

А. М. Елизаров
Е. К. Липачев
М. А. Малахальцев

ОСНОВЫ MathML.
Представление математических
текстов в Internet

Практическое руководство



*Посвящается 200-летию
Казанского университета*

**ОСНОВЫ MathML.
Представление математических
текстов в Internet**

Практическое руководство

А. М. Елизаров, Е. К. Липачев, М. А. Малахальцев

Казань
Издательство КМО
2003

Казанское математическое общество
Российская Федерация,
Татарстан, 420008, Казань,
Университетская, 17-
НИИ математики и механики
им. Н. Г. Чеботарева
Казанского государственного
университета

Kazan Mathematical Society
Chebotarev Institute of
Mathematics and Mechanics
Kazan State University
17 Universitetskaya Str.
Kazan, Tatarstan,
420008,
Russian Federation

e-mail: kmf@ksu.ru



Российский фонд фундаментальных исследований

Издание осуществлено при финансовой поддержке
РФФИ (проекты № 02-07-90230, 0303-07-90252)

УДК
ББК

Печатается по постановлению Редакционно-издательского
совета Казанского математического общества

Основы MathML. Представление математических текстов в Internet. Практическое руководство/ *А. М. Елизаров, Е. К. Липачев, М. А. Малахальцев.* – Казань: Издательство Казанского математического общества, 2003. – 56 с.

Руководство содержит рекомендации по подготовке математических текстов для публикаций в электронной форме с использованием языка MathML. Описаны технология MathML и программные средства, позволяющие конвертировать в MathML документы, подготовленные с помощью имеющихся стандартных технологий (LaTeX, Mathematica, Maple, Word).

Для научных работников, преподавателей, аспирантов и студентов, специализирующихся в области естественных наук.

УДК

ББК

© А. М. Елизаров, Е. К. Липачев, М. А. Малахальцев, 2003
© Казанское математическое общество, 2003

ОГЛАВЛЕНИЕ

Введение	4
1 Кратко о XML и сопутствующих технологиях	5
1.1 Правильно построенные XML–документы	5
1.2 Пример XML–документа	7
1.3 Действительные XML–документы и DTD	9
1.4 DTD для MathML–документов	14
1.5 Пространство имен	14
1.6 XSL–преобразование XML–документов	16
2 Разметка математических текстов по технологии MathML ...	22
2.1 Вводный пример	22
2.2 Особенности отображения в браузерах	25
2.2.1 Особенности отображения в Internet Explorer	25
2.2.2 Особенности отображения в браузере Mozilla	32
3 Синтаксис языка	33
3.1 Токены	35
3.2 Основные элементы	35
3.3 Индексы	36
4 Более сложные примеры	37
4.1 Формула сокращенного умножения	37
4.2 Основное тригонометрическое тождество	38
4.3 Формула с греческими буквами и символом частного дифференцирования	39
4.4 Матрица	40
4.5 Коммутативная диаграмма	41
5 Использование пакета Mathematica для подготовки Web–публикаций	42
5.1 Конвертация в пакете Mathematica 4.2	42
5.2 Конвертация в пакете Mathematica 5.0	44
6 Конвертация из TeX в MML	47
6.1 Установка пакета TeX4ht	47
6.2 Работа с пакетом TeX4ht	48
7 Метаданные в XML–документах	50
Полезные ссылки	54

Введение

Наше руководство предназначается для тех математиков, которые самостоятельно готовят свои тексты к публикации и “дружат” с компьютером. Как правило, они используют издательский пакет TeX, позволяющий в домашних условиях получить типографское качество оригинал–макета публикации. Однако в настоящее время основной становится электронная форма представления информации. Для математиков актуальной является проблема представления и обработки математических текстов в электронной форме. В основном эта проблема касается представления математических формул. Наиболее распространенное на данный момент решение — представление формул в виде графических файлов — неудовлетворительно с точки зрения структурной обработки математических текстов. Поэтому в 1999 году консорциумом W3C (<http://www.w3.org>) была начата разработка языка математической разметки MathML — Mathematical Markup Language. Язык MathML представляет собой подмножество языка разметки XML. Поскольку XML является новым способом организации Web–информации, позволяющим на более высоком уровне решать задачи программной обработки документов (в частности, задачу поиска), MathML способен изменить принципы организации и управления электронными публикациями по математике.

В настоящее время язык MathML постепенно становится стандартом представления математической информации в электронной форме в силу следующих причин:

- технология обработки данных на основе языка MathML реализует одну из основных тенденций современной информатики — разделение разметки и данных, поэтому она представляет широкие возможности многоуровневого структурирования данных и расширенного поиска;
- появилась возможность создания программного обеспечения, использующего технологию MathML;
- созданы и продолжают совершенствоваться программные средства, позволяющие конвертировать в MathML документы, подготовленные с помощью имеющихся стандартных технологий (таких, например, как LaTeX, Mathematica, Maple, Word). Из наиболее распространенных инструментов отметим редактор WebEq (www.dessci.com/en/products/webeq), конверторы TeX → MathML: TtM (<http://hutchinson.belmont.ma.us/tth/mml/>), TeX4ht (входит в стандартный пакет MikTeX), а также конвертор XML →

TeX (MikTeX). MathML поддерживается основными просмотрщиками: Internet Explorer (при установке соответствующего плагина), Mozilla, Amaya;

- технология MathML поддерживается системами Maple[®] и MathCAD 2001, а компания Wolfram Research предложила собственную концепцию использования технологии MathML, которая реализована в пакете Mathematica[®], в частности, в этом пакете предусмотрено сохранение документов в формате MathML.

Мы предполагаем, что читатель знаком с основами HTML, желательно также, чтобы он имел представление о системе TeX.

Настоящее руководство подготовлено и опубликовано в рамках проектов 02-07-90230 и 03-07-90252 Российского фонда фундаментальных исследований.

1. Кратко о XML и сопутствующих технологиях

1.1. Правильно построенные XML–документы

В руководстве повсеместно используется термин *разметка*. Под *разметкой* текста принято понимать расстановку в тексте специальных знаков (*кодов разметки*), с помощью которых осуществляется форматирование текста перед его отображением или печатью. Набор соглашений о разметки называют *языком разметки*. Один и тот же текст можно разметить, используя разные языки разметки. Например, следующий текст размечен с помощью HTML:

```
<html>
<body background="#FFFFFF">
  <h1>Что такое разметка</h1>
  <p>
    Текст обычно форматировать, выделяя определенные фрагменты
    курсивом, подчеркиванием и т. д. Это достигается путём расста-
    новки в тексте специальных значков, так называемых <I>кодов
    разметки</I>.
  </p>
</body>
</html>
```

Этот же текст можно разметить, например, с помощью LaTeX. Документ, сформатированный в MS Word, также содержит разметку, но она является скрытой.

Имеется стандарт SGML (Standard Generalized Markup Language), принятый в качестве международного стандарта для определения методов представления текстов в электронной форме. Отметим, что в соответствии с этим стандартом разработаны языки разметки HTML и XML.

В начале февраля 1998 года международная организация World Wide Web Консорциум (www.w3.org) рекомендовала спецификацию Extensible Markup Language 1.0, за которой закреплена аббревиатура XML. Прежде всего, отметим, что эта спецификация является основой для построения грамматики языков разметки, и лишь условно сам XML можно назвать языком разметки. Язык XML разработан Рабочей группой (заглавная буква используется при описании стандарта и в других документах), образованной в 1996 году. В [1] указаны цели создания этой спецификации, сформулированные в виде десяти заповедей. По своему замыслу технология XML должна обеспечить отделение информации от разметки, что позволяет производить обработку, поиск и представление информации на более высоком технологическом уровне.

На основе этой технологии обработка, поиск и представление информации переходят на совершенно иной уровень. Как отмечено в [2, 3], до сих пор Web была ориентирована на работу человека, но Web следующего поколения (условное название Semantic Web) должна в значительной мере опираться на машинную обработку информации, стандарту XML при этом отводится роль одной из ключевых технологий.

Спецификация XML используется как средство для построения грамматики других языков. В настоящее время создан ряд языков разметки, являющихся подмножествами XML, и, в частности, язык MathML, которому посвящено данное руководство.

Синтаксис XML построен на основе тегов, но в отличие от HTML, в котором множество тегов фиксировано, пользователь создает собственное множество тегов и задает структурные отношения между ними.

При создании документов необходимо учитывать правила, основная задача которых — в отделении данных и формата. Эти правила определены в [1] и состоят из следующих требований.

- Документы XML должны начинаться с **объявления XML**, в котором определяется версия XML, например,

`<?xml version="1.0">`

- Любому открывающему тегу должен соответствовать закрывающий тег.
- В XML учитывается регистр символов.
- Значения атрибутов, используемых в определении тегов, должны заключаться в кавычки.
- В любом документе XML должен содержаться один корневой элемент для всего документа.
- Элементы не должны перекрываться.
- Вся информация, размещенная между открывающим и закрывающим тегами, рассматривается как данные, при этом учитываются все символы форматирования (пробелы, конец строки, табуляция).
- В XML имеются зарезервированные символы, которые используются как элементы синтаксиса. В тексте эти символы нужно заменять последовательностями других символов, называемых *объектами (entities)*.

Зарезервированный символ	Объект
<	<
>	>
&	&
“	"
‘	'

Если перечисленные правила выполнены, то говорят, что документ *правильно построен (well-formed)*, и только в этом случае он является *документом XML*. Такой документ будет правильно отображен в браузере, в ином случае будет выдано сообщение об ошибке. Для чтения документов XML используется программный модуль, называемый процессором XML.

1.2. Пример XML-документа

На простом примере покажем особенности синтаксиса XML. Опишем электронный журнал, причем включим в описание только название журнала и адрес, а также несколько статей с указанием их названия и авторов.

Пример 1.2.1. Листинг файла Example1.xml. Простейшее описание электронного журнала.

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- XML - пример -->
<journal>
<title>Lobachevsky's Journal</title>
<contacts>
<address>Kazan State University</address>
<url>ljm.ksu.ru</url>
</contacts>
<articles>
<article ID="1">
<title>MathML and TeX</title>
<author>M. Malakhaltsev</author>
</article>
<article ID="2">
<title>MathML and RDF</title>
<author>E. Lipachev</author>
</article>
</articles>
</journal>
```

Первая строка приведенного кода является объявлением разметки. Атрибут **version** является обязательным и указывает на версию XML, применяемую для структурирования. Рабочая группа XML намеревается давать последующим версиям спецификации номера, отличные от “1.0”. Получая документы, версию которых они не поддерживают, процессоры должны выдавать соответствующие сообщения. Если в документе есть вставки на русском языке, нужно указать кодировку с помощью атрибута **encoding**. Анализатор XML требовательно относится к разбираемому коду и в случае обнаружения “непонятных” мест просто прекращает обработку. Использование русских букв в документе без атрибута **encoding** в объявлении разметки воспринимается как ошибка.

Во второй строке показано, как добавить комментарии, — правила те же, что и в HTML.

В последующих строках объявлены элементы. Элемент состоит из открывающего и закрывающего тегов, а также данных между этими тегами. Имена не могут содержать пробелы, кроме того, имеет значение регистр символов. Одни элементы могут быть вложены в другие, как, например, элемент **<author> </author>** из приведенного примера вложен в элемент **<article> </article>**, который в свою очередь вложен в элемент **<articles> </articles>**. Один из элементов документа должен содержать все остальные,

этот элемент называют *корневым* или *элементом документа*. В нашем примере таким элементом является `<journal> </journal>`. Таким образом, элементы документа должны образовывать древовидную структуру.

У элементов могут быть атрибуты, которые содержат дополнительную информацию об элементе. В приведенном примере элемент `<article> </article>` содержит атрибут **ID**, с помощью которого каждой статье присваивается числовой идентификатор.

Сохраним этот файл с расширением `.xml` и откроем для просмотра в каком-нибудь браузере, например, MS Internet Explorer. В результате получим представление XML–документа в виде дерева



