

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

Е. А. РУЖИЦКАЯ, А. В. ЛУБОЧКИН

WEB-ТЕХНОЛОГИИ: ИСПОЛЬЗОВАНИЕ DTD ДЛЯ ОПИСАНИЯ СТРУКТУРЫ XML-ДОКУМЕНТА

Практическое пособие

для студентов специальностей
6-05-0612-01 «Программная инженерия»,
6-05-0612-02 «Информатика и технологии программирования»

Гомель
ГГУ им. Ф. Скорины
2025

УДК 004.774.6(076)
ББК 32.973.434я73
Р837

Рецензенты:

доктор технических наук В. С. Смородин,
кандидат физико-математических наук Т. И. Васильева

Рекомендовано к изданию научно-методическим советом
учреждения образования «Гомельский государственный
университет имени Франциска Скорины»

Ружицкая, Е. А.

Р837 Web-технологии: использование DTD для описания
структуры XML-документа / Е. А. Ружицкая, А. В. Лубочкин ;
Гомельский гос. ун-т им. Ф. Скорины. – Гомель : ГГУ им.
Ф. Скорины, 2025. – 42 с.
ISBN 978-985-32-0132-1

Практическое пособие предназначено для оказания помощи студен-
там при выполнении заданий по дисциплинам «Web-технологии» и «Раз-
работка пользовательских интерфейсов» на 3 курсе. В нем излагается
теоретический материал и дано практическое задание.

Адресовано студентам факультета математики и технологий про-
граммирования 3 курса специальностей 6-05-0612-01 «Программная ин-
женерия», 6-05-0612-02 «Информатика и технологии программирования».

УДК 004.774.6(076)
ББК 32.973.434я73

ISBN 978-985-32-0132-1

© Ружицкая Е. А., Лубочкин А. В., 2025
© Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины», 2025

ОГЛАВЛЕНИЕ

Введение.....	4
Тема 1. Определение типа документа (DTD).....	5
1.1. Принципы создания допустимых документов.....	5
1.2. Внутреннее и внешнее DTD.....	5
Тема 2. Базовая разметка DTD.....	9
2.1. Декларация типа элемента (ELEMENT).....	9
2.2. Декларация атрибутов (ATTLIST).....	14
2.3. Условные разделы.....	25
Тема 3. Пример использования DTD.....	26
Практическое задание.....	41
Литература.....	42

ВВЕДЕНИЕ

XML – это технология описания и структурирования данных [1]. Правильно оформленный XML-документ – это документ, который подчиняется определенным грамматическим законам, прописанным в стандарте 1.0.

XML-документы могут быть двух типов:

- корректные (well-formed), удовлетворяющие определенному набору требований;
- допустимые (valid), т. е. соответствующие всем требованиям, предъявляемым к корректным XML-документам, и удовлетворяющие условиям подключенной к ним схемы.

В настоящее время World Wide Web Consortium (W3C) поддерживает две схемы: DTD (Document Type Definition) и XML Schema. В практическом пособии рассмотрена первая, более ранняя из этих двух схем – DTD.

Задача DTD определить [3]:

- состав элементов, которые могут использоваться в XML-документе;
- описание моделей содержания, т. е. правил вхождения одних элементов в другие;
- состав атрибутов, с какими элементами XML-документа они могут использоваться;
- каким образом атрибуты могут применяться в элементах;
- описание сущностей, включаемых в XML-документ.

Практическое пособие содержит:

- описание принципов создания допустимых XML-документов;
- описание базовой разметки DTD;
- пример использования DTD для XML-документа, описывающего метаданные статей журнала «Проблемы физики, математики и техники»;
- практическое задание для студентов.

ТЕМА 1. ОПРЕДЕЛЕНИЕ ТИПА ДОКУМЕНТА (DTD)

1.1. Принципы создания допустимых документов

Одним из методов структурирования документов XML является определение типа документа (Document Type Definitions, DTD). Определение DTD позволяет следить за тем, чтобы документы XML соответствовали не только синтаксическим правилам спецификации этого языка, но и собственным правилам, касающимся его содержания и структуры [1].

Правильно оформленные документы XML всегда имеют правильный синтаксис и правильно вложенную иерархическую (древовидную) структуру, общую для всех данных XML. Само приложение должно выполнять необходимую проверку структуры и содержания, обработку ошибок и интерпретацию данных.

Если описания данных XML отделить от используемых приложений, то взаимодействующие программы смогут использовать одно и то же описание данных, называемое *словарем XML* (XML vocabulary). Группа документов XML с общим словарем называется *типом документа* (document type), а документ, соответствующий типу документа, называется *экземпляром документа*.

Спецификация XML содержит стандартное средство описания типа документов XML: определение типа документа (DTD). Определение DTD представляет собой документ XML, который может включаться в документ XML с данными, а может существовать в виде отдельного документа. DTD описывает правила, показывающие, как надо структурировать документ, какие элементы и разновидности данных следует в него включить, а также определяет значения по умолчанию.

Правильно оформленные документы XML соответствуют базовым правилам структуры и синтаксиса спецификации XML 1.0. Допустимыми (valid) называются те документы, которые соответствуют также правилам синтаксиса, структуры и правилам, определенным в DTD.

1.2. Внутреннее и внешнее DTD

Внутреннее DTD – это определение структуры XML-документа, которое находится непосредственно внутри XML-файла. Все объявления DTD, такие как определения элементов, атрибутов и их типов, прописываются внутри XML-документа, между тегами DOCTYPE.

Определение DTD связывается с документом XML с помощью разметки, называемой декларацией типа документа (Document Type Declaration). Чтобы отличить её от DTD, эту декларацию называют «декларацией DOCTYPE».

Каждый документ XML можно связать с одним и только одним DTD с помощью декларации DOCTYPE. Благодаря этой декларации, проверяющие на допустимость, анализаторы читают DTD (если оно существует) и проверяют документ на соответствие содержащимся в нем правилам.

Декларация DOCTYPE может присутствовать в документе только один раз. Она располагается сразу после декларации XML и предшествует всем элементам и символьным данным. Между декларацией XML и декларацией DOCTYPE могут находиться только комментарии и команды обработки.

Структура декларации DOCTYPE для описания внутреннего DTD:

```
<!-- DTD -->
<!DOCTYPE rootElement
[
    содержание DTD
]>
<!-- XML-документ -->
<rootElement>
...
</rootElement>
```

Декларация DOCTYPE состоит из следующих элементов:

! – означает специальную декларацию XML;

rootElement – имя корневого элемента XML-документа. Это имя обязательно и оно связывает DTD с целым деревом элементов.

Пример XML-документа с подключением внутреннего DTD:

```
<?xml version='1.0' standalone='no' ?>
<!-- Подключение внутреннего DTD -->
<!DOCTYPE journal
[
    содержание DTD
]>
<!-- XML-документ -->
<journal>
...
</journal>
```

Преимущество внутреннего DTD. Внутренние DTD удобны для небольших и простых XML-документов, когда структура документа легко уместается в одном файле.

