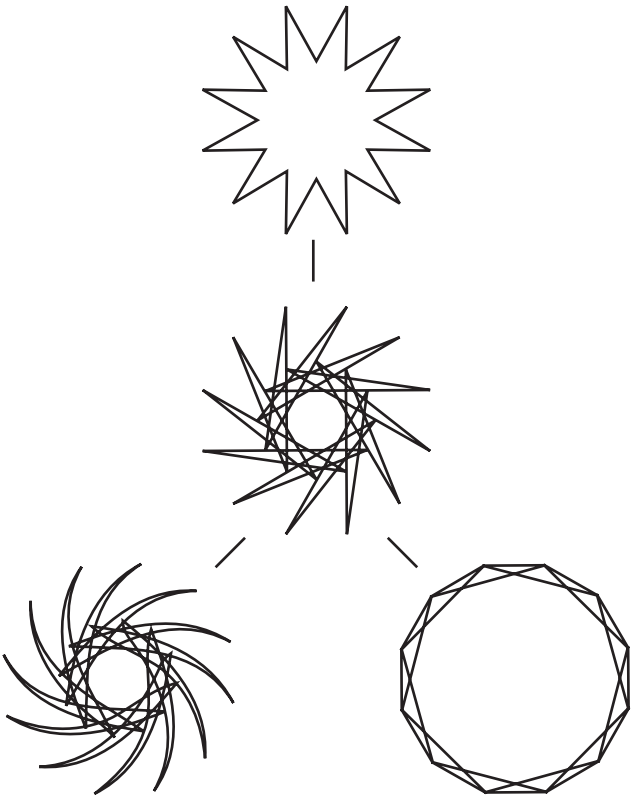
Инструменты трансформации могут быть применены как к объекту целиком, так и к отдельным точкам, выделенным с помощью инструмента прямого выделения (Direct Selection Tool), обозначаемого белая стрелка.

Лабораторная работа №7. Трансформация векторных объектов

Цель работы: приобрести навыки трансформации контуров и отдельных сегментов.

Среда выполнения: Adobe Illustrator (любая версия).

Задание лабораторной работы №7: построить следующие объекты, осуществляя последовательную трансформацию всего объекта или отдельных точек.



Примечание:

В Corel Draw масштабирование, поворот, зеркальное отражение и сдвиг доступны через меню Arrange - Transform.

К каждому векторному объекту могут быть применены параметры обводки и заливки.

Параметры обводки (Stroke) – это в первую очередь, толщина обводки, тип линии (сплошная, пунктир и т.д.), цвет и др. Отдельно следует отметить такой параметр обводки, как стыки сегментов (доступно на палитре Storke). Этот параметр отвечает за формирование изображения на угловых точках соединения сегментов. Это особенно важно для толстых линий. В общем случае существует три варианта таких стыковок: Mitered Joins – оформляет соединения сегментов в виде углов, вершины которых определяются точками пересечений краев линий; Rounded Joins – создает стыки округлой формы; Beveled Joins – срезает углы в точках соединения сегментов контуров.

В качестве заливки могут использоваться следующие варианты:

- сплошная заливка (Solid Fill) – заливка одним цветом;
- градиентная заливка (Gradient Fill) – заливка переходом (растяжкой) от одного цвета к другому; количество цветов в переходе и тип перехода задается в параметрах;
- заливка узором (Pattern Fill, Swatch Fill) - заполняет объект растровыми или векторными образцами.

Примечание:

В Adobe Illustrator можно заливать как замкнутые, так и открытые контуры. В последнем случае разомкнутые края контура как бы соединяются невидимой прямой линией, ограничивающей область заливки. В Corel Draw заливать можно только замкнутые контуры.

Для залитых объектов становится актуальным еще один способ создания векторных объектов (наряду с непосредственным созданием методом Безье, использованием графических примитивов и трансформацией кривых) – логические операции над замкнутыми кривыми.

Логические операции являются одним из вариантов объединения нескольких контуров в один объект. Кроме них, таким вариантом является группирование. Группирование – процесс объединения нескольких объектов, пространственные взаимоотношений которых необходимо сохранить постоянными. Группы могут состоять из одного контура либо содержать неограниченное число объектов. При группировании сами объекты остаются неизменными, и после разгруппирования невозможно определить, были ли эти объекты когда-либо сгруппированы.

Логические операции над объектами, в отличие от группирования, в общем случае необратимы, то есть в результате их применения к нескольким контурам взамен них получается новый контур со своими опорными и управляющими

точками. В Adobe Illustrator логические операции доступны через палитру Pathfinder (Window - Pathfinder).