```
<?xml version="1.0" encoding="windows-1251" ?>
- <journal>
  <title>Lobachevsky' Journal</title>
  - <contacts>
    <address>Kazan State University</address>
    <url>ljm.ksu.ru</url>
  </contacts>
  - <articles>
    - <article ID="1">
      <title>MathML and TeX</title>
      <author>M. Malakhaltsev</author>
    </article>
    - <article ID="2">
      <title>MathML and RDF</title>
      <author>E. Lipachev</author>
    </article>
  </articles>
</journal>
```

Это представление может разочаровать. Для отображения XML–документов в привычном виде используется, например, XSLT–технология (см. раздел 1.6).

1.3. Действительные XML–документы и DTD

С каждым XML–документом можно связать DTD–объявления. Технология DTD (Document Type Definition) разработана как средство описания структуры документа и представляет собой набор инструкций для описания составных частей документа и порядка их следования. DTD определяет множество правил для тегов и атрибутов, допустимых в XML–документе, а также порядок и число вложений тегов, т. е. задается грамматика для опре-

деленного типа документов. Для единичного XML–документа создавать DTD не целесообразно, но при разработке некоторого стандарта документа, например, анкеты, желательно написать DTD, чтобы “отсечь” документы, не отвечающие стандарту. О существовании DTD надо помнить и в случае, если XML–документ включается в уже работающую систему без учета грамматики, определённой в DTD этой системы, Ваш документ не будет действительным и не пройдет обработку.

Документ XML имеет три уровня корректности: лексический, синтаксический и семантический. Если документ правильно построен, например, не нарушена вложенность тегов, все теги закрыты и т. д., то он лексически корректен. Синтаксическая корректность означает, что документ соответствует правилам DTD. Таким образом, каждый набор DTD–правил определяет стандарт некоторого класса документов. Например, для документов MathML используется набор правил **mathml2.dtd**.

Инструкции DTD предназначены для XML–процессора (например, встроенного в браузер), анализирующего содержание документа перед его отображением. Если в документе есть ссылка на набор DTD–правил, и он синтаксически им удовлетворяет, то он называется *действительным (valid)* или *корректным*, используется также термин *допустимый* документ.

Инструкции DTD можно записать в отдельном файле или же включить непосредственно в XML–документ. Файл с таблицей DTD является обычным текстовым файлом и может быть создан в любом текстовом редакторе. Для рассматриваемого примера описания электронного журнала содержание DTD–файла может быть таким.

Пример 1.3.1. Файл example1.dtd.

```
<!-- DTD for journal XML -->
<!ELEMENT journal (jrntitle, contacts, articles)>
<!ELEMENT jrntitle (#PCDATA)>
<!ELEMENT contacts (address,url)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT articles (article)*>
<!ELEMENT article (title, author+)>
<!ATTLIST article
      id ID #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
```

Для включения внешнего файла DTD в документ XML используется тег

<!DOCTYPE имя_корня SYSTEM "имя_файла.dtd">

Имя, указанное за словом DOCTYPE, должно совпадать с именем корневого элемента.

Пример 1.3.2. Файл example2.xml.

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- DTD for journal XML -->
<!DOCTYPE journal SYSTEM "example.dtd">
<journal>
<jrntitle>Lobachevsky&apos;Journal</jrntitle>
<contacts>
<address>Kazan State University</address>
<url>ljm.ksu.ru</url>
</contacts>
<articles>
<article ID="1">
<title>MathML and TeX</title>
<author>M. Malakhaltsev</author>
</article>
<article ID="2">
<title>MathML and RDF</title>
<author>E. Lipachev</author>
</article>
</articles>
</journal>
```

Определения DTD можно включать непосредственно в XML-файл. Инструкции DTD размещают в начале в контейнере

<!DOCTYPE имя_корня [

...

]>

Отметим, что блок DTD заканчивается символами]>.

Пример 1.3.3. Тот же пример, но с внутренним DTD.

```
<?xml version="1.0" encoding="windows-1251"?>
<!DOCTYPE journal [
<!ELEMENT journal (jrntitle, contacts, articles)>
<!ELEMENT jrntitle (#PCDATA)>
<!ELEMENT contacts (address,url)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT articles (article)*>
<!ELEMENT article (title, author+)>
```

```

<!ATTLIST article
      id ID #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
]>
<!-- Конец DTD определений -->
<journal>
<jrntitle>Lobachevsky&apos;Journal</jrntitle>
<contacts>
<address>Kazan State University</address>
<url>ljm.ksu.ru</url>
</contacts>
<articles>
<article ID="1">
<title>MathML and TeX</title>
<author>M. Malakhaltsev</author>
</article>
<article ID="2">
<title>MathML and RDF</title>
<author>E. Lipachev</author>
</article>
</articles>
</journal>

```

Поясним на основе этого набора DTD–правил наиболее существенные детали DTD. Как уже было отмечено, с помощью правила **<!DOCTYPE** указывается местонахождение DTD. Для описания элементов используется правило

<!ELEMENT имя_элемента содержание>

Для описания содержания используется довольно простая система обозначений. Круглые скобки в содержании означают, что элемент составной. Так, в приведенном примере элементы **journal**, **contacts**, **articles** и **article** составные. Запятая в скобках используется в качестве разделителя элементов. Вместо запятой можно использовать символ “|” (вертикальная черта), который разделяет варианты содержимого, если их у элемента несколько. Например,

<!ELEMENT phone (#PCDATA|EMPTY)>

определяет, что элемент **phone** может содержать символьные данные (номер телефона) или оставаться пустым (это указано с помощью ключевого слова **EMPTY**). Скобки задают также порядок появления элементов в документе.

Например,

<!ELEMENT contacts (address,url)>

указывает, что элемент **contacts** содержит элемент **address**, и этот элемент должен быть первым, а также элемент **url**. Кроме того, можно задать количество повторений элемента, поставив один из знаков “+” (плюс), “*” (звёздочка) или “?” (знак вопроса). Эти знаки можно записывать не только после элементов, но и после скобок — в этом случае действие знака распространяется на множество элементов, заключённых в скобки. Знак “+” означает, что элемент или множество элементов встречаются в документе один или более раз. Знак “*” указывает, что в документе может быть любое число вхождений данного элемента, в частности, элемент может отсутствовать. Знак “?” используется в тех случаях, когда элемент или множество элементов встречаются не более одного раза. Например,

<!ELEMENT article (title, author+)>

указывает, что элемент **article** содержит элемент **title** и он должен быть первым, а также один или более элементов **author**.

Если элемент содержит символьные данные, то при его описании используется содержание (**#PCDATA**). Ключевое слово **EMPTY** в содержании документа указывает, что это пустой элемент. Как примеры пустых элементов приведём теги **
** и **<hr>** в HTML. Ключевое слово **ANY** в содержании означает, что элемент может содержать как символьные данные, так и другие элементы.

Если у элемента есть атрибуты, как, например, атрибут **ID** у элемента **article**, то необходимо использовать инструкцию

<!ATTLIST	имя_элемента	
	имя_атрибута	значение_по_умолчанию
	тип
	значение_по_умолчанию>
	имя_атрибута	
	тип	

В этом объявлении необходимо перечислить все атрибуты, которые могут использоваться с данным элементом. Типы атрибутов можно разбить на три группы: строковые атрибуты, маркированные и перечислимые. Строковые атрибуты описываются ключевым словом **CDATA**, и в документе такие атрибуты могут содержать любые символьные данные. Маркированные атри-

буты включают в себя predefined атрибуты **ID**, **IDREF**, **IDREFS**, **ENTITY**, **ENTITIES**, **NMTOKEN** и **NMTOKENS**. Эти атрибуты предназначены для конкретного использования, например, атрибут **ID** задаёт уникальный идентификатор для элемента в документе. В нашем примере с помощью **ID** каждой статье присваивался числовой идентификатор. Подробности о назначении этих атрибутов можно найти в [4]. Если атрибут содержит список значений, то атрибут называется перечислимым. Например,

```
<!ELEMENT volume (#PCDATA)>
<!ATTLIST volume
    month (1|2|3|4|5|6|7|8|9|10|11|12) #REQUIRED>
```

Параметр **#REQUIRED** определяет, что данный атрибут является обязательным. Есть ещё два параметра: **#IMPLIED**, указывающий, что атрибут является необязательным, и **#FIXED**, требующий, чтобы только указанное в атрибуте значение использовалось в документе.

1.4. DTD для MathML–документов

Для документов, подготовленных по спецификации MathML, разработано несколько вариантов DTD. Они доступны по адресу <http://www.w3.org/Math/DTD/> и имеют следующее назначение:

mathml2.dtd — основная версия (так называемая “stand alone” DTD for MathML);

xhtml-math11-f.dtd — версия, предназначенная для использования в XHTML (“Combined XHTML + MathML DTD”).

При использовании этих DTD можно указать Web-адрес, например,

```
<!DOCTYPE math SYSTEM
    “http://www.w3.org/Math/DTD/mathml2/mathml2.dtd”>
```

или же указать локальное расположение DTD–файла (если он уже “скачан”)

```
<!DOCTYPE math SYSTEM “mathml2.dtd”>
```

1.5. Пространство имен

Спецификация XML Namespaces (пространство имен XML) [5, 7] рассчитана на разрешение проблем, связанных с конфликтами имен. Поскольку множество тегов в XML не фиксировано, как, например, в HTML, пользователь имеет возможность выбирать имена тегов. При подготовке сложных документов, например, использующих несколько DTD, некоторые теги,

разные по своему назначению, могут получить одинаковые имена. Поясним эту ситуацию простым примером.

Пример 1.5.1.

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- Конфликт имен title -->
<journal>
<title>Lobachevsky&apos;Journal</title>
<url>ljm.ksu.ru</url>
<articles>
  <article ID="1">
    <title>MathML and TeX</title>
  </article>
  <article ID="2">
    <title>MathML and RDF</title>
  </article>
</articles>
</journal>
```

В этом примере имя **title** используется для тега, содержащего название журнала, и тега с названием статьи. Заметим, что даже при наличии конфликта имен в документе он может без проблем отображаться в браузере, но более сложная обработка документа может оказаться невозможной.

Пример 1.5.2. Разрешение конфликта имен с помощью Namespaces.

```
<?xml version="1.0" encoding="windows-1251"?>
<journal xmlns:x="http://www.kcn.ru/one"
          xmlns:y="http://www.kcn.ru/two">
<x:title>Lobachevsky&apos;Journal</x:title>
<url>ljm.ksu.ru</url>
<articles>
  <article ID="1">
    <y:title>MathML and TeX</y:title>
  </article>
  <article ID="2">
    <y:title>MathML and RDF</y:title>
  </article>
</articles>
</journal>
```

Как правило, для обозначения пространства имен используются ссылки, образованные по спецификации URL, т. е. с каждым пространством имен связывают некоторый адрес. Механизм образования Web-адресов в силу своей природы должен обеспечить уникальность обозначения про-

пространства имен. Поскольку используется лишь алгоритм образования адреса, нет необходимости в выборе адреса реально существующего Web-сайта, и поэтому адрес можно составить совершенно произвольно. Атрибут **xmlns** используется как ключевое слово XML для обозначения объявления пространства имен. Пространству имен назначается префикс пространства имен — он указывается после атрибута **xmlns** и отделяется двоеточием, а затем после знака равенства записывается адрес, однозначно идентифицирующий пространство имен. В приведенном примере образовано два пространства имен с префиксами **x** и **y**. Префиксы используются в тегах, указывая, к какому пространству имен относится данный тег.

Пример 1.5.3. Пространства имен **html** и **journal**.

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- Пространства имен html и journal -->
<html xmlns:h="http://www.kcn.ru/three"
      xmlns:j="http://www.kcn.ru/four">
<h:head>
  <j:jrntitle>Lobachevsky&apos;Journal</j:jrntitle>
</h:head>
<h:body>
<j:jrntitle>Lobachevsky&apos;Journal</j:jrntitle>
<j:url>ljm.ksu.ru</j:url>
<j:articles>
<j:article ID="1">
<j:title>MathML and TeX</j:title>
</j:article>
<j:article ID="2">
<j:title>MathML and RDF</j:title>
</j:article>
</j:articles>
</h:body>
</html>
```

1.6. XSL-преобразование XML-документов

Как видно из приведенных примеров, XML совершенно не “заботится” о представлении информации, поскольку разработан для других целей. Для отображения информации в “привычном” виде, как это делалось с помощью языка HTML, необходимо использовать другие технологии.