Недостаток внутреннего DTD. Для больших и сложных XML-документов внутренние DTD могут сделать XML-файл громоздким и менее удобным для чтения и поддержки.

В целом, внутренние DTD являются полезным инструментом для определения структуры XML-документов, особенно в простых случаях, но для более сложных документов предпочтительнее использовать внешние DTD для лучшей организации и читаемости.

Внешнее DTD хранится в отдельном файле, который должен иметь расширение .dtd, например, Journal.dtd, а в XML-документе указывается только ссылка на этот файл.

Структура декларации DOCTYPE для описания внешнего DTD:

```
<!DOCTYPE rootElement  
        SYSTEM или PUBLIC  
        "URI">
```

где rootElement – имя корневого элемента XML-документа;
SYSTEM или PUBLIC – альтернативно используемые ключевые слова;
URI – указание местоположения схемы DTD.

При указании SYSTEM анализатор должен найти DTD по явно указанному URI, например:

```
<!DOCTYPE journal SYSTEM "Journal.dtd">
```

Ключевое слово PUBLIC позволяет указать с помощью URI некорректную ссылку на DTD, посредством вторичного (конкретного) адреса URI. При указании PUBLIC считается, что будет использоваться доступная схема DTD, расположенная на определенном сайте в Интернете. Если для работы с библиотечными XML-документами используется не стандартный, а специализированный анализатор, то он может сам знать, где находится необходимая ему схема DTD. В этом случае не обязательно указывать точное URI расположения файла со схемой DTD, например:

```
<!DOCTYPE journal PUBLIC "world/Journal.dtd">
```

Можно совместить описание PUBLIC с указанием конкретного расположения файла со схемой DTD. В этом случае, если анализатор не найдет доступной схемы, он будет использовать схему, указанную по конкретному адресу, например:

```
<!DOCTYPE library PUBLIC "world/Journal.dtd"  
    "http://www.library.org/dtd/Journal.dtd">
```

Если анализатор или приложение не могут найти DTD по первичному ресурсу (тип PUBLIC), привлекается вторичный адрес (SYSTEM). Обратите внимание, что ключевое слово SYSTEM подразумевается в этом примере, но никогда не включается в декларацию вместе со словом PUBLIC.

Также в XML-документе возможно одновременно использовать оба типа DTD. При использовании внешнего DTD может возникнуть ситуация, когда какие-либо его описания могут не удовлетворять структуре XML-документа. Например, в XML-документе необходимо использовать несколько новых элементов или изменить тип ряда атрибутов. Непосредственно корректировать внешний DTD нельзя, так как его могут использовать другие программы, и можно нарушить правильность их работы. В такой ситуации рекомендуется использовать внешний DTD, а необходимые изменения выполнить во внутреннем DTD. В случае возникновения конфликта между описаниями внешнего и внутреннего DTD декларации DTD из внутреннего подмножества имеют приоритет по сравнению с декларациями из внешнего.

Пример XML-документа с подключением внешнего и внутреннего DTD:

```
<?xml version='1.0' standalone='no' ?>  
<!-- сначала внешний, а за ним внутренний DTD -->  
<!DOCTYPE journal SYSTEM "Journal.dtd"  
[  
    содержание внутреннего DTD  
>  
<!-- XML-документ -->  
<journal>  
    ...  
</journal>
```

ТЕМА 2. БАЗОВАЯ РАЗМЕТКА DTD

Декларации DTD помещаются внутри обычных тегов-ограничителей языка XML (“<” и “>”). Как и декларация DOCTYPE, все декларации DTD содержат восклицательный знак, за которым следует ключевое слово и его параметры [1, 3, 4]

```
<!keyword parameter1  
        parameter2  
        ...  
        parameterN>
```

В декларациях DTD применяются четыре основных ключевых слова:
ELEMENT – декларирует имя типа элемента XML и его допустимых потомков;

ATTLIST – декларирует имена атрибутов элемента XML, а также допустимые и/или предлагаемые по умолчанию их значения;

ENTITY – декларирует специальные ссылки на символы, текстовые макросы (как оператор #define в языке C/C++) и другое повторяющееся содержание из внешних источников (как оператор #include в языке C/C++);

NOTATION – декларирует внешнее содержание не в формате XML (например, двоичные данные изображения) и внешние приложения, обрабатывающие это содержание.

Первые два ключевых слова используются для описания любого содержания XML или модели данных (элементов и их атрибутов). Последние два представляют собой удобный способ создания повторно используемого содержания и методы обработки данных не в формате XML.

2.1. Декларация типа элемента (ELEMENT)

Элементы представляют собой основные строительные блоки данных XML. Каждый документ содержит элементы нескольких типов, часть которых вложены в другие [1–4].

Элемент является единственным обязательным компонентом документа XML – даже большинство тривиальных документов XML должны содержать, по крайней мере, элемент документа. Ни один другой компонент, в том числе декларации xml и DOCTYPE, комментарии, команды обработки и атрибуты элементов, не являются обязательными для создания правильно оформленного документа.

Для описания элементов используется декларация типа элемента. Такая декларация может принимать одну из двух различных форм, в зависимости от значения параметра `category` (в последней форме он подразумевается неявно):

```
<!ELEMENT name category>  
<!ELEMENT name (content_model)>
```

Параметр `name` является именем описываемого элемента.

Параметры `category` (категория содержания) и `content_model` (модель содержания) описывают тип содержания, которое может присутствовать в элементах с данным именем.

Категории содержания элементов. Существует пять категорий содержания элементов:

`ANY` – элемент может содержать любые правильно оформленные данные XML;

`EMPTY` – элемент не может содержать текста или порожденных элементов, разрешены только атрибуты (элемент должен быть пустым);

`element (children content` – дочерняя информация) – элемент содержит только порожденные элементы, в элементах этого типа запрещена текстовая информация;

`mixed (mixed content` – смешанная информация) – элемент может содержать текст и (или) порожденные элементы;

`PCDATA (Parsed Character Data)` – элемент может содержать только текст (символьные данные).

Списки последовательности и выбора. Эти списки состоят из имен типов потомков, разделенных одним или двумя символами оператора списка (и содержащимися в скобках). Рядом с каждым именем типа потомка в списке может стоять оператор множественности.

Существует два оператора списка:

`,` (запятая) – последовательность, потомки должны быть представлены в указанном порядке;

`|` (вертикальная черта) – выбор, разрешается использовать только один из нескольких потомков.

Эти операторы используются для того, чтобы разделить названия типов потомков в модели содержания. Скобки применяются для того, чтобы ограничить всю модель содержания, а также для группирования индивидуальных списков.