Примечание:

В Adobe Illustrator версии CS и выше на палитре логических операций (Pathfinder) появилась кнопка Expand. Таким образом, простое применение логических операций является обратимым, пока не нажата эта кнопка. Это удобно с точки зрения возможностей отката в случае осуществления ошибочных действий, но имеет и отрицательную сторону – использование контуров, полученных путем применения логических операций без применения Expand, в других эффектах (например, группах перетекания) часто приводит к ошибкам.

Всего можно выделить пять основных логических операций.

1. Unite (объединение) (OR) позволяет объединить выделенные перекрывающиеся объекты. В результате в новый контур будут включены все предварительно выделенные объекты. Все внутренние контуры пересечения объектов удаляются. Новому объекту присваиваются атрибуты стиля окраски самого верхнего объекта. Если какие-либо объекты полностью попадают внутрь других объектов, они «поглощаются» более крупными объектами. Если в каком-либо контуре были отверстия, они будут удалены из общего контура.
2. Intersection (пересечение) (AND) в результате дает пересечение (общую часть) выделенных контуров. Любые части контуров, не попавшие в область пересечения, удаляются. Если у выделенных объектов нет общей области, в результате применения команды пересечения получится пустое множество, то есть все выделенные объекты будут удалены. Если один из контуров полностью содержится внутри других, он и будет результатом пересечения. У контура, полученного при пересечения, будут те же атрибуты стиля окраски, которые имеет самый верхний контур.
3. Exclude (исключение) (XOR) обратна функции Intersection (пересечение). При ее применении области пересечения, удаляются, а все остальные области, объединяются. Контур имеет стиль окраски верхнего из объектов до применения команды.
4. Minus Front (Минус передний) вычитает из самого нижнего контура все остальные выделенные контуры, расположенные над ним.
5. Minus Back (Минус задний) обратна функции Minus Front. Она вычитает из самого верхнего контура все остальные выделенные контуры, расположенные под ним.

Отметим, что эффект, аналогичный применению логической операции исключения, возможен, если применить к нескольким контурам команду создания составной линии (Меню – Object – Compound Path – Make). В результате сразу создается единый контур, то есть нет необходимости в применении команды Expand. При использовании команды создания составной линии над несколькими объектами область нового контура будет заполненной, если в ней пересекалось

нечетное количество областей исходных объектов, и пустой, если это количество было четным.

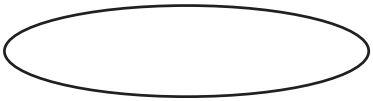
Лабораторная работа №8. Создание сложных контуров с помощью логических операций

Цель работы: приобрести навыки использования логических операций для создания ложных контуров.

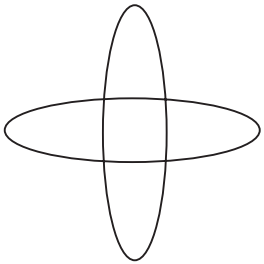
Среда выполнения: Adobe Illustrator (любая версия) или Corel Draw (любая версия).

Пример создания сложного контура (Adobe Illustrator)

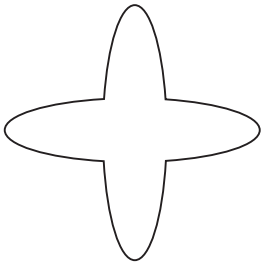
1. *Создаем эллипс (Инструменты - Ellipse Tool)*



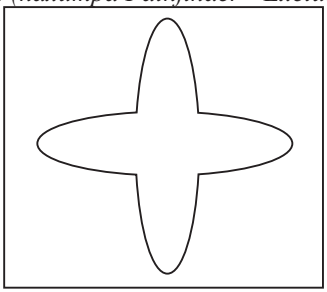
2. *Дублируем эллипс и поворачиваем копию на 90 градусов. Проще всего воспользоваться командой Меню - Object - Transform - Rotate, ввести угол 90° и нажать кнопку Copy – создастся новый развернутый объект. Если дублирование и поворот осуществлялись другим способом, может быть актуальным выравнивание обоих объектов относительно их центров. Для этого можно воспользоваться палитрой Align.*



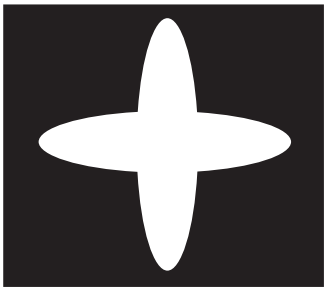
3. *Применяем к эллипсам операцию объединения (палитра Pathfinder - Unite).*