Универсальным механизмом управления отображением XML–документов являются расширяемые стилевые таблицы XSL (eXtensible Stylesheet Language).

Стилевыми таблицами (стилевыми листами) принято называть специальные инструкции, управляющие процессом отображения элемента в окне программы-клиента (например, в окне браузера) и включаемые либо в тело самого документа, либо передаваемые по сети отдельным файлом.

XSL позволяет создавать шаблоны, в которых содержатся инструкции, управляющие преобразованием XML–документа. Отображение XML–документа в формате HTML — один из частных случаев использования XSL. Шаблон, создаваемый для этой цели, состоит из тегов двух типов, а именно, тегов HTML, используемых для форматирования, и тегов XSL, с помощью которых производится выборка данных из XML–документа. Шаблон должен преобразовать дерево XML–файла в другое дерево, которое будет соответствовать формату HTML. Шаблон сохраняется в файле с расширением **.xsl** и является документом XML, следовательно, при разработке шаблона должны учитываться требования, предъявляемые к правильно построенным документам XML.

Например, создадим шаблон преобразования для отображения файла **example2.xml**, приведенного в примере 1.3.2. Шаблон сохраним в XSL–файле.

Пример 1.6.1. XSL–файл (one.xsl) для XML–файла из примера 1.3.2.

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<h2>
  <xsl:value-of select="//jrntitle"/>
</h2>
<hr/>
<P>
<I><xsl:value-of select="//contacts/address"/></I>
</P>
<P>
<xsl:value-of select="//contacts/url"/>
</P>
</xsl:template>
</xsl:stylesheet>
```

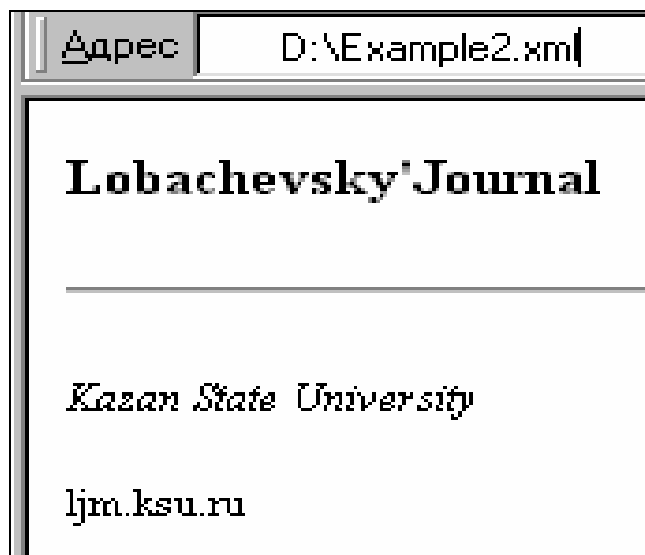
Для того, чтобы браузер выполнил необходимое преобразование, нужно в XML-файле указать ссылку на XSL-файл

```
<?xml-stylesheet type='text/xsl' href='one.xsl'?>
```

Пример 1.6.2. Листинг файла example2.xml. Добавлена ссылка на внешний XSL-шаблон.

```
<?xml version="1.0" encoding="windows-1251"?>
<?xml-stylesheet type='text/xsl' href='one.xsl'?>
<journal>
<jrntitle>Lobachevsky's Journal</jrntitle>
<contacts>
<address>Kazan State University</address>
<url>ljm.ksu.ru</url>
</contacts>
<articles>
<article ID="1">
<title>MathML and TeX</title>
<author>M. Malakhaltsev</author>
</article>
<article ID="2">
<title>MathML and RDF</title>
<author>E. Lipachev</author>
</article>
</articles>
</journal>
```

Если мы теперь откроем файл **example2.xml** в браузере Internet Explorer, то получим “привычную” форму представления документа:



Дадим несколько пояснений к приведенному примеру. Первая строка файла содержит тег элемента **xsl:stylesheet**. Атрибуты элемента — номер версии и ссылка на пространство имен (**xmlns:xsl**). Эти атрибуты элемента **xsl:stylesheet** являются обязательными. В нашем случае пространство имен — это все имена элементов и их атрибутов, которые могут использоваться в XSL-файле. Для XSL-файлов ссылка на пространство имен является стандартной.

Как было отмечено, XSL-файл является одной из разновидностей XML-файлов. Он не содержит пользовательских данных, но формат его тот же самый. Файл содержит элемент верхнего уровня **xsl:stylesheet**, а далее идет дерево правил преобразования.

Элемент **<xsl:template>** с помощью атрибутов **match** и **language** определяет узел XML-дерева, с которого будет производиться преобразование, и используемый для этого язык сценариев (по умолчанию JScript). Атрибут **match=""** указывает, что обработка начнется с корневого элемента. Элементов **<xsl:template>** в шаблоне может быть несколько, но они не должны быть вложенными друг в друга.

Элемент **<xsl:value-of>** позволяет получить значение из элемента XML-документа и отобразить в виде текста. С помощью атрибута **select** задается строка, служащая образцом для определения нужного элемента, например,

```
<xsl:value-of select="//contacts/address"/>
```

Ссылка на элемент может быть сокращенной, как в приведенном примере, или полной, с указанием всех ветвей дерева, содержащих элемент:

```
<xsl:value-of select="journal/contacts/address"/>
```

Элемент **<xsl:for-each>** **</for-each>** позволяет осуществить просмотр набора элементов XML-документа. Элементы отбираются в соответствии с атрибутом **select** и ко всем этим элементам применяется одинаковый набор преобразований. Например, с помощью шаблона

```
<xsl:for-each select="journal/articles/article">
```

```
...
```

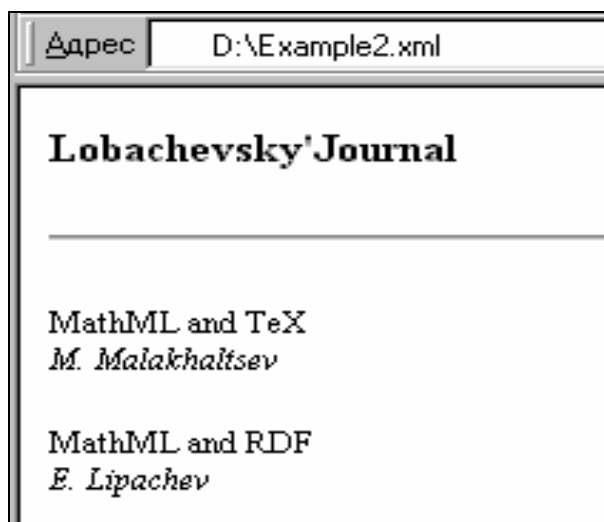
```
</xsl:for-each>
```

будут просмотрены все вложенные элементы ветви **journal/articles/article** (в нашем примере это элементы **title** и **author**).

Пример 1.6.3. Ещё одна версия файла one.xsl.

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<h2>
  <xsl:value-of select="journal/jrntitle"/>
</h2>
<hr/>
<xsl:for-each select="journal/articles/article">
<P>
<xsl:value-of select="title"/><br/>
<I><xsl:value-of select="author"/></I>
</P>
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

При запуске XML-файла example2.xml из примера 1.6.2 результат будет иной:



Если в файле one.xsl изменить тег `<xsl:for-each ...` на

```
<xsl:for-each select="journal/articles/article"
  order-by="author">
```

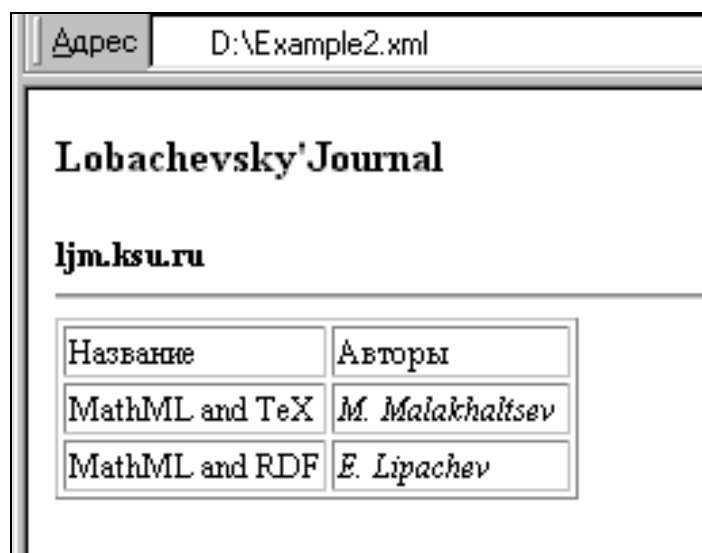
то перед выводом статьи будут упорядочены по авторам. Критерий сортировки задаётся атрибутом **order-by**, при этом символ “;” используется как разделитель, если критериев несколько.

Пример 1.6.4. Следующий стилевой файл используется для отображения XML-файла в виде таблицы. Обращаем внимание на русские названия

столбцов таблицы. Для этого в XSL-файле с помощью атрибута **encoding** указана кодировка символов.