Пример использования списка последовательности:

```
<!ELEMENT author
(firstName, middleName, lastName, (PhD | ScD))>
<!ELEMENT firstName (#PCDATA)>
<!ELEMENT middleName (#PCDATA)>
<!ELEMENT lastName (#PCDATA)>
<!ELEMENT PhD EMPTY>
<!ELEMENT ScD EMPTY>
```

Такой декларации может соответствовать следующий экземпляр элемента:

```
<author>
  <firstName>Григорий</firstName>
  <middleName>Яковлевич</middleName>
  <lastName>Перельман</lastName>
  <PhD/>
</author>
```

Этот пример имеет один недостаток: он не показывает способ задания необязательных элементов или какой-либо другой, ограничивающий число вхождений потомков (множественность).

Модель содержания – множественность. Операторы множественности показывают, сколько потомков может быть представлено в модели содержания.

К любой модели содержания и любому потомку можно добавить оператор множественности, который указывает на то, сколько раз модель содержания или потомок может появиться.

Имеется четыре оператора множественности:

[отсутствует] – отсутствие символа оператора множественности показывает, что элемент должен присутствовать только один раз;

? (вопросительный знак) – определяет, что элемент не будет использоваться вообще или будет использоваться только один раз;

* (звездочка) – определяет, что элемент не будет использоваться вообще или может использоваться несколько раз;

+ (знак плюс) – определяет, что элемент будет использоваться один или несколько раз.

Пример использования операторов множественности:

```
<!ELEMENT author
(firstName, middleName*, lastName, (PhD | ScD)?)>
<!ELEMENT firstName (#PCDATA)>
<!ELEMENT middleName (#PCDATA)>
<!ELEMENT lastName (#PCDATA)>
<!ELEMENT PhD EMPTY>
<!ELEMENT ScD EMPTY>
```

Такой декларации может соответствовать несколько экземпляров элемента:

```
<author>
  <firstName>Григорий</firstName>
  <middleName>Яковлевич</middleName>
  <lastName>Перельман</lastName>
  <PhD/>
</author>
```

или

```
<author>
  <firstName>Курт</firstName>
  <lastName>Кэгл</lastName>
</author>
```

В DTD должны быть объявлены все элементы, которые можно будет использовать в XML-документе. Также для каждого элемента должна быть определена модель содержания, позволяющая определить:

- может ли данный элемент входить в другие элементы;
- какие элементы, в каком порядке и количестве могут входить в данный элемент.

При построении модели содержания необходимо помнить, что описание вложенности выполняется только для непосредственно ниже лежащего уровня. Так, элемент, формирующий контейнер первого уровня, может описывать структуру не всех входящих в него элементов и контейнеров, а только элементов следующего за ним второго уровня. Элементы второго уровня описывают структуру элементов и контейнеров третьего уровня и так далее. В модель содержания могут включаться элементы одного из четырех типов:

- элементный;
- текстовый;

- смешанный;
- произвольный.

Такая классификация типов содержания элементов обусловлена возможностью включения в элемент текстовой информации.

Элементный тип состоит только из элементов и не может включать текст. К данному типу относится, например, элемент `author`.

Модель содержания для элементного типа позволяет не только указать, какие элементы будут содержаться в элементе-контейнере, но задать их порядок и количество вложений (ни разу, один раз, много раз).

Текстовый тип включает только текст. К данному типу относятся такие элементы, как `firstName`, `middleName`, `lastName`. Данный тип идентифицируется словом `#PCDATA` в поле модели содержания. Например:

```
<!ELEMENT firstName (#PCDATA)>
```

Смешанный тип может включать как текст, так и элементы. Пример описания элемента смешанного типа:

```
<annotation>  
  Область применения  
  <scope>Интернет</scope>  
</annotation>
```

и его определения

```
<!ELEMENT annotation (#PCDATA | scope)*>
```

На использование смешанного типа накладывается ряд ограничений:

- нельзя определить порядок следования элементов, входящих в смешанный тип;
- нельзя определить количество повторов элементов, входящих в смешанный тип;
- одно и то же имя не может быть представлено в объявлении смешанного типа более одного раза.

Произвольный тип может включать любое содержание. Характеризуется словом `ANY` в поле модели содержания. Пример описания элемента произвольного типа:

```
<!ELEMENT имя_элемента ANY>
```

Создавать и использовать элементы произвольного типа обычно не рекомендуется.

2.2. Декларация атрибутов (ATTLIST)

Атрибуты позволяют описывать метаданные или свойства элемента. Кроме того, атрибуты являются альтернативным способом разметки данных документа [1–4].

Для описания атрибутов элементов используется декларация списка атрибутов ATTLIST. Записывается она в обычном формате декларации DTD с помощью ключевого слова ATTLIST и одного или нескольких определений атрибутов:

```
<!ATTLIST elementName attrName attrType  
          attrDefault defaultValue>
```

Параметры атрибутов:

`elementName` – определяет имя элемента, которому принадлежит атрибут;

`attrName` – имя объявляемого атрибута;

`attrDefault` – определяет поведение анализатора XML. Данный параметр показывает анализатору, обязателен ли атрибут и что делать, если он необязателен и отсутствует;

`defaultValue` – определяет значение атрибута по умолчанию, даже если он в документе не представлен.

Параметры атрибутов по умолчанию (attribute default) позволяют декларации ATTLIST показать, обязателен ли атрибут, и если нет, то как проверяющий на допустимость анализатор будет обрабатывать его отсутствие в документе. Существует четыре параметра атрибутов по умолчанию:

`#REQUIRED` – атрибут должен присутствовать в каждом экземпляре документа;

`#IMPLIED` – атрибут необязателен;

`#FIXED` (плюс значение по умолчанию) – атрибут может присутствовать, а может и отсутствовать в документе. Если он есть, его значение должно быть равно значению по умолчанию; если его нет, анализатор способен сам ввести для него значение по умолчанию;

`значение (я) по умолчанию` – атрибут может присутствовать, а может и отсутствовать в документе. Если он есть, то способен принимать любое значение, соответствующее декларации ATTLIST; если его нет, анализатор может сам ввести для него значение по умолчанию.

Пример использования #IMPLIED и #REQUIRED:

```
<!ATTLIST journalInfo lang CDATA #IMPLIED>  
<!ATTLIST author num CDATA #REQUIRED>
```

В соответствии с этой декларацией все приведенные ниже примеры элемента <journalInfo> будут допустимы:

```
<journalInfo lang="RUS">  
...  
</journalInfo>
```

или

```
<journalInfo>  
...  
</journalInfo>
```

Исключение: внешние ссылки на объекты недопустимы в атрибутах типа CDATA:

```
<journalInfo lang="&Alpha;&Gamma;&Delta;">  
...  
</journalInfo>
```

Атрибуты могут быть необязательными, но при этом ограниченными определенным значением или значениями. Если разрешено только одно значение, для его определения используется параметр #FIXED, при этом анализатор получает право самому вводить это значение, если атрибут отсутствует. Пример использования #FIXED для элемента

```
<secTitle lang="RUS">ФИЗИКА</secTitle>
```

Для того чтобы объявить фиксированный атрибут lang в DTD, используется ключевое слово #FIXED вместе со значением по умолчанию:

```
<!ELEMENT secTitle (#PCDATA)>  
<!ATTLIST secTitle lang CDATA #FIXED "RUS">
```

Если элемент <secTitle> появится без атрибута lang="RUS", анализатор сообщит об ошибке.