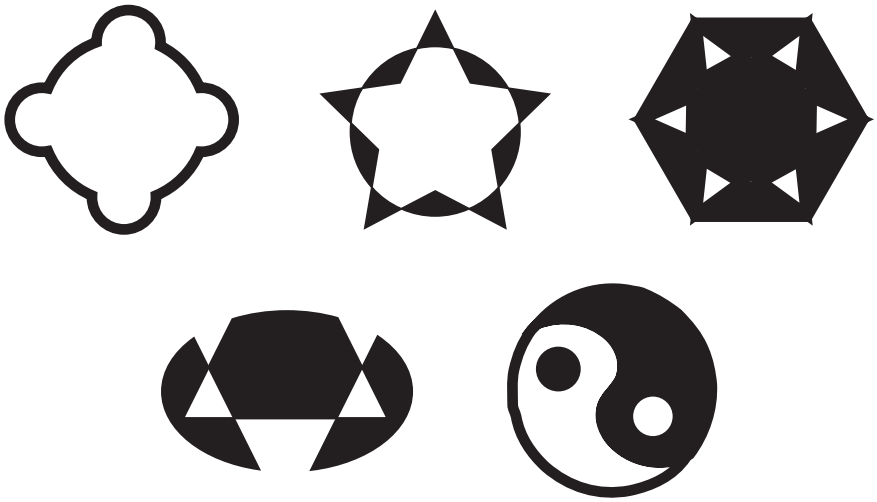
4. *Создаем прямоугольник вокруг получившейся на предыдущем этапе фигуры, выравниваем все фигуры с помощью палитры Align и применяем к ним операцию исключения (палитра Pathfinder - Exclude).*



5. *Заливаем получившийся объект черным цветом (палитра Color).*



Задание лабораторной работы №8: построить следующие объекты, используя только графические примитивы, операции трансформации и логические операции. Результатом каждого действия должен стать единый контур.



Примечание:

В Corel Draw логические операции объединения, пересечения и вычитания доступны через меню Arrange - Shape - соответствующая операция, а исключение (XOR) является отдельным пунктом меню Arrange - Combine.

Одним из самых мощных инструментов векторной графики являются перетекания (Blend). Этот инструмент позволяет создать ряд трансформированных контуров между двумя концевыми контурами. Промежуточные контуры видоизменяются, превращаясь из одного концевого контура в другой. Эта трансформация захватывает как форму, так и цветовые атрибуты объекта. Перетекание может быть осуществлено как по кратчайшему расстоянию между объектами (по умолчанию), так и по произвольному контуру.

Количество промежуточных фигур в переходе может быть задано тремя способами: через количество шагов, через дистанцию между фигурами или автоматическим подбором, при котором будет подобрано минимальное количество шагов, чтобы реализовать гладкий цветовой переход. В Adobe Illustrator эти параметры доступны в меню Object - Blend - Blend Options

Еще раз отметим, что достраиваемые объекты будут не только принимать промежуточные формы, но и обладать промежуточными значениями всех своих параметров. Такими параметрами являются, в частности, цвет заливки, цвет, толщина и тип обводки и т.д.

При построении группы перетекания необходимо указать начальный и конечный контур. При этом Adobe Illustrator позволяет указать на этих контурах пары точек, которые должны переходить друг в друга, что может влиять на построение промежуточных шагов.

Построенная группа перетекания является единым объектом-функцией. Это значит, что если изменить один из концевых контуров, входящих в группу перетекания, вся группа перестроится заново. Чтобы получить возможность работы с отдельными промежуточными контурами, группу перетекания нужно разбить на части (например, командой Expand в Adobe Illustrator). После этого все промежуточные значения перетекания становятся самостоятельными объектами.

Значительно расширяет возможности использования групп перетекания реализация построения промежуточных контуров не по кратчайшему расстоянию, а вдоль произвольного пути. Для этого нужно выделить уже созданную группу перетекания и еще один контур-путь и вызвать команду меню Object - Blend - Replace Spine.

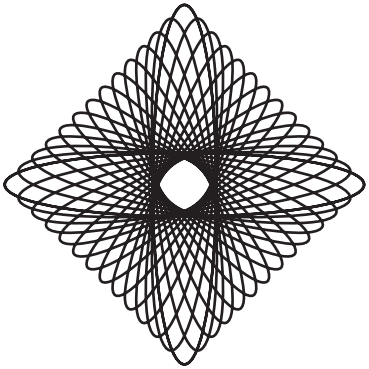
Можно перечислить очень много бластей применения групп перетекания. Среди наиболее востребованных – создание векторных фоновых заливок, в том числе защитных муаров для документов.