```
<?xml version="1.0" encoding="windows-1251"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<h2>
  <xsl:value-of select="journal/jrntitle"/>
</h2>
<b><xsl:value-of select="journal/contacts/url"/> </b>
<hr/>
<table border="1">
<tr>
  <td> Название </td><td> Авторы </td>
</tr>
<xsl:for-each select="journal/articles/article">
<tr>
  <td><xsl:value-of select="title"/></td>
  <td><I><xsl:value-of select="author"/></I></td>
</tr>
</xsl:for-each>
</table>
</xsl:template>
</xsl:stylesheet>
```

Если этот файл сохранить как **one.xml**, то при запуске файла **example2.xml**, код которого приведен ранее, получим



Название	Авторы
MathML and TeX	<i>M. Malakhaltsev</i>
MathML and RDF	<i>E. Lipachev</i>

2. Разметка математических текстов по технологии MathML

2.1. Вводный пример

Допустим, что нам надо осуществить MathML–разметку для отображения следующего текста:

Quadratic equation

Корни квадратного уравнения $ax^2 + bx + c = 0$
имеют вид

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Код для отображения этого фрагмента можем подготовить в стандартном блокноте Windows. Откроем блокнот и наберем:

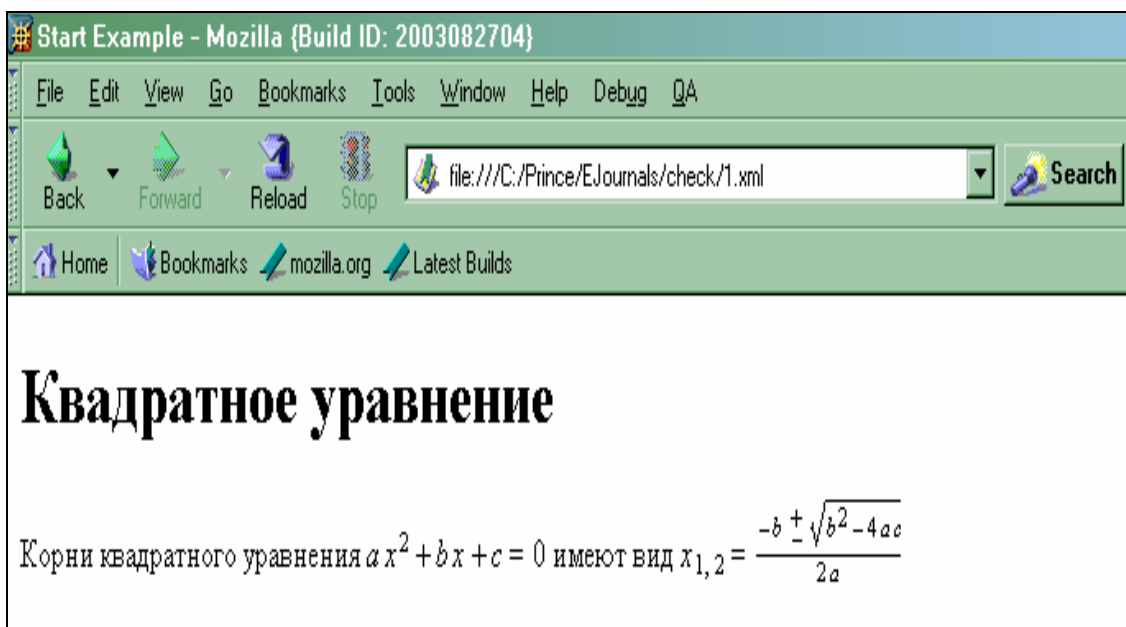
```
<?xml version="1.0" encoding="windows-1251"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
<title>Start Example</title>
</head>
  <body>
    <h1>Quadratic equation</h1>
    Корни квадратного уравнения
<math xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>a</mi>
      <msup>
        <mi>x</mi>
        <mn>2</mn>
      </msup>
      <mo>+</mo>
      <mi>b</mi>
      <mi>x</mi>
      <mo>+</mo>
      <mi>c</mi>
      <mo>=</mo>
      <mn>0</mn>
    </math>
    имеют вид
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <msub>
```

```

    <mi>x</mi>
    <mrow>
      <mn>1</mn>
      <mo>,</mo>
      <mn>2</mn>
    </mrow>
  </msub>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mo>-</mo>
      <mi>b</mi>
      <mfrac linethickness="0">
        <mo>+</mo>
        <mo>-</mo>
      </mfrac>
      <msqrt>
        <msup>
          <mi>b</mi>
          <mn>2</mn>
        </msup>
        <mo>-</mo>
        <mn>4</mn>
        <mi>a</mi>
        <mi>c</mi>
      </msqrt>
    </mrow>
    <mrow>
      <mn>2</mn>
      <mi>a</mi>
    </mrow>
  </mfrac>
</math>
</body>
</html>

```

Теперь сохраним файл под именем **example1.xml**. Откроем его в браузере Mozilla, мы должны увидеть картинку



Теперь попытаемся разобраться в тексте созданного нами файла. Первые две строки указывают, что данный документ представляет код XHTML. Напомним, что первая строка отмечает, что это XML–документ, а вторая определяет пространство имен XHTML как <http://www.w3.org/1999/xhtml>. Хотя эта строка выглядит как ссылка, она является лишь уникальным идентификатором пространства имен, который распознается браузером, и, следовательно, не требует соединения с Интернетом. Отметим, что пространство имен используется по умолчанию (т. к. нет префикса после **xmlns**), поэтому перед тегами не указан префикс, задающий пространство имен.

Для разметки текста используются стандартные теги HTML, а для представления формул — теги MathML. Блоки с кодом MathML заключаются в теги

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
.....
</math>
```

Открывающий тег **<math>** определяет ссылку на пространство имен MathML. В примере необходимо отобразить две формулы, поэтому он содержит два блока **$$**.

Выражения MathML строятся в виде дерева, каждый узел которого является элементом MathML (например, числом или операцией). Все элементы MathML разделены на три группы: *элементы представления* (presentation elements), *элементы содержания* (content elements) и *интерфейсные*

элементы (interface elements). Одно и то же выражение может быть размечено в виде “разметки представления” (presentation markup) и “разметки содержания” (content markup). В нашем примере используется разметка представления. *Так как разметка содержания в настоящий момент не поддерживается браузерами, мы ее не рассматриваем.* В рекомендации MathML [23] элементы представления называются также *токенами представления* (presentation token elements)

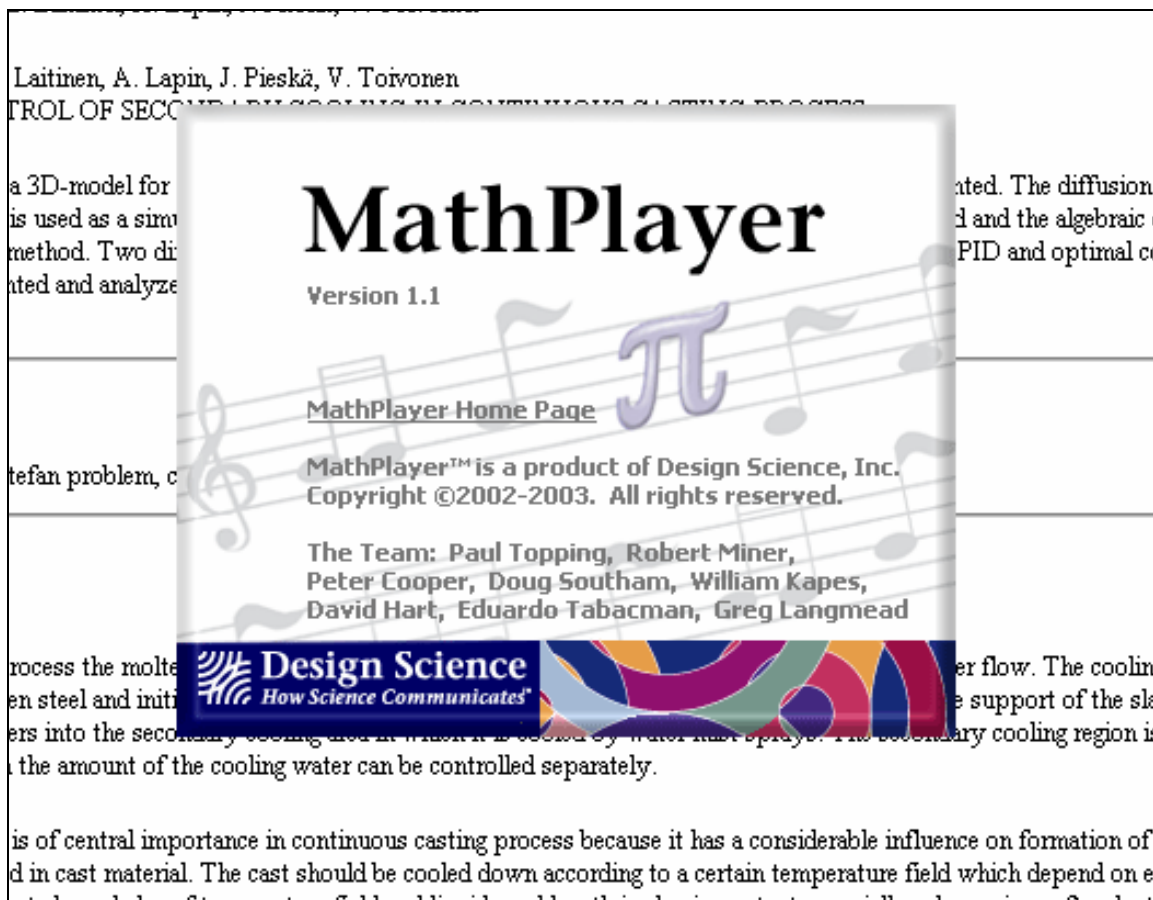
Элементы представления **mi**, **mn** и **mo** используются чаще всего — с их помощью производится разметка для идентификаторов (например, переменных), чисел и операций. В примере имеются элементы **msup** и **msub**, которые используются для отображения верхних и нижних индексов. Во второй формуле нижний индекс двойной, поэтому потребовалась группировка элементов, представляющих индекс. Для группировки использовали элемент **mrow**, который предназначен для обозначения горизонтального ряда частей выражения. Элемент **msqrt** отвечает за представление квадратного корня, а элемент **mfrac** — за представление дроби. Подробно теги MathML описаны ниже в разделе 3.

2.2. Особенности отображения в браузерах

В настоящее время (конец 2003 года) в основном используются браузеры Internet Explorer 6.0, Netscape 7.0, Mozilla 1.5b. Браузеры Mozilla и Netscape отображают MathML одинаково, поэтому мы объясним различие в подготовке MathML–разметки для Internet Explorer и Mozilla.

2.2.1. Особенности отображения в Internet Explorer

Для просмотра MathML в Internet Explorer необходимо “скачать” пакет MathPlayer (например, с сайта <http://www.dessci.com/en/>). Установка не вызовет затруднений. В каждом сеансе работы с Internet Explorer при открытии MathML–файла загружается MathPlayer, сообщая о себе выводом окна



Просмотр XML-файла

Прежде надо скачать с сети файл **pmathml.xml**. В начало файла помещаются строки

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="pmathml.xml"?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html" />
</head>
<body>
```

Здесь предполагается, что файл **pmathml.xml** размещен в одном каталоге с создаваемым xml-файлом, в противном случае требуется указать полный путь к нему. Каждый фрагмент MathML заключается в теги:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
</math>
```

В конце файла не забудьте поставить закрывающие теги

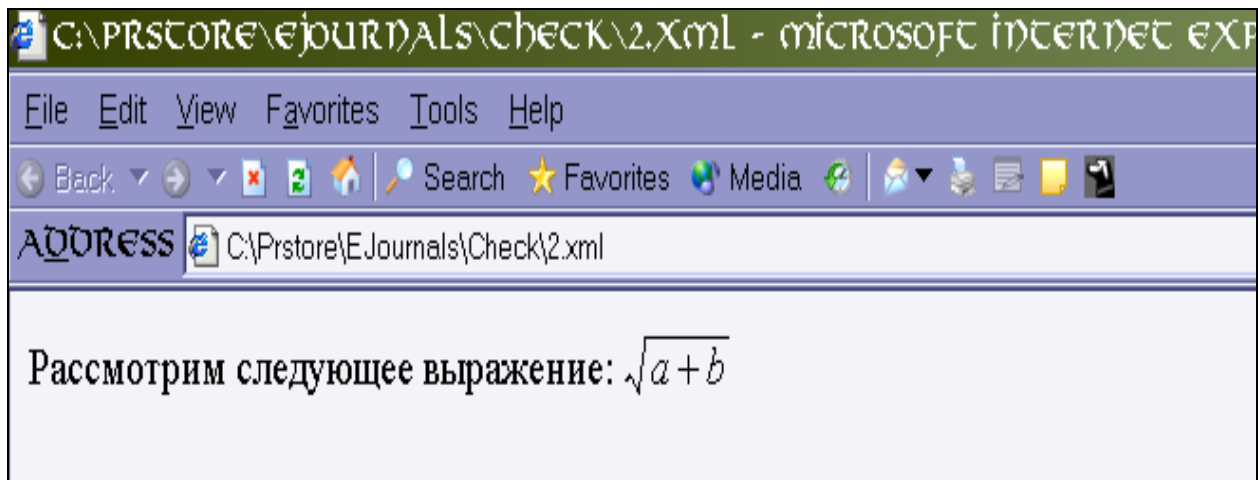
```
</body>
</html>
```

Пример XML-файла для браузера Internet Explorer

Наберите следующий текст в Блокноте.

```
<?xml version="1.0" encoding="windows-1251"?>
<?xml-stylesheet type="text/xsl" href="pmathml.xsl"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html" />
  </head>
  <body>
    <p> Рассмотрим следующее выражение:
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <msqrt>
        <mi>a</mi>
        <mo>+</mo>
        <mi>b</mi>
      </msqrt>
    </math>
  </p>
</body>
</html>
```

Сохраните этот файл как **2.xml**. Открыв этот файл в браузере Internet Explorer, вы должны увидеть следующую картинку:



Просмотр HTML-файла

Приведем простейший пример html-файла, содержащего MathML-код, использующий MathPlayer для отображения:

```

<HTML XMLNS:m="http://www.w3.org/1998/Math/MathML">
<head>
  <OBJECT ID=example1
  CLASSID="clsid:32F66A20-7614-11D4-BD11-00104BD3F987">
  </OBJECT>
  <?IMPORT NAMESPACE="M" IMPLEMENTATION="#example1">
</head>

<body>
<p> Рассмотрим следующее выражение
  <m:math>
    <m:msqrt>
      <m:mi>a</m:mi>
      <m:mo>+</m:mo>
      <m:mi>b</m:mi>
    </m:msqrt>
  </m:math>
</p>
</body>
</html>

```

MathPlayer подключается при помощи тега **OBJECT**. В нем указывается атрибут ID, значение которого можно изменять (это же значение используется в теге **IMPORT**).