Можно позволить анализатору вводить значения по умолчанию, но не требовать их в каждом экземпляре типа элемента (хотя при этом теряется способ ограничивать атрибут одним возможным значением).

```
<!ELEMENT secTitle (#PCDATA)>
<!ATTLIST secTitle lang CDATA "RUS">
```

Теперь документ может содержать элементы `secTitle` с атрибутом `lang` или без него:

```
<secTitle lang="RUS">ФИЗИКА</secTitle>
<secTitle>ФИЗИКА</secTitle>
```

Элемент `<article>` с атрибутами `title`, `author`, `pages` можно описать следующим образом:

```
<ELEMENT article EMPTY >
<!ATTLIST article
  title CDATA #REQUIRED
  author CDATA #REQUIRED
  pages CDATA #IMPLIED
>
```

Так как все потомки `<article>` заменены на атрибуты, модель содержания изменилась, теперь элемент не может иметь никакого содержания (он стал пустым элементом). Объявлено, что три атрибута представляют собой символьные данные (CDATA), причем два из них всегда должны присутствовать (`#REQUIRED`), а один необязательный (`#IMPLIED`).

Типы атрибутов. В рекомендации XML 1.0 определено 10 различных типов атрибутов:

- CDATA – символьные данные (простая текстовая строка);
- перечисляемые (значение1 | значение2 | ... | значениеN) – атрибут всегда принимает одно значение из серии;
- ID – уникальный идентификатор;
- IDREF – ссылка на элемент с атрибутом ID;
- IDREFS – список ссылок ID, разделенных пробелами;
- NMTOKEN – текст без пробелов;
- NMTOKENS – список текстов без пробелов;
- ENTITY – сущность заранее определенного объекта;
- ENTITIES – список сущностей разделенных пробелами;
- NOTATION – значение атрибута должно относиться к типу нотации, явным образом объявленному где-либо в DTD.

Тип атрибутов CDATA. Большинство атрибутов представляют собой обыкновенный текст. Они декларируются с типом CDATA. Например:

```
<!ATTLIST author num CDATA #REQUIRED>
```

Эта декларация означает, что у типа элемента `author` имеется один обязательный атрибут под названием `num`, значением которого является текстовая строка. Если атрибут обязателен, он должен присутствовать во всех экземплярах этого типа элемента, в противном случае документ не будет считаться допустимым.

Приведенной декларации соответствует следующий фрагмент документа:

```
<author num="001"> Купт Кэгл </author>
```

Перечисляемый тип атрибутов. Часто в качестве значения атрибута используется одна из набора определенных текстовых строк.

```
<!ATTLIST artTitle lang (RUS|ENG) #REQUIRED>
```

Типы атрибутов ID / IDREF / IDREFS (связи между элементами). Атрибуты с типом ID содержат имя, уникально идентифицирующее данный экземпляр элемента. Значение атрибута ID должно соответствовать правилам писания имен XML, а также оно должно быть уникальным для документа.

Каждый тип элемента может применить только один атрибут ID, и поэтому на каждый экземпляр элемента можно сослаться с помощью одного уникального идентификатора. Кроме того, все атрибуты этого типа должны быть объявлены как #IMPLIED (необязательные) или #REQUIRED (обязательные).

Пример. Создадим новый тип элемента `<author>`, использующий обязательный атрибут ID и модель содержания с несколькими потомками:

```
<!ELEMENT author (firstName, middleName?, lastName,
  corpName?, email*, biography?)>
<!ATTLIST author authorID ID #REQUIRED>
```

Необходимо, чтобы этот элемент всегда имел уникальный идентификатор, поэтому атрибут `authorID` относится к типу #REQUIRED.

Новая модель содержания типа элемента содержит по одному (и только по одному для каждого) обязательному элементу `<firstName>` и `<lastName>`, а также необязательный элемент `<middleName>`. Кроме того, элемент `<author>` содержит три простых элемента-потомка

типа PCDATA, описывающих название организации, биографические данные автора (ноль или один такой элемент), а также его электронный адрес (ноль или больше).

Соответствующий этому определению фрагмент документа может выглядеть следующим образом:

```
<author authorID="KurtiCagle">
  <firstName>Курт</firstName>
  <lastName>Кэгл</lastName>
  <corpName>Microsoft</corpName>
  <email>kurti_cagle@gmail.com</email>
  <biography>Курт Кэгл – автор книг и разработчик,
специализирующийся в области проблем, связанных с XML
и Интернетом...
</biography>
</author>
```

Чтобы установить связь типа «один-к-одному» между элементом `<author>` и другим элементом, в последнем применяется атрибут типа IDREF, определяющий источник ссылки (link source), (атрибут ID представляет собой цель ссылки, link target). Других элементов (источников ссылки), ссылающихся на данный атрибут ID, может быть много.

Например, элемент `<article>` может установить связь с автором статьи следующим образом:

```
<article author="KurtiCagle">
...
</article>
```

В определении DTD это декларируется следующим образом:

```
<!ATTLIST article author IDREF #REQUIRED>
```

Однако часто у статей бывает несколько авторов (связь типа «один-ко-многим»). Можно было бы для представления этих ссылок попытаться использовать несколько экземпляров атрибута IDREF, но поскольку в правильно оформленном элементе допустим только один атрибут с данным именем, такой подход неприемлем. Пример неправильного задания статьи с несколькими авторами:

```
<article author="KurtiCagle" author="DavidHanter">
...
</article>
```

Язык XML позволяет в один элемент включить несколько атрибутов типа IDREF. Для этого вместо использования нескольких экземпляров атрибута `author` необходимо объявить, что он относится к типу IDREFS:

```
<!ATTLIST article author IDREFS #REQUIRED>
```

Теперь в качестве его значения можно использовать разделенный пробелами список атрибутов IDREF:

```
<article author="KurtiCagle DavidHanter">
...
</article>
```

Анализатор XML представит приложению только одно значение атрибута `author`, и оно само выделит из списка индивидуальные значения IDREF. В данном случае получим два таких значения: `"KurtiCagle"` и `"DavidHanter"`.

Эти типы атрибутов позволяют задавать связи между элементами «один-к-одному» и «один-ко-многим». Такая особенность очень полезна при использовании языка XML в качестве формата обмена данными между системами управления реляционными базами данных.