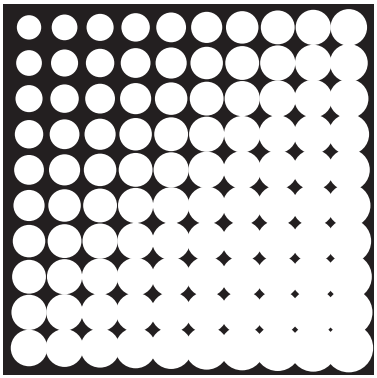
Лабораторная работа №9. Применение групп перетекания

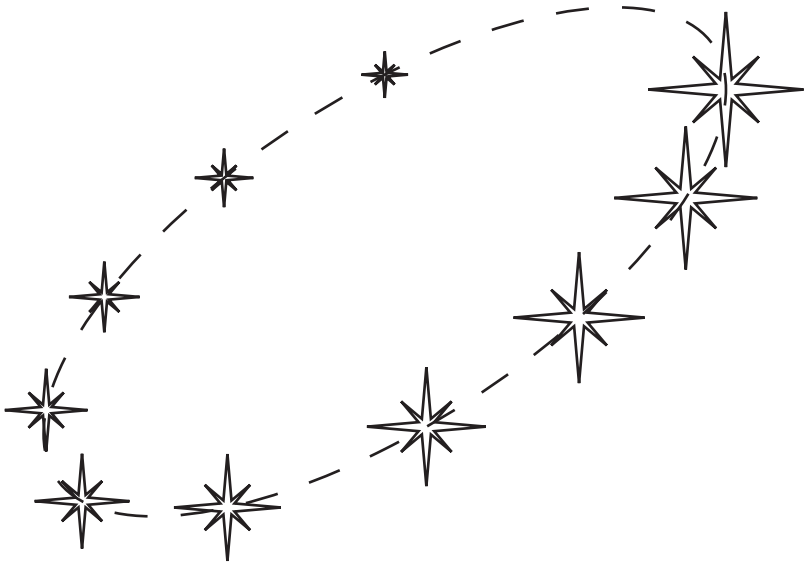
Цель работы: приобрести навыки использования групп перетекания.

Среда выполнения: Adobe Illustrator (любая версия) или Corel Draw (любая версия).

Задание лабораторной работы №9: воспроизвести следующие векторные объекты, используя группы перетекания.







Примечание:

Если в задании со звездочками исходная звездочка создавалась с помощью логических операций, необходимо применить команду Expand перед созданием группы перетекания.

Еще одним, достаточно неожиданным, приложением групп перетекания стала имитация объемных объектов (псевдо-3D). Основная идея здесь состоит в формировании светотени как основы для восприятия изображения объемным. Часто для этого группы перетекания используют совместно с градиентными заливками.

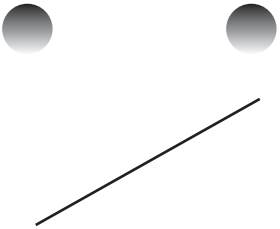
Лабораторная работа №10. Имитация объемных объектов с помощью групп перетекания

Цель работы: приобрести навыки использования групп перетекания и градиентных заливок для имитации объемных объектов.

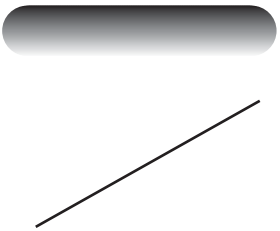
Среда выполнения: Adobe Illustrator (любая версия) или Corel Draw (любая версия).

Пример создания объемного объекта (Adobe Illustrator)

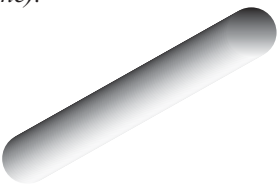
1. *Создаем две одинаковых окружности (Инструменты - Ellipse Tool) и заливаем их линейной градиентной заливкой от черного к белому цвету (Палитра Gradient). Рядом создаем отрезок прямой линии (Инструменты - Line Segment Tool).*



2. *Строим группу перетекания (Инструменты - Blend Tool) между двумя кружками с большим количеством шагов, например, около 100 (Меню - Object - Blend - Blend Options - Specified Steps).*



3. *Используем отрезок прямой как путь для группы перетекания (Меню - Object - Blend - Replace Spine).*



Задание лабораторной работы №10: воспроизвести следующие векторные объекты, используя группы перетекания.

Примечание:

Работая с текстом, лучше перевести его в кривые (Меню - Type - Create Outlines) перед использованием его в группах перетекания.

