*1 слайд*

Здравствуйте, Уважаемые члены Государственной экзаменационной комиссии! Представляю вашему вниманию к защите выпускную квалификационную работу на тему: «АЛГОРИТМ ПРЕОБРАЗОВАНИЯ XML ФОРМАТА В СТИЛИЗОВАННЫЙ ТЕКСТ И ЕГО РЕАЛИЗАЦИЯ».

*2 слайд*

Сейчас составляется большое число электронной отчетности и документации. Они должны соответствовать стандартам принятых в той или иной организации. В связи с этим становится актуальным вопрос упрощения создания электронных документов.

*3 слайд*

Целью выпускной квалификационной работы является разработка XML парсера, обрабатывающего данные определенного вида. Для ее выполнения необходимо разработать модель хранения данных и их представления в файлах, описать алгоритм обработки файлов, реализующие данную модель. Поставленная цель позволит решить проблему отображения содержимого, вследствие непреднамеренного упущения какого-либо правила при составлении документа.

*4 слайд*

Новизна данной работы обусловлена использованием подхода разделения данных от представления. Исходные данные и правила их отображения должны находиться в разных файлах.

*5 слайд*

Для реализации подхода разделения данных от представления будет использован язык разметки XML. Содержимое документа, будь то заголовки абзацы, таблицы, рисунки, располагаются внутри соответствующих тегов. Некоторые из них будут содержать атрибуты, необходимые для дополнения информации об объекте.

*6 слайд*

Так, например в теге, отведенному для заголовков содержится информации об уровне данного заголовка. Данная информация позволит соотнести правила отображения для заголовков разного уровня.

*7 слайд*

Каждый абзац текста размещается внутри тега <p>

*8 слайд*

Существуют и другие структуры, которые может содержать документ. К таким относятся списки, как упорядоченные, так и неупорядоченные, код на языке программирования, математические объекты, в том числе формулы.

На левом верхнем скриншоте показан пример записи ненумерованного списка в файле содержимого. каждый объект списка записывается внутри тега <li> (list item), а весь список заключен в тег <ul> (unordered list)

На правом верхнем скриншоте пример записи нумерованного списка. Весь список заключается в другой тег <ol>, а также содержит атрибуты: type для обозначения вида перечисления, и атрибут sep отвечающий за разделитель между цифрой или буквой и элементом списка.

Левый нижний скриншот показывает пример записи кода на языке программирования. Код должен заключатся в тег <code>. целесообразно наделить необязательным атрибутом lang, который будет нести дополнительную информацию о языке, на котором написан текст, будь то язык программирования или язык разметки. Такая информация может использоваться, например, средствами для подсветки синтаксиса языка в различных редакторах.

На правом нижнем скриншоте показано, как формула нормального распределения должна быть записана. Подобного рода объекты должны располагаться в теге <math> в формате LaTeX. Такая унификация необходима для их последующей обработки, чтобы представить в более наглядном виде.

*9 слайд*

В файле представления, по-другому можно назвать файле-стиле содержимого, описываются правила отображения структурных элементов в документе.

*10 слайд*

Правила так же записываются на языке разметки XML. Заключаются в теги с тем же названием, что и объекты, на которые они распространяются. Внутри этих тегов каждое правило представляет собой также тег, с соответствующим правилу именем, и характерными атрибутами. Формализация правил отображения заключается в параметризации атрибутов.

*11 слайд*

Здесь приведены два скриншота, но относящиеся к одному объекту – заголовкам. здесь мы видим правила, которые используются для оформления документации, принятых в нашем Университете. Секция <special> нужна специальных настроек, применяемых к разным типам заголовков.

*12 слайд*

Алгоритм преобразования заключается в последовательной обработке файла содержимого и соотнесения правил отображения с теми структурами, которые были прочитаны. Полученные исходные данные позволят создавать документы любого типа, соответствующими инструментами, будь то docx, pdf, или html файл.

*13 слайд*

На этом слайде приведен пример работы XML парсера. Она прочитал содержимое документа, расположенного в одном файле, и соотнёс настройки отображения, находящиеся в другом файле, и вывел их для наглядности на экран.

*14 слайд*

В ходе проделанной работы была рассмотрена концепция отделения хранения данных и их представления через использование языка разметки XML. Был разработан алгоритм обработки файлов, который реализуют эту модель. Работу алгоритма демонстрирует специально созданная программа, которая выполняет функции XML-парсера. Путем использования различных фреймворков для создания документов можно расширить возможности программы. Этот подход позволит легко изменить внешний вид документа с нажатием одной кнопки, при условии описания необходимых правил для его отображения. Также модель хранения данных упростит их интеграцию в HTML разметку интернет-страниц.