Текст без пробелов (NMTOKEN). Значением атрибута данного типа в XML-документе может быть любая символьная строка, не включающая символы разметки и пробелы. NMTOKEN рекомендуется применять для объявления атрибутов, являющихся идентификаторами. Пример объявления атрибута `country` с типом NMTOKEN для элемента `author`.

```
<!ATTLIST author country NMTOKEN #IMPLIED>
```

Список текстов без пробелов (NMTOKENS). Значением атрибута в XML-документе может быть несколько строк типа NMTOKEN, разделенных пробелом. Например, автор работает в нескольких ВУЗах: в Гомельском государственном университете имени Франциска Скорины и Белорусском государственном университете транспорта. Если необходимо допустить возможность указания принадлежности автора к нескольким ВУЗам, то атрибут `country` элемента `author` в DTD можно объявить следующим образом:

```
<!ATTLIST author orgName MNTOKENS #IMPLIED>
```

Тогда элемент `author` в XML-документе будет иметь вид:

```
<author orgName="GSU BSUT">
...
</author>
```

Сущности (ENTITY). Все документы XML состоят из единиц хранения информации, называемых сущностями (entities). Сущность документа служит точкой входа для анализатора XML и содержит весь документ целиком (включая элемент документа, дерево его потомков и все декларации, предшествующие и следующие за этим элементом). Основные категории сущностей [3, 4]:

- общие сущности (general entities) – используются внутри любого документа XML;
- параметрические сущности (parameter entities) – могут использоваться только в DTD.

Общие сущности могут быть внутренними и внешними (расположенными внутри DTD или во внешнем файле). Они бывают двух типов:

- анализируемые (parsed) сущности (могут быть либо внутренними, либо внешними);
- неанализируемые (unparsed) сущности (бывают только внешними).

Анализируемая сущность может включать любое правильно оформленное содержание, называемое *текстом замещения* (replacement text). Ссылка на анализируемую сущность осуществляется по имени:

```
<!ENTITY theName "The replacement text">
...
&theName;
```

Неанализируемая сущность может быть, а может не быть текстовой. Кроме того, если она текстовая, то обязательно должна быть текстом XML. Эти сущности всегда являются именованными и могут использоваться только в качестве значения атрибута типа ENTITY или ENTITIES. С неанализируемой сущностью должна быть связана нотация, также идентифицируемая по имени:

```
<!-- определение нотации -->
<!NOTATION png SYSTEM "http://wrox.com/PNG_Viewer.exe">
<!-- определение неанализируемой сущности -->
<!ENTITY theImageRef SYSTEM "image.png" NDATA png>
...
<!-- пустой элемент, ссылающийся на неанализируемую
сущность в значении атрибута -->
<foo img="theImageRef"/>
```

Сущность – это абстрактное понятие, которое представляет следующие типы данных:

- текстовые,
- параметрические,
- двоичные,
- символьные,

и сущность определяет, каким образом анализатор XML может обращаться к этим данным. Синтаксис объявления сущности в DTD в самом общем виде:

```
<!ENTITY имя_сущности "определение_сущности">
```

где имя сущности – имя, под которым сущность может появляться в XML-документе;

определение сущности – зависит от типа информации сущности, а также от того, является сущность внутренней или внешней.

Значение информации, которую представляет сущность, может находиться вне сущности во внешнем файле (внешняя сущность), или присутствовать внутри определения сущности (внутренняя сущность). В случае внешней сущности параметр «определение_сущности» представляет собой URI, указывающий на местоположение информации сущности. В случае внутренней сущности параметр «определение_сущности» непосредственно содержит данные соответствующего типа.

Объявление текстовых сущностей и обращение к ним. Текстовая сущность представляет собой фрагмент текста, замещающий в XML-документе ссылку, соответствующую значению «имя_сущности». Ограничений на размер фрагмента текста нет. Он может состоять из нескольких символов, а может быть большим документом.

Синтаксис текстовой сущности:

- для внутренней сущности

```
<!ENTITY имя_сущности "текстовое значение сущности">
```

- для внешней сущности

```
<!ENTITY имя_сущности SYSTEM или PUBLIC "URI">
```

Использование ключевых слов SYSTEM или PUBLIC такое же, как и в DOCTYPE. URI – местоположения текстового файла с текстовым значением сущности:

```
<!ENTITY description SYSTEM "lib.org/description.txt">
```

Обращение к текстовым сущностям в XML-документе:

&ИМЯ_ССЫЛКИ;

где имя_ссылки должно совпадать с именем сущности, объявленным в DTD. Оно должно начинаться с символа амперсанд (&) и заканчиваться символом «точка с запятой» (;).

Текстовый объект удобно применять для замены повторяющихся одинаковых фрагментов в XML-документах. Пример определения элемента без и с использования сущностей:

```
<!-- определение элемента без использования сущности -->
<orgName>Гомельский государственный университет
        имени Франциска Скорины
</orgName>
```

```
<!-- определение элемента с использованием сущности -->
<!ENTITY GSU_russian "Гомельский государственный
        университет имени Франциска Скорины">
<orgName>&GSU_russian;</orgName>
```

Объявление параметрических сущностей и обращение к ним. Текстовые сущности удобно применять в качестве своеобразных макроподстановок в XML-документах. Параметрические сущности играют аналогичную роль, но уже применительно к DTD, и являются удобным сокращением для повторяющихся деклараций DTD или для фрагментов деклараций.

Синтаксис и значения параметров параметрической сущности определяются аналогично, как и для текстовой сущности, но перед именем сущности ставится символ процента (%).

Синтаксис параметрической сущности:

– для внутренней сущности

```
<!ENTITY % имя_сущности "содержание">
```

– для внешней сущности

```
<!ENTITY % имя_сущности SYSTEM или PUBLIC "URI">
```

Обращение к параметрическим сущностям в DTD:

%ИМЯ_ССЫЛКИ;

где имя_ссылки должно совпадать с именем сущности, объявленным в DTD. Оно должно начинаться с символа процента (%) и заканчиваться символом «точка с запятой» (;). Пример определения атрибутов без и с использования параметрической сущности:

```
<!-- определение атрибутов без использования
  параметрической сущности -->
<!ATTLIST individInfo lang (RUS|ENG) "RUS">
<!ATTLIST artTitle lang (RUS|ENG) "RUS">
<!ATTLIST abstract lang (RUS|ENG) "RUS">

<!-- определение атрибутов с использованием
  параметрической сущности -->
<!ENTITY % rus_eng "lang (RUS|ENG) 'RUS'">
<!ATTLIST individInfo %rus_eng;>
<!ATTLIST artTitle %rus_eng;>
<!ATTLIST abstract %rus_eng;>
```

Использовать параметрические сущности лучше во внешнем DTD. В этом случае параметрическую ссылку можно использовать внутри других объявлений. Для внутреннего DTD так делать нельзя. Параметрическая ссылка может заменять только целое объявление.