В html-файл можно добавить скрипт, который проверяет версию браузера. Примерно такой же код получается при конвертации TeX-файлов в MathML программой TeX4ht (см. раздел 6.1).

Начало файла содержит следующие обязательные строки:

```

<html xmlns:m="http://www.w3.org/1998/Math/MathML"
  xmlns="http://www.w3.org/1999/xhtml" >

<head>
<object
  id="MathPlayer"
  classid="clsid:32F66A20-7614-11D4-BD11-00104BD3F987">
</object>
<?import namespace="m" implementation="#MathPlayer" ?>
</head>

```

В начале раздела **<body>** надо вызвать MathPlayer:

```

<body>

  <!--1. 31--><p class="noindent"><script language =
"javascript">
  <!--
  if( navigator.appName=="Microsoft Internet Explorer"
  && navigator.platform=="Win32"
  && parseFloat(navigator.appVersion.substr(
  navigator.appVersion.indexOf("MSIE ") +5))>="5.5"
  ){
  try {
  var oMP = new ActiveXObject("MathPlayer.Factory.1");
  }
  catch(e) { alert("Can't find Design Science's Math-
Palyer" +
  "(http://www.dessci.com/webmath/mathplayer)");}
  } else {
  alert("Requires MSIE version 5.5 or later");
  }

  -->
</script>
</p>

```

Синтаксис MathML–разметки также отличается от xml–варианта. Именно, каждый тег начинается с **<m:**, например:

```

<m:math>
<m:mi>T</m:mi>
<m:mo>=</m:mo>
<m:mi>A</m:mi>
</m:math>

```

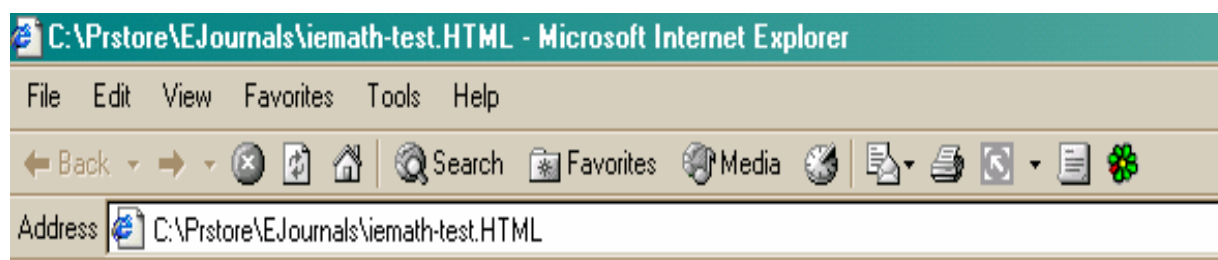
Таким образом, формально это выглядит как использование CSS (Cascade Style Sheets), однако фактически соответствующий CSS–лист встроен в MathPlayer. В частности, можно использовать атрибут **class**, например,

```

<m:math display="inline">
<m:mrow>
  <m:mo class="MathClass-open">[</m:mo>
    <m:mrow>
      <m:msub>
        <m:mrow><m:mi>W</m:mi></m:mrow>
        <m:mrow><m:mn>0</m:mn></m:mrow>
      </m:msub>
      <m:mo class="MathClass-punc">,</m:mo>
      <m:msub>
        <m:mrow><m:mi>L</m:mi></m:mrow>
        <m:mrow><m:mi>X</m:mi></m:mrow>
      </m:msub>
    </m:mrow>
  <m:mo class="MathClass-close">]</m:mo>
</m:mrow>
<m:mo class="MathClass-bin">⋄;</m:mo>
<m:mi> Y </m:mi>
<m:mo class="MathClass-rel">=</m:mo>
<m:mrow>
<m:mi> Y </m:mi>
</m:math>

```

При просмотре этот фрагмент выглядит так:



$$[W_0, L_X] \times Y = Y$$

Пример HTML-файла для браузера Internet Explorer

Наберите этот текст в Блокноте:

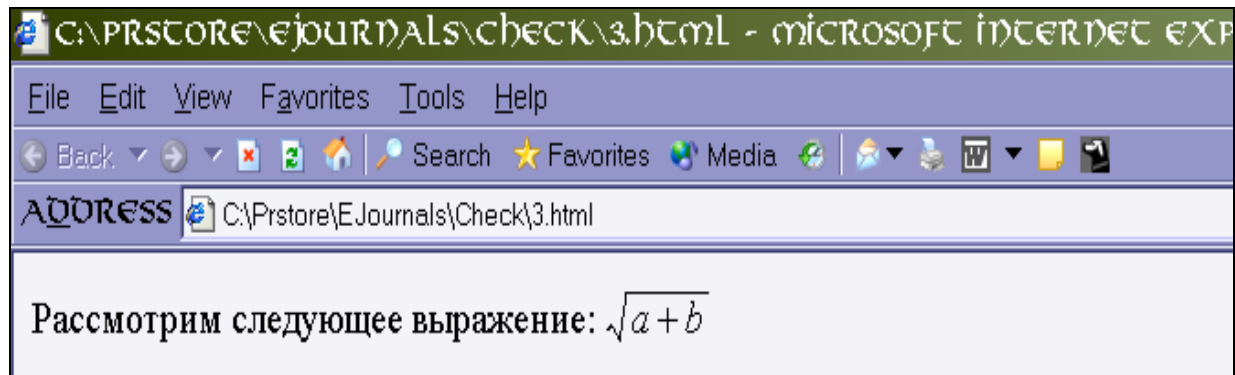
```

<html xmlns:m="http://www.w3.org/1998/Math/MathML"
xmlns="http://www.w3.org/1999/xhtml">
<head>
<object id="MathPlayer"
classid="clsid:32F66A20-7614-11D4-BD11-00104BD3F987">
</object>
<?import namespace="m" implementation="#MathPlayer" ?>
<!--http://www.dessci.com/webmath/mathplayer/-->
</head>
<body>
<!--1. 31--><p class="noindent">
<script language = "javascript">
<!--
if( navigator.appName=="Microsoft Internet Explorer"
&& navigator.platform=="Win32"
&& parseFloat(navigator.appVersion.substr(
navigator.appVersion.indexOf("MSIE")+5))>="5.5"
){
try {
var oMP = new ActiveXObject("MathPlayer.Factory.1");
}
catch(e) {
alert("Can't find Design Science's Math-Palyer" +
"(http://www.dessci.com/webmath/mathplayer)");}
}
else {
alert("Requires MSIE version 5.5 or later");
}
-->
</script>
</p>
<p>
Рассмотрим следующее выражение:
<m:math>
<m:msqrt> <m:mrow> <m:mi>a</m:mi>
<m:mo>+</m:mo>
<m:mi>b</m:mi>
<m:mrow>
</m:msqrt>
</m:math>
</p>
</body>

```

```
</html>
```

Сохраните файл как 3.html. Вы должны увидеть следующую картинку:



Пакет TechExplorer

Кроме пакета (точнее, плагина) MathPlayer, для отображения MathML-документов в браузере Internet Explorer можно использовать другой плагин. Он называется TechExplorer, информация о нем находится на сайте <http://www-306.ibm.com/software/network/techexplorer/>. Подробности использования программы TechExplorer можно найти в [9].

2.2.2. Особенности отображения в браузере Mozilla

Браузер Mozilla имеет встроенную поддержку MathML для формата XML. Файл, в котором имеются MathML-включения, надо начинать со строк

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml">
```

Каждый фрагмент MathML заключается в теги:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
.....
</math>
```

В конце файла ставятся закрывающие теги

```
</body>
</html>
```

Так, уже рассмотренный пример для браузера Internet Explorer будет для браузера Mozilla выглядеть следующим образом:

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
<p> Рассмотрим следующее выражение
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <msqrt> <mrow> <mi>a</mi>
  <mo>+</mo>
  <mi>b</mi>
  </mrow>
  </msqrt>
</math>
</p>
</body>
</html>
```

Открыв этот файл в браузере Mozilla, вы увидите такую же картинку, какая была в браузере Internet Explorer.

3. Синтаксис языка

Из примеров, приведенных в предыдущих разделах, мы видим, что разметка MathML очень громоздкая, поэтому вручную ввести реальный текст невозможно (впрочем, если читатель привык к TeX-набору, он без особого труда справится с MathML-набором; надо лишь использовать редактор с автозаменой). В дальнейшем мы описываем программы, которые автоматически генерируют MathML-код. Тем не менее, надо иметь представление о синтаксисе языка, так как часто приходится поправлять сгенерированный код вручную.

Если проанализировать уже приведенные примеры, то увидим, что каждый токен (элемент) формулы заключается в тег с атрибутами, указывающими его специализацию. Например, число заключается в тег `<mn> </mn>`, операция — в `<mo> </mo>`, символ — в `<mi> </mi>`. При этом одни теги могут быть вложены друг в друга. Таким образом, выражения MathML можно представить в виде деревьев, где каждый узел соответствует элементу MathML, ветвь под “родительским” узлом соответствует “дочерним” узлам и листья дерева соответствуют атомарным элементам нотации или содержания, таким, как числа, символы и т. д. Все элементы MathML делятся на три группы: элементы представления, элементы содержания и интер-

фейсные элементы. Элементы представления описывают визуально ориентированную двумерную структуру математической нотации. Типичным примером является элемент **mrow**, который обычно применяется для обозначения горизонтального ряда частей выражения, и элемент **msup**, который отмечает верхний индекс. Как правило, каждый элемент представления соответствует одному типу нотационной схемы, такой, как ряд, верхний индекс, нижний индекс и так далее. Любая формула состоит из частей, которые могут состоять из простейших элементов, таких, как цифры, буквы или другие символы.

Разметка MathML состоит из приблизительно 30 элементов, которые имеют более 50 атрибутов. Элементы представления делятся на два класса. *Токены (token elements)* представляют индивидуальные символы, названия, числа, обозначения и т. д. В основном в качестве содержания токены могут иметь только символы. Единственными исключениями являются элемент вертикального выравнивания `malignmark`, `mglyph` и ссылки на сущности (entity references). *Элементы схемы (layout schemata)* формируют выражения из частей и могут иметь только элементы в качестве содержания (исключая пробелы, которые они игнорируют). Также имеется несколько пустых элементов, используемых только вместе с определенным элементом схемы.

Все индивидуальные “символы” в математическом выражении должны быть представлены токенами MathML. Основные типы токенов MathML — идентификаторы (т. е. переменные или имена функций), числа и операторы (включая различные ограничители, такие, как круглые скобки, и разделители, такие, как запятые). Имеются также токены для представления текста или пробелов, что имеет больше эстетическое, чем математическое значение, и для представления “строчных констант” (string literals) для совместимости с системами компьютерной алгебры. Надо заметить, что хотя токен представляет одиночную смысловую единицу (имя, число, знак, математический символ и т. д.), такая единица может состоять более чем из одной литеры (character). Например, **sin** и **24** представлены одиночными токенами `<mi>sin</mi>` и `<mn>24</mn>` соответственно.

Далее мы приводим основные элементы MathML (более подробное изложение с перечислением всех тегов и многочисленных атрибутов можно найти по адресу <http://www.w3.org/TR/MathML2>).

3.1. Токены

Тег	Определение	Пример
mi	идентификатор (identifier)	<code><mi> a </mi></code>
Mn	число (number)	<code><mn> 1324 </mn></code>
Mo	оператор (operator), ограничитель (fence) или разделитель (separator)	<code><mo>+</mo></code>
mtext	текст (text)	<code><mtext></code> Это пример <code></mtext></code>
Ms	текстовая строка (string literal)	<code><ms> ННННННН</ms></code>

Применение основных токенов ясно из вышеприведенных примеров.

3.2. Основные элементы

Тег	Определение	Пример	Результат
Mrow	группирует любое количество подвыражений в строку	см. вводный пример	
mfrac	формирует дробь из двух подвыражений	<code><mfrac></code> <code><mi>a</mi></code> <code><mi>b</mi></code> <code></mfrac></code>	$\frac{a}{b}$
msqrt	формирует квадратный корень (радикал без индекса)	<code><math></code> <code><msqrt></code> <code><mi>a</mi></code> <code><mo>+</mo></code> <code><mi>b</mi></code> <code></msqrt></code> <code></math></code>	$\sqrt{a+b}$
mroot	формирует радикал с определенным индексом	<code><math></code> <code><mroot></code> <code><mrow></code> <code><mi>a</mi></code> <code><mo>+</mo></code> <code><mi>d</mi></code> <code></mrow></code> <code><mrow></code> <code><mi>b</mi></code> <code><mo>+</mo></code> <code><mi>c</mi></code> <code></mrow></code> <code></mroot></code> <code></math></code>	$b+\sqrt[3]{a+d}$

mpadded	регулирует отступы вокруг содержимого	<pre><math> <mpadded width="+5em"> <mi>a</mi> <mo>+</mo> <mi>b</mi> </mpadded> <mo>-</mo> <mi>c</mi> </math></pre>	$a + b$
mphantom	делает содержимое невидимым, но сохраняет его размер	<pre><math> <mi> x </mi> <mphantom> <mo> + </mo> <mi> y </mi> </mphantom> <mo> + </mo> <mi> z </mi> </math></pre>	$x + z$
mfenced	окружает содержимое парой скобок	<pre><math> <mfenced> <mi>x</mi> </mfenced> </math></pre>	(x)

3.3. Индексы

Тег	Определение	Пример	Результат
Msub	добавляет нижний индекс к основанию	<pre><math> <msub> <mi>x</mi> <mn>1</mn> </msub> </math></pre>	x_1
Msup	добавляет верхний индекс к основанию	<pre><math> <msup> <mi>x</mi> <mn>12</mn> </msup> </math></pre>	x^{12}
msubsup	добавляет верхний и нижний индексы к основанию	<pre><math> <msubsup> <mi>x</mi> <mn>z</mn> <mn>12</mn> </msubsup> </math></pre>	x_z^{12}


```

        <mi>b</mi>
    </mrow>
    <mo class="MathClass-close">)</mo>
</mrow>
</mrow>
<mrow>
    <mn>2</mn>
</mrow>
    </msup>
    <mo class="MathClass-rel">=</mo>
    <msup>
    <mrow>
        <mi>a</mi>
    </mrow>
    <mrow>
        <mn>2</mn>
    </mrow>
    </msup>
    <mo class="MathClass-bin">+</mo>
    <mn>2</mn><mi>a</mi><mi>b</mi>
    <mo class="MathClass-bin">+</mo>
    <msup>
    <mrow>
        <mi>c</mi>
    </mrow>
    <mrow><mn>2</mn>
    </mrow>
    </msup>
</math>

```

Результат:

$$(a+b)^2 = a^2 + 2ab + c^2$$

4.2. Основное тригонометрическое тождество

Код MathML:

```

<math display="inline">
    <mo class="MathClass-op">sin</mo>
    <msup>
    <mrow>
    <mrow>
        <mo class="MathClass-open">( </mo>
        <mrow>
            <mi>x</mi>
        </mrow>
    </mrow>
    </msup>

```

```

    <mo class="MathClass-close">)</mo>
  </mrow>
</mrow>
  </mrow>
  <mrow>
    <mn>2</mn>
  </mrow>
</msup>
<mo class="MathClass-bin">+</mo>
<mo class="MathClass-op"> cos</mo>
<msup>
  <mrow>
    <mrow>
      <mo class="MathClass-open">( </mo>
      <mrow>
        <mi>x</mi>
      </mrow>
      <mo class="MathClass-close">)</mo>
    </mrow>
  </mrow>
  <mrow>
    <mn>2</mn>
  </mrow>
</msup>
<mo class="MathClass-rel">=</mo>
<mn>1</mn>
</math>

```

Результат:

$$\sin(x)^2 + \cos(x)^2 = 1$$

4.3. Формула с греческими буквами и символом частного дифференцирования

Символы в MathML-коде задаются, как правило, с помощью кодов. Коды приведены в [28].

Код MathML:

```

<math display="inline">
  <msub>
    <mrow><mi>&#x2202;</mi></mrow>
    <mrow><mi>i</mi></mrow>
  </msub>
  <mrow>
    <mo class="MathClass-open">[ </mo>

```

```

<mrow>
  <mover accent="true">
    <mrow> <mi>#x03B1;</mi> </mrow>
    <mo class="MathClass-op">#x2192;</mo>
  </mover>
  <mo class="MathClass-punc">,</mo>
  <mover accent="true">
    <mrow><mi>#x03B2;</mi></mrow>
    <mo class="MathClass-op">#x2192;</mo>
  </mover>
</mrow>
<mo class="MathClass-close">]</mo>
</mrow>
<mo class="MathClass-rel">=</mo>
<mover accent="true">
  <mrow><mi>#x03B3;</mi></mrow>
  <mo class="MathClass-op">#x2192;</mo>
</mover>
</math>

```

Результат:

$$\partial_i[\vec{\alpha}, \vec{\beta}] = \vec{\gamma}$$

4.4. Матрица

При написании MathML-кода матрицы используются следующие теги:

Тег	Определение
mtable	таблица или матрица
mtr	строка в таблице или матрице
mtd	одна ячейка в таблице или матрице

Код MathML:

```

<math display="block">
  <mfenced separators="" open="(" close=")" >
  <mrow>
    <mtable>
      <mtr>
        <mtd> <mi>A</mi></mtd>
        <mtd> <mi>B</mi> </mtd>
      </mtr>
      <mtr>
        <mtd><mi>C</mi></mtd>
        <mtd><mi>D</mi></mtd>
      </mtr>
    </mtable>
  </mrow>
</math>

```

```
</mtable>
</mrow>
</mfenced>
</math>
```

Результат:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

4.5. Коммутативная диаграмма

Используя матрицы, можно построить коммутативные диаграммы. Отметим, что они лучше всего воспроизводятся с помощью MathPlayer.

MathML-код:

```
<math display="block">
  <mtable >
    <mtr>
      <mtd>
        <mi>A</mi>
      </mtd>
      <mtd>
        <mover>
          <mrow>
            <mo class="MathClass-rel">&#x2192; </mo>
          </mrow>
          <mrow><mi>f</mi></mrow></mover>
        </mtd>
      <mtd>
        <mi>B</mi>
      </mtd>
    </mtr>
    <mtr>
      <mtd>
        <mi>&#x2193;</mi>
      </mtd>
      <mtd>
        </mtd>
      <mtd>
        <mi>&#x2191;</mi>
      </mtd>
    </mtr>
  </mtable>
```

```

    <mtd>
      <mi>C</mi>
    </mtd>
    <mtd>
      <munder>
        <mrow>
          <mo class="MathClass-rel"> &#x2192; </mo>
        </mrow>
        <mrow><mi>g</mi></mrow>
      </munder>
    </mtd>
    <mtd>
      <mi>D</mi>
    </mtd>
  </mtr>
</mtable>
</math>

```

Результат:

$$\begin{array}{ccc}
 A & \xrightarrow{f} & B \\
 \downarrow & & \uparrow \\
 C & \xrightarrow{g} & D
 \end{array}$$

5. Использование пакета Mathematica для подготовки Web-публикаций

Компания Wolfram Research в последние годы активизировала работу по поддержке технологии MathML в своем широко известном пакете Mathematica. Развитие пакета идет не только в сторону увеличения вычислительной мощности, но и как удобного инструмента для подготовки всего цикла Web-публикаций, включая набор и конвертацию в нужные форматы [25, 26]. В разных версиях пакета Mathematica конвертация в MathML происходит по-разному.

5.1. Конвертация в пакете Mathematica 4.2

Есть два способа использования Mathematica для подготовки MathML-кода: сохранить файл

- с помощью команды **Save as Special->HTML+MathML;**

- с помощью команды MathMLForm, примененной к некоторому выражению; при этом Mathematica выдает MathML-код результата вычисления этого выражения.

Например, команда

MathML[Integrate[Sin[x],x]]

выдает следующий ответ

```
<math>
<mrow>
  <mo>-</mo>
  <mrow>
    <mi>cos</mi>
    <mo>&af;</mo>
    <mrow>
      <mo>( </mo>
      <mi>x</mi>
      <mo>)</mo>
    </mrow>
  </mrow>
</math>
```

К сожалению, в обоих случаях MathML-код не воспринимается браузерами без дополнительных изменений. Чтобы просмотреть файл, надо сделать следующее:

- изменить расширение файла на .xml;
- в начале файла убрать строки

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
```

```
<!-- This HTML document was created by converting a Mathematica
notebook using HTMLSave. -->
```

- поставить первой строку

```
<?xml version="1.0"?>
```

- к открывающему тегу <html> добавить

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

- каждый открывающий тег <math> заменить на

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
```

или, если формула выключенная, заменить на

```
<math mode="display" xmlns="http://www.w3.org/1998/Math/MathML">
```

Дальнейшие изменения зависят от версии браузера.

Корректировка файла для браузера Mozilla

Для того, чтобы браузер Mozilla воспринял MathML–код, подготовленный с помощью Mathematica, необходимо дополнительно сделать следующее:

- второй строкой поставить

```
<!DOCTYPE math SYSTEM
```

```
http://www.w3.org/Math/DTD/mathml1/mathml.dtd">
```

- посмотреть файл в Mozilla и исправить ошибки XML–кода; далее убрать из кода команды, не воспринимаемые Mozilla (они видны при просмотре в виде знаков вопроса).

Корректировка файла для браузера Internet Explorer

Вставить, начиная со второй строки:

```
<?xml-stylesheet type="text/xsl" href="pmathml.xsl"?>
```

```
<!DOCTYPE math
```

```
PUBLIC "-//W3C//DTD MathML 2.0//EN"
```

```
"mathml.dtd">
```

5.2. Конвертация в пакете Mathematica 5.0

Эта версия генерирует XML–код лучше. Если использовать команды меню

File->Save as Special->XML(XHTML+MathML),

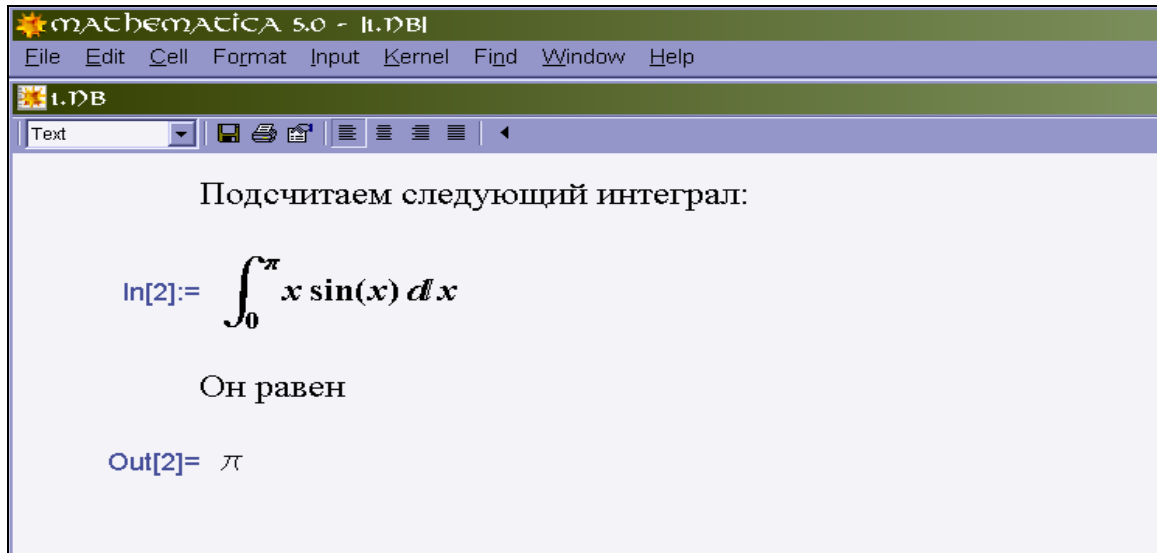
то получившийся файл просматривается Internet Explorer.

Одну формулу можно конвертировать в MathML–файл с помощью команды

Export["filename", требуемая формула, "MathML"]

из полученного файла с именем "filename" эту формулу можно скопировать в xml–файл.

Другие возможности, в частности, документ NotebookML, описаны в документации к этой версии программы Mathematica. Например, следующий фрагмент файла 1.nb

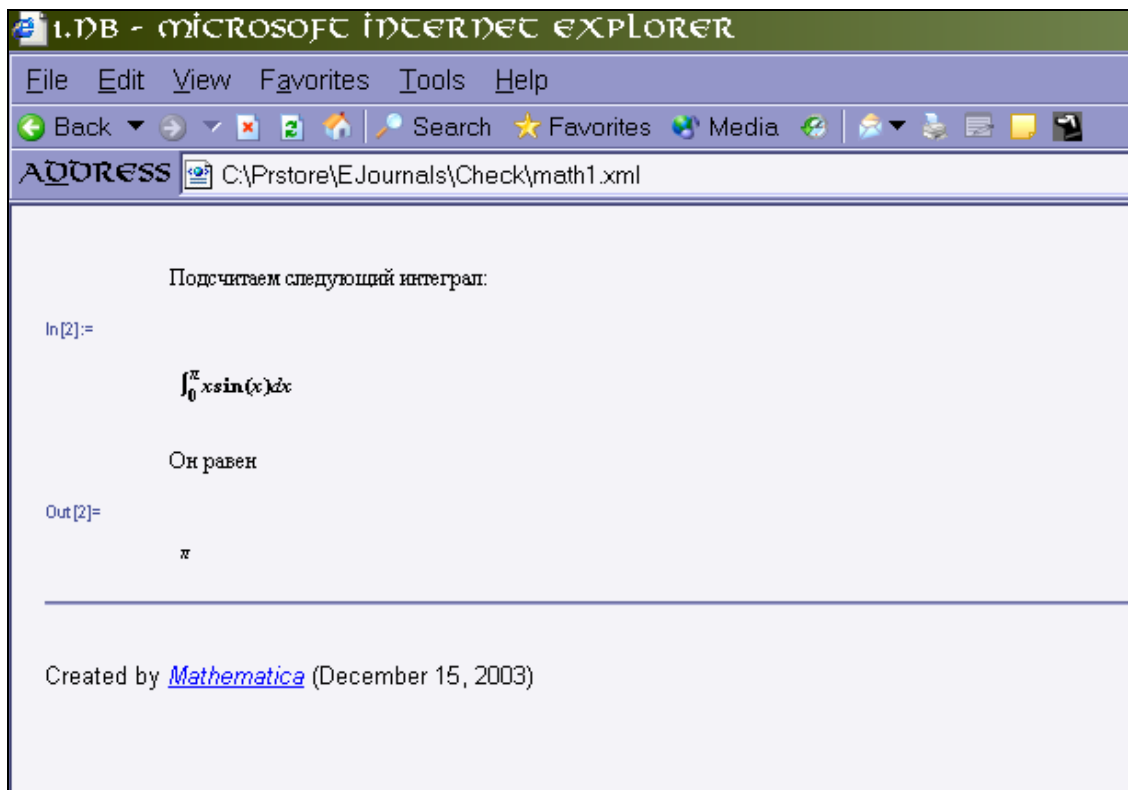


конвертируется пакетом Mathematica в xml-файл:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="HTMLFiles/pmathml.xsl"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN"
"HTMLFiles/xhtml1-transitional.dtd">

<!-- This HTML document was created by converting a
Mathematica notebook using Export. -->
<html xmlns="http://www.w3.org/1999/xhtml" class="Notebook">
<head >
<title>
  1.nb
</title>
<link href="HTMLFiles/Default.css" rel="stylesheet"
type="text/css" />
</head>
<body>
<p class="Text">
  &#1055;&#1086;&#1076;&#1089;&#1095;&#1080;&#1090;&#1072;&#10
77;&#1084;
  &#1089;&#1083;&#1077;&#1076;&#1091;&#1102;&#1097;&#1080;&#10
81;
  &#1080;&#1085;&#1090;&#1077;&#1075;&#1088;&#1072;&#1083;:
</p>
<p class="CellLabel">
  In[2]:=
</p>
<p class="Input">
  <math xmlns='http://www.w3.org/1998/Math/MathML'
```


Отметим, что русский текст конвертируется в Unicode. В Internet Explorer этот файл выглядит так:



6. Конвертация из TeX в MML

Для конвертации TeX–файла в MML–файл можно использовать пакет tex4ht (автор Eitan M. Gurari) или пакет MathZilla (автор Paul Gartside). Их можно скачать по адресам:

<http://www.cis.ohio-state.edu/~gurari>, <http://pear.math.pitt.edu/mathzilla>

Мы опишем работу с пакетом tex4ht для системы MikTeX (работа с MathZilla аналогична). Для упрощения обозначений будем считать, что все программы установлены в стандартных каталогах на диске C:.

6.1. Установка пакета TeX4ht

- Скачать файл **tex4ht.zip** с указанного web–адреса;
- Создать каталог **Tex4ht**;
- Распаковать файл **tex4ht.zip** в каталог **Tex4ht**;
- Все файлы из каталога **tex4ht\texmf\tex\generic\tex4ht** переместить в каталог **C:\tex4ht\bin\win32**;
- В файле
C:\tex4ht\texmf\tex4ht\base\win32\tex4ht.env

строку `tc:\path\tfm!` заменить на строку `tc:\texmf\fonts\tfm!`;

- Проверьте: если каталог `C:\tex4ht\texmf\tex4ht\htfonts` содержит подкаталог `htfonts`, то содержимое этого подкаталога надо переместить в `C:\tex4ht\texmf\tex4ht\`.

6.2. Работа с пакетом TeX4ht

Исходный TeX-файл нужно переписать в каталог `C:\tex4ht\bin\win32`. Используя командную строку или любой навигатор файлов (например, Windows Commander), выполнить команды из следующей таблицы, в зависимости от требуемого типа MML-файла.

Команда	Результат
<code>Mzlatex filename</code>	xml-файл для браузера Mozilla
<code>Mzlatex filename "html,mathplayer"</code>	html-файл для Internet Explorer с установленной программой MathPlayer
<code>xhtmlatex filename</code>	html-файл для Internet Explorer (работает не очень хорошо)

Например, следующий математический текст

Это пример математического текста.

Подсчитаем следующий интеграл:

$$I = \int_1^0 \frac{2v}{1+v^2} dv$$

Для этого сделаем замену $u = v^2$, тогда получим

$$I = \int_1^0 \frac{1}{1+u} du = \log(1+u)|_0^1 = \log 2$$

в нотации LaTeX2e может быть записан так:

```

\documentclass[12pt]{amsart}
\usepackage[cp866]{inputenc}
\usepackage[russian]{babel}

\begin{document}
\textbf{Это пример математического текста.}
\par
Подсчитаем следующий интеграл:
$$
I=\int\limits^0_1\frac{2v}{1+v^2}dv
$$
Для этого делаем замену  $u=v^2$ , тогда получим
$$
I=\int\limits^0_1\frac{1}{1+u}du = \log(1+u) \Big|_0^1 = \log 2
$$
\end{document}

```

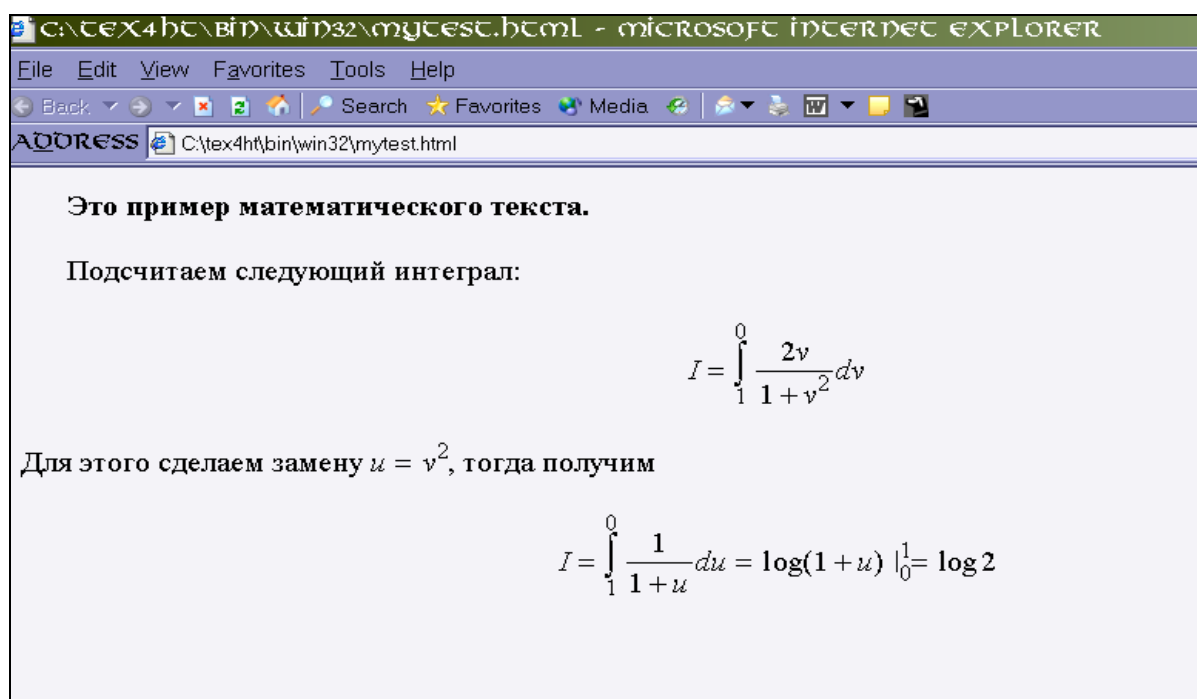
Наберем этот текст в любом редакторе и сохраним под именем **mytest.tex**.
Теперь скопируем этот файл в каталог

C:\tex4ht\bin\win32

и выполним команду (например, в Windows Commander)

mzlatex ex.tex "html,mathplayer"

В результате получим MathML–файл mytest.html, который в браузере Internet Explorer отображается следующим образом:



Рекомендация: на время отладки добавьте в конец используемых bat-файлов команду pause. Это позволит проследить процесс выполнения команд bat-файла.