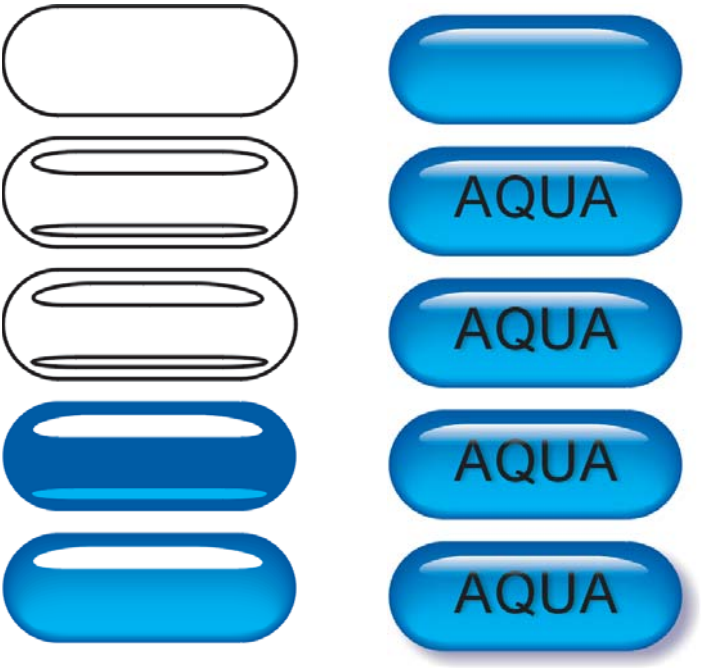
Лабораторная работа №11. Создание объемной кнопки

Цель работы: закрепить навыки работы с инструментами векторной графики.

Среда выполнения: Adobe Illustrator (любая версия) или Corel Draw (любая версия).

Задания лабораторной работы №11:

Задание 1. Воспользовавшись подсказкой – изображением результатов выполнения отдельных операций, создать прямоугольную объемную кнопку.



Примечание:

Белый блик в верхней части кнопки получен с помощью маски прозрачности. Маска прозрачности создается для конкретного векторного объекта (вкладка Transparency - Make Opacity Mask) и управляет прозрачностью этого объекта относительно низлежащих (черный цвет – прозрачно, белый – непрозрачно, серый – частично прозрачно в зависимости от яркости серого цвета). Таким образом, для получения искомого эффекта на маске необходимо создать объект, заполненный градиентной заливкой.

Задание 2. По аналогии с предыдущим заданием построить круглую объемную кнопку и кнопку произвольной формы (исходный контур задать в виде замкнутой кривой Безье).

РАЗДЕЛ 3. МНОГОПОЛОСНАЯ ВЕРСТКА

Верстка – монтаж полос издания заданного размера из составных элементов: наборных строк текста, заголовков, формул, таблиц, репродукций иллюстраций, украшений, колонцифр, колонтитулов и т. д.

Верстка неразрывно связана с дизайном издания, но в рамках данной дисциплины мы ограничимся технологическими аспектами верстки. Рассматриваться эти этапы будут на примере издательского пакета Adobe Indesign.

Любая верстка начинается с создания шаблона – разметки типовых положений полос текста и элементов оформления на форматном листе. Для этого используются страницы-шаблоны *(Master Pages)*. Все объекты, размещенные на этих страницах, будут повторяться на всех страницах, к которым применен шаблон. На шаблоне помещаются непечатные лини привязки *(Guides)*, размечающие поверхность листа для последующего точного размещения элементов верстки, и колонтитулы (колонцифры) – элементы, повторяющиеся на каждой странице издания.

Затем, руководствуясь шаблоном, создаются текстовые полосы – замкнутые объекты (в простейшем случае – прямоугольники), в которые будет импортироваться текст. Текст помещается в первую полосу, а затем, при необходимости его перетекания в другие полосы, организуется связь между полосами. Текст всегда импортируется без предварительного форматирования.

После размещения текстовых полос на верстку помещается графический материал. В простейшем случае просто импортируется графический файл, который затем при необходимости кадрируется. Если требуется непрямоугольная форма изображения, создается векторный объект нужной формы, и изображение импортируется в него. Взаимоположение текстовых полос и изображения реализуется одним из двух способов – путем создания фигурных (непрямоугольных) полос текста, обтекающих изображения, или присвоением изображению атрибута обтекания, приводящего к автоматическому обтеканию изображения накладывающимися на него полосами текста.

После размещения всех элементов верстки на форматных листах осуществляется форматирование текста внутри полос: оформление заголовков различных уровней, контроль висячих строк, переносов и других артефактов текстовой верстки. Для оформления заголовков и других выделяемых оформлением участков текста, как правило, применяются стили, работа с которыми аналогична реализациям в любом текстовом процессоре. Иногда вместо создания стиля используется взятие параметров оформления с оформленного сегмента текста с помощью специального инструмента (в Adobe Indesign – инструмент «пипетка»).

Особое внимание необходимо уделить требованиям к текстовому форматированию. Рассмотрим основные из них.