Объявление двоичных сущностей. Двоичная сущность, в большинстве случаев, это внешний объект в виде двоичного файла, представляющего графическую, видео, музыкальную и т. п. информацию.

Синтаксис объявления двоичной сущности:

```
<!ENTITY имя_сущности
  SYSTEM или PUBLIC "URI" NDATA тип_файла>
```

где имя_сущности, SYSTEM, PUBLIC и URI определяются так же, как и для текстовой сущности; NDATA – признак двоичной сущности; тип_файла – определяет тип двоичного файла, например gif, mp3.

Для того, чтобы подсказать анализатору XML, с помощью какой программы необходимо обрабатывать указанный двоичный файл, используется объявление NOTATION.

Синтаксис объявления NOTATION:

```
<!NOTATION тип_файла SYSTEM или PUBLIC "URI">
```

где тип_файла – определяет тип двоичного файла, например, gif, mp3; SYSTEM и PUBLIC определяются так же, как и для текстовой сущности; URI – задает местоположение программы, используемой для обработки двоичного файла.

Например, требуется обработать обложку журнала, которая хранится в jpg файле с помощью программы Paint:

```
<!ENTITY picture SYSTEM "journal_cover.jpg" NDATA jpg>
<!NOTATION jpg SYSTEM
"c:/Windows/system32/mspaint.exe">
```

Если анализатор XML знает, как обработать двоичный файл, он может использовать свой собственный механизм обработки, игнорируя объявление NOTATION.

Объявление символьных сущностей. Для любого символа можно объявить сущность и в XML-документе вместо символа использовать ссылку на сущность. Пример использования символьной сущности:

```
<?xml version="1.0"?>
<!DOCTYPE journalInfo
[
  <!ELEMENT journalInfo (title, dateYear)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT dateYear (#PCDATA)>
  <!ENTITY copyright "&#169;">
]>

<journalInfo>
  <title>Проблемы физики, математики и техники</title>
  <dateYear>&copyright;2025</dateYear>
</journalInfo>
```

При просмотре данного XML-документа в браузере, вместо ссылки copyright будет подставлен символ авторского права «©»:

```
<journalInfo>
  <title>Проблемы физики, математики и техники</title>
  <dateYear>©2025</dateYear>
</journalInfo>
```

Пять символов, представленных в таблице 1, предопределены в XML. Их ссылки можно использовать в XML-документе без объявления в DTD.

Таблица 1 – Предопределенные символы

Предопределенный символ	Символьное значение ссылки	Числовое значение ссылки
амперсant (&)	&	&
больше (>)	>	>
меньше (<)	<	<
апостроф (')	"	"
двойные кавычки (")	'	'

2.3. Условные разделы

Условные разделы в DTD используются для определения структуры XML-документа, позволяя указывать, какие элементы могут присутствовать, как они могут быть вложены, и какие атрибуты они могут иметь. Они не являются физическими разделами в документе, а логическими конструкциями, определяющими правила для валидации XML файла [1].

Для создания условных разделов применяются две декларации:

```
<![INCLUDE или IGNORE
[
    условные декларации DTD
]]>
```

Декларации из раздела INCLUDE используются для проверки допустимости, а из раздела IGNORE – читаются, но не обрабатываются. Пример:

```
<!ENTITY %useSubjects "IGNORE">
<!ENTITY %useNoSubjects "INCLUDE">

<![%useSubjects;
[
    <!ELEMENT article (title, author+, subjects)>
    <!ELEMENT subjects (#PCDATA)>
]]>

<![%useNoSubjects;
[
    <!ELEMENT article (title, author+)>
]]>

<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
```

ТЕМА 3. ПРИМЕР ИСПОЛЬЗОВАНИЯ DTD

Опишем XML-документ, содержащий метаданные четырех статей журнала «Проблемы физики, математики и техники»:

ФИЗИКА

УДК 539.12

EDN: QWERTY

СПЕКТРАЛЬНАЯ ЗАДАЧА УРАВНЕНИЯ КЛЕЙНА – ГОРДОНА – ФОКА

Н. В. Максименко¹, С. М. Кучин²

¹Гомельский государственный университет имени Франциска Скорины

²Брянский государственный университет имени И. Г. Петровского

SPECTRAL PROBLEM OF THE KLEIN – GORDON – FOCK EQUATION

N. V. Maksimenko¹, S. M. Kuchin²

¹Francisk Skorina Gomel State University

²I. G. Petrovsky Bryansk State University

Определены собственные значения и собственные функции связанной релятивистской системы на основе уравнения Клейна – Гордона – Фока со смешанной, скалярно-векторной связью.

Ключевые слова: релятивистская связанная система, собственные значения и собственные функции.

The eigenvalues and eigenfunctions of the relativistic bound system based on Klein – Gordon – Fock equation with a mixed, scalar-vector coupling are determined.

Keywords: relativistic bound system, eigenvalues and eigenfunctions.

ЛИТЕРАТУРА

1. Быков, А. А. Потенциальные модели кваркония / А. А. Быков, И. М. Дремин // Успехи физических наук. – 1984, Т. 143, Вып. 1. – С. 3–31.
2. Дремин, И. М. Уравнения движения частиц с переменной массой и удержание кварков / И. М. Дремин, А. В. Леонидов // Письма в ЖЭТФ. – 1983. – Т. 37, вып. 12. – С. 617–619.

Поступила в редакцию 14.06.10.

**МОЛЕКУЛЯРНАЯ СТРУКТУРА И МОРФОЛОГИЯ ПОКРЫТИЙ
ПОЛИЭТИЛЕНА, ЛЕГИРОВАННЫХ ПРИ ИХ ФОРМИРОВАНИИ
ИЗ ГАЗОВОЙ ФАЗЫ НИЗКОМОЛЕКУЛЯРНЫМИ
СОЕДИНЕНИЯМИ**

М. А. Ярмоленко, А. В. Рогачев

Гомельский государственный университет имени Франциска Скорины

**MOLECULAR STRUCTURE AND MORPHOLOGY
OF THE POLYETHYLENE COATINGS ALLOYED WHEN FORMING
FROM GAS PHASE BY LOW-MOLECULAR COMPOUND**

M. A. Yarmolenko, A. V. Rogachev

Francisk Skorina Gomel State University

Определена молекулярная структура и морфология покрытий полиэтилена, содержащих различные по своей природе органические добавки. Установлено, что покрытия, сформированные из продуктов электронно-лучевого диспергирования, имеют более высокую транс-виниленовую ненасыщенность и больший размер надмолекулярных образований в сравнении с покрытиями, полученных термическим диспергированием.

Ключевые слова: морфология, молекулярная структура, полиэтилен.

The molecular structure and morphology of the polyethylene coatings, which inherently contain functional additions, are defined. It was established that the coatings that were received by electron beam dispersion have higher trans-vinylene nonsaturation and bigger size of permolecular formations in comparison with the coatings that were received by thermal dispersion.