7. Метаданные в XML-документах

В Web-страницах метаданные используются для обозначения ключевой информации о содержимом данной страницы. Роботы поисковых машин осуществляют выборку, в основном, по информации, представленной в блоке метаданных. Поэтому включение метаданных в HTML-файл позволяет определить параметры поиска на этапе создания. Известной рекомендацией по описанию основных характеристик информационного ресурса является Dublin Core Metadata Set (DC) [31].

Пример 7.1. Метаданные используются для описания статьи электронного журнала

```
<HTML XMLNS:M="http://www.w3.org/1998/Math/MathML">
<head>
  <OBJECT ID=article1
    CLASSID="clsid:32F66A20-7614-11D4-BD11-
00104BD3F987">
    </OBJECT>
    <?IMPORT NAMESPACE="M" IMPLEMENTATION="#article1">
    <!-- Блок метаданных DC: -->
    <META content="Splitting iterative methods and parallel
solution of variational inequalities" name=DC.Title>
    <META content="E.Laitinen" name=DC.Author.PersonalName>
    <META content="A.Lapin" name=DC.Author.PersonalName.2>
    <META content="J.Pieska" name=DC.Author.PersonalName.3>
    <META content = 'Splitting iterative methods for the
sum of maximal monotone and single-valued monotone opera-
tors in a finite-dimensional space are studied: conver-
gence, rate of convergence and optimal iterative parame-
ters are derived.' name=DC.Description>
    <META content="mathematics, applied mathematics, sci-
entific computing" name=DC.Subject>
    <META content="Lobachevskii Journal of Mathematics"
name=DC.Publisher>
    <!--Конец блока метаданных DC -->
  </head>
</body>
```

```

<H2 align=center>E. Laitinen, A.V. Lapin, J.Pieska</H2>
<H2 align=center>Splitting iterative methods and paral-
lel solution of
variational inequalities
</H2>
<H2 align=center>Lobachevskii Journal of Mathematics,
</H2>
<H2 align=center>Vol.8, pp.167-184 </H2>
<P>
Splitting iterative methods for the sum of maximal
monotone and .....
</P>
<!-- MathML Content Markup -->
<m:math>
<m:mi>a</m:mi>
<m:msup>
<m:mi>x</m:mi>
<m:mn>2</m:mn>
</m:msup>
<m:mo>+</m:mo>
<m:mi>b</m:mi>
<m:mi>x</m:mi>
<m:mo>+</m:mo>
<m:mi>c</m:mi>
<m:mo>=</m:mo>
<m:mn>0</m:mn>
</m:math>
<p>
...
</p>
</body>
</html>

```

Включить метаданные в XML-файл подобным образом не удастся.

Консорциум W3C разработал рекомендацию Resource Description Framework (RDF), предназначенную для стандартизации метаданных, описывающих Web-ресурсы [30]. Это одна из ключевых технологий Semantic Web [2, 3]. Метаданные из предыдущего примера запишем в RDF-файл. В этом случае для электронной публикации создаются два файла: xml-файл, содержащий разметку, и rdf-файл, содержащий метаданные.

Пример 7.2.

```
<?xml version="1.0" encoding="WINDOWS-1251"?>
<!-- article.xml -->
<article>
  <autor>E. Laitinen, A.V. Lapin, J.Pieska</autor>
  <title>Splitting iterative methods and parallel so-
lution of
    variational inequalities
  </title>
  <bibl>Lobachevskii Journal of Mathematics,
    Vol.8, pp.167-184)
  </bibl>
  <abstract>
    Splitting iterative methods for the sum of maxi-
mal monotone and .....
  <math>
    <mi>a</mi>
    <msup>
      <mi>x</mi>
      <mn>2</mn>
    </msup>
    <mo>+</mo>
    <mi>b</mi>
    <mi>x</mi>
    <mo>+</mo>
    <mi>c</mi>
    <mo>=</mo>
    <mn>0</mn>
  </math>
  </abstract>
</article>
```

```
<?xml version="1.0"?>
<!-- article.rdf -->
<rdf:RDF xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/metadata/dublin_core#">
<rdf:Description
rdf:about="http://ljm.ksu.ru/vol8/latin.htm">
  <dc:Publisher>
    Lobachevskii Journal of Mathematics
  </dc:Publisher>
  <dc>Title>
Splitting iterative methods and parallel
```

```

solution of variational inequalities
</dc:Title>
  <dc:Author>
    <rdf:Seq ID="AutorsAlphabeticalBySurname"
      rdf:_1="E.Laitinen"
      rdf:_2="A.V. Lapin"
      rdf:_3="J.Pieska"/>
  </dc:Author>
  <dc:Subject>
    mathematics, applied mathematics,
    scientific computing
  </dc:Subject>
</rdf:Description>
</rdf:RDF>

```

Можно использовать RDF и XML-описания в одном файле.

Пример 7.3.

```

<?xml version="1.0" encoding="windows-1251"?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
</head>
<body>
<!-- RDF описания: -->
<rdf:RDF xmlns:rdf=
  "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/metadata/dublin_core#">
<rdf:Description rdf:about=
  "http://ljm.ksu.ru/vol8/latin.htm">
  <dc:Publisher>
    Lobachevskii Journal of Mathematics
  </dc:Publisher>
  <dc:Title>
Splitting iterative methods and parallel
solution of variational inequalities
</dc:Title>
  <dc:Author>
    <rdf:Seq ID="AutorsAlphabeticalBySurname"
      rdf:_1="E.Laitinen"
      rdf:_2="A.V. Lapin"
      rdf:_3="J.Pieska"/>
  </dc:Author>
  <dc:Subject>mathematics, applied mathematics, sci-
entific computing</dc:Subject>
  </rdf:Description>
</rdf:RDF>

```

```

    <p> Splitting iterative methods for the sum of maximal
monotone and .....
    <!-- MathML вставка -->
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>a</mi>
      <msup>
        <mi>x</mi>
        <mn>2</mn>
      </msup>
      <mo>+</mo>
      <mi>b</mi>
      <mi>x</mi>
      <mo>+</mo>
      <mi>c</mi>
      <mo>=</mo>
      <mn>0</mn>
    </math>
  </p>
</body>
</html>

```

Полезные ссылки

1. *Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation*. <http://www.w3.org/TR/2000/REC-xml-20001006> (русский перевод: <http://www.rol.ru/news/it/helpdesk/xml01.htm>).
2. Tim Berners-Lee. *Semantic Web Road map*. – <http://www.w3.org/DesignIssues/Semantic.html>.
3. *Semantic Web: роли XML и RDF* // Открытые системы. – 2001. – № 9 (<http://osp.admin.tomsk.ru/os/2001/09/041.htm>).
4. Питц-Моултис Н., Кирк Ч. *XML* – СПб: БХВ – Петербург, 2000. – 736 с.
5. Хабибуллин И. Ш. *Самоучитель XML*. – СПб: БХВ – Петербург, 2003.
6. Tim Bray, Dave Hollander, Andrew Layman (Editors). *Namespaces in XML (W3C Recommendation)* – <http://www.w3.org/TR/REC-xml-names/>.
7. Pankaj Kamthan. *XML Namespaces: Universal Identification in XML Markup*. – <http://tech.irt.org/articles/js193/>.
8. *XSL Transformations (XSLT) Version 1.0 (W3C Recommendation)* / <http://www.w3.org/TR/1999/REC-xslt-19991116> (русский перевод <http://www.rol.ru/news/it/helpdesk/xslt01.htm>).

9. Гуссенс М., Ратц С. *Путеводитель по пакету LaTeX и его Web-приложениям.* – М.: Мир, 2001.
10. Александр Печерский. *Язык XML – практическое введение.*
www.citforum.kcn.ru/internet/xml/.
11. *An XHTML + MathML + SVG Profile.* – [http://www.w3.org/TR / 2002/ WD-XHTMLplusMathMLplusSVG-20020430/xhtml-math-svg.html](http://www.w3.org/TR/2002/WD-XHTMLplusMathMLplusSVG-20020430/xhtml-math-svg.html).
12. David Carlisle. *MathML Files: DSSSL style sheet for MathML.* –<http://www.nag.co.uk/projects/OpenMath/mml-files/>.
13. *Gentle Introduction to MathML.* – <http://www.dessci.com/en/support/tutorials/mathml/default.htm>.
14. *History of MathML. The Two Threads of Mathematical Computer Languages.*
– <http://www.mathmlcentral.com/history.html>.
15. *MathML – What's in it for us.* – <http://tech.irt.org/articles/js081/>.
16. *MathPlayer. Display MathML in your browser.* – <http://www.dessci.com/en/products/mathplayer/>.
17. *OpenMath and MathML.* – <http://www.openmath.org/cocoon/openmath/projects/esprit/final/node6.htm>.
18. *OpenMath and MathML: Semantic Mark Up for Mathematics.* – <http://www.acm.org/crossroads/xrds6-2/openmath.html>.
19. *OpenMath: Accessing and Using Mathematical Information Electronically.* – <http://www.nag.co.uk/projects/OpenMath/>.
20. *Putting mathematics on the Web with MathML.* – <http://www.w3.org/Math/XSL/>.
21. *Strategies for Math on the Web.* – <http://www.dessci.com/en/reference/web-math/strategies.htm>.
22. *MathML for Math and Science Communication.* – <http://www.dessci.com/en/reference/webmath/tech/mathml.htm>.
23. *Math Typesetting for the Internet.* – <http://mathforum.org/typesetting/index.html>.
24. *TtM, a TeX to MathML translator.* – <http://hutchinson.belmont.ma.us/tth/mml/>.
25. *Wolfram Research Contributes Central Ideas to Web Math Standard.* – <http://www.wolfram.com/news/archive/mathml.html>.
26. Stephen Wolfram. *Mathematical Notation: Past and Future.* –<http://www.stephenwolfram.com/publications/talks/mathml/>.

27. Michael Kohlhase *MathML Presenting and Capturing Mathematics for the Web*. – <http://www.w3.org/Math/Documents/mathml-tutorial.pdf>.
28. *W3C MathML 2.0 Specification*. – <http://www.w3.org/Math> (частичный русский перевод – на <http://www.raleigh.ru/MathML/MathML2/>).
29. *W3C MathML 1.0 Specification*. – <http://www.w3.org/TR/REC-MathML-19980407>.
30. *Resource Description Framework (RDF) / W3C Semantic Web Activity*. – <http://www.w3.org/RDF/>.
31. *Dublin Core Metadata Element Set*. – http://purl.org/metadata/dublin_core_elements.

А. М. Елизаров, Е. К. Липачёв, М. А. Малахальцев

ОСНОВЫ MathML.
Представление математических текстов в Internet

Практическое руководство

Изд. лиц. № 0243 от 20.01.99. Подписано в печать 16.11.03

Бумага офсетная. Формат 60х90 1/16. Гарнитура «Таймс».

Тираж 100 экз., объем – 57 с.

Заказ 162. Печать ризографическая

Издательство Казанского математического общества
420008, Казань, Университетская, 17