1. Висячие строки. Если абзац состоит из нескольких строк, одна последняя строка не может переходить на следующую полосу (страницу). Аналогично, одна первая строка абзаца не может оставаться в конце полосы, когда все остальные строки абзаца переходят на следующую полосу (страницу). Заголовок любого уровня не может остаться в конце полосы, если следующий за ним текст или заголовок более низкого уровня переходит на следующую полосу (страницу). Первый или последний элемент маркированного или нумерованного списка не может оставаться в конце полосы или, соответственно, переноситься на следующую полосу, если он состоит из одной строки.
2. Маркированные и нумерованные списки оформляются единообразно для всего издания. В списке у всех строк абзацев делается одинаковый отступ слева, чтобы маркеры или номера не перемежались по вертикали строками текста. Для этого в списке делается выступ первой строки (от-

рицательная красная строка). Положение текста в первой строке регулируется табулятором.

3. Верхние и нижние строки полос, расположенных на одном форматном листе (развороте), должны находиться на одном уровне.
4. Переносы. Запрещено использовать перенос в словах, если переносимое слово находится в конце полосы с продолжением на следующей (переворачиваемой) странице.
5. Неразрывные пробелы. Строка не может оканчиваться на инициал(ы), если за ними следует фамилия, переносящаяся на следующую строку. Аналогично, строка не может оканчиваться на цифру, если следующая за ней единица измерения переходит на следующую строку. Также строка не может заканчиваться на предлоге – он должен быть перенесен на следующую строку.

Для выполнения перечисленных требований, как правило, используют изменение межбуквенного и межстрочного интервала. При этом следует помнить, что межбуквенный и межстрочный интервалы нельзя изменять более чем на 5%.

Лабораторная работа №12. Верстка многополосного издания по фиксированному шаблону

Цель работы: приобрести навыки многополосной верстки и форматирования текста.

Среда выполнения: Adobe Indesign (CS или CS2).

Задание лабораторной работы №12: сверстать текст «Сказ про Федота Стрельца, удалого молодца» в соответствии с предоставленным примером шаблона.

1. ***Создать формные объекты полос текста – объекты, которые будут использоваться как рамки для размещения текста (используйте стандартные примитивы и логические операции над замкнутыми контурами, доступные во вкладке Pathfinder также как в ЛР 8), вставить несколько пустых страниц (вкладка Pages) и разместить на них формные объекты полос текста.***
2. ***Вставить текст в первый формный объект и, устанавливая связи между объектами, поместить весь текст на страницах (щелкайте левой кнопкой мыши по красному плюсику в конце очередной полосы, а затем по следующей полосе).***
3. ***Задать шрифтовое и абзацное оформление текста с учетом указанных выше требований к форматированию текста, получив окончательное количество заполненных страниц. Если последняя страница заполнена не полностью – выровнять на ней полосы так, чтобы они заканчивались на одном уровне.***
4. ***Отобрать необходимое количество иллюстраций – по одной на каждую страницу – таким образом, чтобы каждая иллюстрация хотя бы в какой-то мере отражала содержание страницы («портретное сходство»***

не обязательно, но смысл появления каждой иллюстрации на странице должен быть понятен). Создать маску для изображений и поместить изображения на страницах, вписав их в маски. Подписать каждое изображение.

6. ***Создать текст-колонтитул (название произведения) и поместить его на странице-шаблоне (Master Page) так, как показано в примере (используйте инструмент Type on Path Tool).***
7. ***Создать обложку издания, в оформлении которой нашли бы отражение графические элементы оформления издания.***

Образец шаблона для верстки

Потешник

Верьте аль не верьте, а жил на белом свете
Федот-стрелец, удалой молодец.
Был Федот ни красавец, ни урод, ни румян,
ни бледен, ни богат, ни беден,
ни в парше, ни в парче, а так, вообче.
Служба у Федота - рыбалка да охота.
Царю - дичь да рыба, Федоту - спасибо.
Гостей во дворце - как семян в огурце.
Один из Швеции, другой из Греции,
третий с Гавай - и всем жрать подавай!
Одному - омаров, другому - кальмаров,
третьему - сардин, а добытчик один!
Как-то раз дают ему приказ -
чуть свет поутру явиться ко двору.
Царь на вид сморчок, башка с кулачок,
а злобности в ем - агромадный объем.
Смотрит на Федьку,
как язвенник на редьку.
На Федьке от страха
намокла рубаха,
в висках застучало,
в пузе заурчало,
тут, как говорится, и
сказке начало…

Царь

К нам на утренний
рассол
Прибыл аглицкий посол,
А у нас в дому закуски -
Полгорбушки да мосол.
Снаряжайся, братец, в путь
Да съестного нам добудь -
Глухаря аль куропатку.
Аль ишо кого-нибудь.
Не смогешь - кого винить? -
Я должон тебя казнить.
Государственное дело,
Ты улавливаешь нить?..