Keywords: morphology, molecular structure, polyethylene.

ЛИТЕРАТУРА

1. Gritsenko, K. P. Deposition of Polymers by Vacuum Degradation / K. P. Gritsenko, A. M. Krasovsky // Chem. Rev. – 2003. – Vol. 103, № 9. – P. 3607–3649.

Поступила в редакцию 10.06.10.

**ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ТЕПЛОВЫМ ПРОЦЕССОМ
В СТЕРЖНЕ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ****Р. Габасов¹, Д. С. Кузьменков²**¹*Белорусский государственный университет*²*Гомельский государственный университет имени Франциска Скорины***OPTIMAL CONTROL PROBLEM OF A THERMAL PROCESS
IN THE ROD UNDER UNCERTAINTIES****R. Gabasov¹, D. S. Kuzmenkov²**¹*Belarusian State University*²*Francisk Skorina Gomel State University*

Рассматривается задача оптимального управления тепловым процессом в стержне в условиях неопределенности. Предлагается метод ее решения путем сведения к задачам оптимального наблюдения линейной системой и оптимального управления детерминированной системой по полученным оценкам.

Ключевые слова: тепловой процесс, оптимальная обратная связь по распределению, задача оптимального управления.

An optimal control problem of a thermal process in the rod under uncertainties is considered. The proposed solution method is the reduction to linear system optimal observation problems and optimal control of the determinate system on the obtained evaluations.

Keywords: thermal process, optimal feedback on allocation, optimal control problem.

ЛИТЕРАТУРА

1. Габасов, Р. Принципы оптимального управления / Р. Габасов, Ф. М. Кириллова // Доклады НАН Беларуси. – 2004. – Т. 48, № 1. – С. 15–18.
2. Габасов, Р. Оптимальное управление тепловым процессом / Р. Габасов, Ф. М. Кириллова, Д. С. Кузьменков // Доклады НАН Беларуси. – 2009. – Т. 53, № 1. – С. 5–9.

Поступила в редакцию 30.04.10.

РЕШЕНИЕ КЛАССИЧЕСКОЙ ЗАДАЧИ РЕГУЛИРОВАНИЯ
ОПТИМАЛЬНЫМИ УПРАВЛЕНИЯМИ
ЛИНЕЙНО-КВАДРАТИЧНЫХ ЗАДАЧ

А. В. Лубочкин

Гомельский государственный университет имени Франциска Скорины

SOLUTION OF THE CLASSICAL REGULATION PROBLEM
BY OPTIMAL CONTROLS OF LINEAR-QUADRATIC PROBLEMS

A. V. Lubochkin

Francisk Skorina Gomel State University

В классе ограниченных управлений рассматривается базовая задача классической теории регулирования. Методами оптимального управления строится алгоритм работы регулятора.

Ключевые слова: динамическая система, задача регулирования, ограниченная стабилизирующая обратная связь, вспомогательная задача оптимального управления.

The basic problem of the classical regulation theory in the bounded control class is considered. Optimal control methods are used to construct the regulator work algorithm.

Keywords: dynamic system, regulation problem, bounded stabilizing feedback, auxiliary optimal control problem.

ЛИТЕРАТУРА

1. Габасов, Р. Оптимизация линейной системы управления в режиме реального времени / Р. Габасов, Ф. М. Кириллова, О. И. Костюкова // Известия РАН. Техническая кибернетика. – 1992. – № 4. – С. 3–19.
2. Лубочкин, А. В. Дискретная реализация позиционного решения в линейно-квадратичной задаче с ограничениями / А. В. Лубочкин // Известия ГГУ имени Ф. Скорины, Гомель. – 2003. – № 3 (18). – С. 32–37.
3. Габасов, Р. К методам стабилизации динамических систем / Р. Габасов, Ф. М. Кириллова, О. И. Костюкова // Известия РАН. Техническая кибернетика. – 1994. – № 3. – С. 67–77.

Поступила в редакцию 02.06.10.

XML-документ для описания такой структуры статей журнала Journal.xml:

```
<?xml version='1.0' standalone='no' ?>
<!-- Подключение внешнего файла DTD -->
<!DOCTYPE journal SYSTEM "Journal.dtd"
[
<!-- Описание внутреннего DTD -->
<!ENTITY GSU_russian "Гомельский государственный
    университет имени Франциска Скорины">
<!ENTITY GSU_english "Francisk Skorina Gomel State
    University">
]>

<journal>
  <!-- Компоненты журнала -->
  <journalInfo lang="RUS">
    <title>Проблемы физики, математики и техни-
ки</title>
    <number>2</number>
    <dateYear>2010</dateYear>
  </journalInfo>

  <articles>

    <secTitle lang="RUS">ФИЗИКА</secTitle>

    <!-- Определение элемента Статья -->
    <article>

      <!-- Содержимое статьи: Авторы -->
      <authors>
        <author num="001">
          <individInfo lang="RUS">
            <surname>Максименко</surname>
            <initials>Н. В.</initials>
            <orgName>&GSU_russian;</orgName>
          </individInfo>
          <individInfo lang="ENG">
            <surname>Maksimenko</surname>
            <initials>N. V.</initials>
            <orgName>&GSU_english;</orgName>
          </individInfo>
        </author>
      </authors>
    </article>
  </articles>
</journal>
```

```

<author num="002">
  <individInfo lang="RUS">
    <surname>Кучин</surname>
    <initials>С. М.</initials>
    <orgName>Брянский государственный универси-
тет имени И. Г. Петровского</orgName>
  </individInfo>
  <individInfo lang="ENG">
    <surname>Kuchin</surname>
    <initials>S. M.</initials>
    <orgName>I. G. Petrovsky Bryansk State
University</orgName>
  </individInfo>
</author>
</authors>

```

```

<!-- Содержимое статьи: Название статьи -->
<artTitles>
  <artTitle lang="RUS">СПЕКТРАЛЬНАЯ ЗАДАЧА
УРАВНЕНИЯ КЛЕЙНА - ГОРДОНА - ФОКА</artTitle>
  <artTitle lang="ENG">SPECTRAL PROBLEM
OF THE KLEIN - GORDON - FOCK EQUATION</artTitle>
</artTitles>

```

```

<!-- Содержимое статьи: Аннотация -->
<abstracts>
  <abstract lang="RUS">Определены собственные
значения и собственные функции связанной релятивистской
системы на основе уравнения Клейна - Гордона - Фока со
смешанной, скалярно-векторной связью.</abstract>
  <abstract lang="ENG">The eigenvalues and
eigenfunctions of the relativistic bound system based
on Klein - Gordon - Fock equation with a mixed,
scalar-vector coupling are determined.</abstract>
</abstracts>