Федот

Нешто я да не пойму
При моем-то при уму?
Чай, не лаптем ци хлебаю,
Соображаю что к чему.
Получается, на мне
Вся политика в стране:
Не добуду куропатку -
Беспременно быть войне.
Чтобы аглицкий посол
С голодухи не был зол.
Головы не пожалею,
Обеспечу разносол!..

Потешник

Слово царя тверже сухаря.
Пошлет на медведя - пойдешь на медведя,
а куды деваться - надо, Федя!
Или дичь и рыба - или меч и дыба.
Обошел Федот сто лесов, сто болот,
да все зря - ни куропатки, ни глухаря!
Устал, нет мочи, да дело к ночи.
Хоть с пустой сумой, а пора домой.
Вдруг видит - птица,лесная Голубица,
сидит, не таится, ружья не боится...

Голубица

Федот

Вот несчастье, вот беда,
Дичи нету и следа.
Подстрелю-ка голубицу,
Хоть какая да еда!

Голубица

Ты, Федот, меня не
трожь,
Пользы в этом ни
на грош -
И кастрюлю не
наполнишь,
И подушку не набьешь.

Федот

То ли леший нынче рьян,
То ли воздух нынче пьян,
То ли в ухе приключился
У меня какой изъян?

Голубица

Не твори, Федот, разбой,
А возьми меня с собой.
Как внесешь меня в светелку -
Стану я твоей судьбой.

Федот

Что за притча - не пойму?..
Ладно, лезь ко мне в суму!..
Там на месте разберемся,
Кто куды и что к чему!

Лабораторная работа №13. Верстка научно-технического текста

Цель работы: приобрести навыки многополосной верстки и форматирования научно-технического текста.

Среда выполнения: Adobe Indesign (CS или CS2).

В этой ЛР верстку научно-технического текста необходимо произвести не по заданному шаблону, а выработав собственную композицию. Верстка должна подчиняться композиционным (художественным) и техническим правилам, с которыми можно ознакомиться в литературе [3, 9], а также на сайтах [1, 10, 14]. Ниже приводится минимальный перечень требований, которым должна удовлетворять верстка документов, наряду с требованиями к форматированию текстов, рассмотренным в предыдущей ЛР.

Верстка должна быть единообразной:

- высота всех полос одинакова и кратна кеглю шрифта основного текста;
- отбивки от текста элементов одного типа (сноски, иллюстрации, подписи к ним, таблицы, формулы и т. д.) идентичны по всему документу;
- элементы оформления (размеры шрифтов, абзацные отступы, межстрочные интервалы, выключка строк, оформление заголовков, подписей к рисункам и т.д.) идентичны по всему документу, независимо от редактора, в котором они выполнялись. В частности, одни и те же математические символы, набираемые в текстовом редакторе и в редакторе формул, а также входящие в состав иллюстраций, должны иметь одинаковый размер и начертание.

Верстка заголовков:

- точка в конце заголовка не ставится;
- предлоги, союзы и частицы в конце строки многострочного заголовка переносятся в начало следующей строки.

Верстка иллюстраций:

- иллюстрация размещается на той же полосе, что и текст со ссылкой на нее, или на развороте с таким текстом;
- все иллюстрации должны иметь содержательные подписи вида «Рисунок 1. Название рисунка» внизу под рисунком;
- иллюстрацию нельзя заверстывать сразу после заголовка или в конце документа.

Верстка отдельных элементов:

- все виды сокращенных слов, в том числе сокращения метрических мер и технических величин, должны быть набраны тем же шрифтом, что и текст;
- в математических формулах латинские буквы набираются курсивом; греческие и русские буквы, а также цифры набираются прямым шрифтом; матрицы и векторы набираются полужирным шрифтом;
- дефис («черточка» внутри слова) и тире («черточка» между словами) при верстке различаются.

Задание лабораторной работы №13: сверстать текст, созданный в рамках ЛР 5 «Подготовка текстового материала». Шаблон верстки - свободный. Объем текста должен быть не менее 20 тыс. знаков. Верстка должна содержать не менее трех рисунков и двух таблиц и включать в себя оформленный список использованных источников.

Литература

1. MetaGuide. Руководство для разработчиков технической документации к программному обеспечению [электронный ресурс]. http://www.philosoft.ru/_subsites/tcportal/metaguide/
2. Большой полиграфический словарь [электронный ресурс]. www.pushel.ru
3. Гиленсон П.Г. Справочник художественного и технического редакторов. М.: Книга, 1988. 528 с.
4. Жвалевский, А. Adobe Illustrator CS в теории и на практике / А. Жвалевский, Ю. Гурский, Г. Корабельникова. - М.: Новое знание, 2004. - 608 с. - ISBN 5-94735-048-3.
5. Заботин, Ю. Практические советы по pre-press / Ю. Заботин. - М.: Майор, 2003. - 224 с. - ISBN 5-901321-54-5.
6. Кнут, Д.Э. Компьютерная типография / Д.Э. Кнут. - М.: Мир, 2003. - 672 с. - ISBN 5-03-003361-0.
7. Латышев, Л.К. Технология перевода / Л.К. Латышев. - М.: Academia, 2005. - 317 с. - ISBN 5-7695-2020-5.
8. Маркина, И.В. Основы издательских технологий / И.В. Маркина. - СПб.: БХВ-Петербург, 2005. - 368 с. - ISBN 5-94157-529-7.
9. Мильчин, А.Э. Справочник издателя и автора. Редакционно-издательское оформление издания. 2-е издание, исправленное и дополненное / А.Э. Мильчин, Л.К. Чельцова. - М.: ОЛМА-Пресс, 2003. - 800 с. - ISBN 5-224-04565-7.
10. Общие вопросы типографики и правил вёрстки [электронный ресурс]. http://www.prodtp.ru/index.php?showforum=23
11. Олспач, Т. Illustrator CS. Библия пользователя / Т. Олспач, Д. Олспач. - М.: Диалектика, 2004. - 656 с. - ISBN 5-8459-0672-5.
12. Пономаренко, С. Adobe InDesign. Дизайн и верстка / С. Пономаренко. - С-Пб.: BHV, 2000. - 544 с. - ISBN 5-8206-0089-4.
13. Романо, Ф. Принт-медиа бизнес. Современные технологии издательско-полиграфической отрасли. / Ф. Романо. - М.: Принт-Медиа центр, 2005. - 456 с. - ISBN 5-98951-007-1.
14. Форум разработчиков технической документации, технических писателей и переводчиков [электронный ресурс]. http://forum.philosoft.ru/cgi-bin/forum/ikonboard.cgi
15. Шерберн, К. Услуги в сфере цифровой печати. Как стать прибыльным / К. Шерберн. - М.: Принт-Медиа центр, 2006. - 192 с. - ISBN 5-98951-011-X

В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

Студенческая кафедра

Студенческая кафедра естественнонаучного факультета была создана в июне 1996 года. Одной из основных целей ее создания было обеспечение рабочих мест студентам для проведения практик, в особенности технологической и педагогической, раскрытие перед студентами широкого поля выбора деятельности в области профессионально-педагогического образования в дальнейшем. Задачей преподавателей, научных руководителей кафедры, было показать студентам возможность выбора, которого им в силу их ограниченного жизненного опыта, недостатка знаний было трудно осознать самостоятельно.

Деятельность студенческой кафедры началась с участия в мероприятиях по автоматизированной обработке тестирования первокурсников, созданию межкафедрального компьютерного класса и с работы в редакционно-издательском отделе университета. Все меньше и меньше требовалось вмешательства в ее деятельность преподавателей как руководителей, чем дальше развивалась студенческая кафедра, тем самостоятельнее в своих делах становились и студенты.

Одним из важнейших достижений кафедры было создание в 2006 г. Союза кураторов, главной задачей которого было ускорить и упростить процесс адаптации студентов первого курса, сразу же привлечь их в многогранную студенческую жизнь.

В настоящее время студенческую кафедру ЕНФ можно рассматривать как форму студенческого самоуправления. Участие в деятельности студенческой кафедры воспитывает в студентах такие качества, как творчество, самостоятельность, способность быстро принимать решения, формировать адекватную ответную реакцию в общении с другими людьми, выступать перед аудиторией. При этом в обстановке непринужденности, свободы, эмоционального подъема, характерных для подобной деятельности, у студентов формируются ценностные ориентации. Это позволяет студентам почувствовать и оценить свои возможности, создает условия для самоуправления студенческих коллективов и воспитывает в них профессиональные качества.

Наталия Федоровна Гусарова
Юрий Валерьевич Дорогов
Роман Владимирович Иванов
Александр Владимирович Маятин

**ИЗДАТЕЛЬСКИЕ СИСТЕМЫ.
КОМПЬЮТЕРНАЯ ИЗДАТЕЛЬСКАЯ ГРАФИКА**

Часть 1

Учебное пособие

В авторской редакции
Верстка: А.В. Маятин
Дизайн обложки: А.В. Маятин
Редакционно-издательский отдел Санкт-Петербургского государственного университета информационных технологий, механики и оптики
Зав. РИО Н.Ф. Гусарова
Лицензия ИД №00408 от 05.11.99
Подписано к печати 20.06.07
Тираж 100 экз Заказ № 1059 Отпечатано на ризографе