```

```

<!-- Содержимое статьи: Коды -->
<codes>
  <udk>539.12</udk>
  <edn>QWERTY</edn>
</codes>

```

```

<!-- Содержимое статьи: Ключевые слова -->

```

```

<keywords>
  <kwdGroup lang="RUS">
    <keyword>релятивистская связанная
система</keyword>
    <keyword>собственные значения и собственные
функции</keyword>
  </kwdGroup>
  <kwdGroup lang="ENG">
    <keyword>relativistic bound system</keyword>
    <keyword>eigenvalues and
eigenfunctions</keyword>
  </kwdGroup>
</keywords>

```

```

<!-- Содержимое статьи: Список литературы -->
<references>
  <reference>Быков, А. А. Потенциальные модели
кваркония / А. А. Быков, И. М. Дремин, А. В. Леонидов
// Успехи физических наук. - 1984, Т. 143, Вып. 1. -
С. 3-31.</reference>
  <reference>Дремин, И. М. Уравнения движения
частиц с переменной массой и удержание кварков /
И. М. Дремин, А. В. Леонидов // Письма в ЖЭТФ. - 1983.
- Т. 37, вып. 12. - С. 617-619.</reference>
</references>

```

```

<!-- Содержимое статьи: Дата поступления -->
<dateReceived>14.06.10.</dateReceived>

```

```

</article>

```

```

<secTitle lang="RUS">ФИЗИКА</secTitle>

```

```

<article>

```

```

<authors>
  <author num="001">
    <individInfo lang="RUS">
      <surname>Ярмоленко</surname>
      <initials>М. А.</initials>
      <orgName>&GSU_russian;</orgName>
    </individInfo>
    <individInfo lang="ENG">
      <surname>Yarmolenko</surname>

```

```

        <initials>M. A.</initials>
        <orgName>&GSU_english;</orgName>
    </individInfo>
</author>

<author num="002">
    <individInfo lang="RUS">
        <surname>Рогачев</surname>
        <initials>A. B.</initials>
        <orgName>&GSU_russian;</orgName>
    </individInfo>
    <individInfo lang="ENG">
        <surname>Rogachev</surname>
        <initials>A. V.</initials>
        <orgName>&GSU_english;</orgName>
    </individInfo>
</author>
</authors>

```

```

<artTitles>

```

```

    <artTitle lang="RUS">МОЛЕКУЛЯРНАЯ СТРУКТУРА
И МОРФОЛОГИЯ ПОКРЫТИЙ ПОЛИЭТИЛЕНА, ЛЕГИРОВАННЫХ ПРИ ИХ
ФОРМИРОВАНИИ ИЗ ГАЗОВОЙ ФАЗЫ НИЗКОМОЛЕКУЛЯРНЫМИ
СОЕДИНЕНИЯМИ</artTitle>

```

```

    <artTitle lang="ENG">MOLECULAR STRUCTURE AND
MORPHOLOGY OF the POLYETHYLENE COATINGS ALLOYED WHEN
FORMING FROM GAS PHASE BY LOW-MOLECULAR
COMPOUND</artTitle>

```

```

</artTitles>

```

```

<abstracts>

```

```

    <abstract lang="RUS">Определена молекулярная
структура и морфология покрытий полиэтилена, содержащих
различные по своей природе органические добавки.
Установлено, что покрытия, сформированные из продуктов
электронно-лучевого диспергирования, имеют более высо-
кую транс-виниленовую ненасыщенность и больший размер
надмолекулярных образований в сравнении с покрытиями,
полученных термическим диспергированием. </abstract>

```

```

    <abstract lang="ENG">The molecular structure and
morphology of the polyethylene coatings, which inherently

```

contain functional additions, are defined. It was established that the coatings that were received by electron beam dispersion have higher trans-vinylene nonsaturation and bigger size of permolecular formations in comparison with the coatings that were received by thermal dispersion.</abstract>

</abstracts>

<codes>

<udk>678.743.41:620.197:621.793</udk>

<edn>ASDFGH</edn>

</codes>

<keywords>

<kwdGroup lang="RUS">

<keyword>морфология</keyword>

<keyword>молекулярная структура</keyword>

<keyword>полиэтилен</keyword>

</kwdGroup>

<kwdGroup lang="ENG">

<keyword>morphology</keyword>

<keyword>molecular structure</keyword>

<keyword>polyethylene</keyword>

</kwdGroup>

</keywords>

<references>

<reference>Gritsenko, K.P. Deposition of Polymers by Vacuum Degradation / K.P. Gritsenko, A.M. Krasovsky // Chem. Rev. - 2003. - Vol. 103, № 9. - P. 3607-3649.</reference>

</references>

<dateReceived>10.06.10.</dateReceived>

</article>

<secTitle lang="RUS">МАТЕМАТИКА</secTitle>

<article>

<authors>

<author num="001">

<individInfo lang="RUS">

<surname>Габасов</surname>

```

        <initials>P.</initials>
        <orgName>Белорусский государственный
университет</orgName>
    </individInfo>
    <individInfo lang="ENG">
        <surname>Gabasov</surname>
        <initials>R.</initials>
        <orgName>Belarusian State
University</orgName>
    </individInfo>
</author>

<author num="002">
    <individInfo lang="RUS">
        <surname>Кузьменков</surname>
        <initials>Д. С.</initials>
        <orgName>&GSU_russian;</orgName>
    </individInfo>
    <individInfo lang="ENG">
        <surname>Kuzmenkov</surname>
        <initials>D. S.</initials>
        <orgName>&GSU_english;</orgName>
    </individInfo>
</author>
</authors>
<artTitles>

    <artTitle lang="RUS">ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ
ТЕПЛОВЫМ ПРОЦЕССОМ В СТЕРЖНЕ В УСЛОВИЯХ НЕОПРЕДЕЛЕННО-
СТИ</artTitle>
    <artTitle lang="ENG">OPTIMAL CONTROL PROBLEM
OF A THERMAL PROCESS IN THE ROD UNDER
UNCERTAINTIES</artTitle>
</artTitles>

<abstracts>
    <abstract lang="RUS">Рассматривается задача
оптимального управления тепловым процессом в стержне
в условиях неопределенности. Предлагается метод ее
решения путем сведения к задачам оптимального наблюде-
ния линейной системой и оптимального управления детер-
минированной системой по полученным оценкам.</abstract>

```

<abstract lang="ENG">An optimal control problem of a thermal process in the rod under uncertainties is considered. The proposed solution method is the reduction to linear system optimal observation problems and optimal control of the determinate system on the obtained evaluations.</abstract>

</abstracts>

<codes>

<udk>517.977</udk>

<edn>ZXCVBN</edn>

</codes>

<keywords>

<kwdGroup lang="RUS">

<keyword>тепловой процесс</keyword>

<keyword>оптимальная обратная связь по распределению</keyword>

<keyword>задача оптимального управления</keyword>

</kwdGroup>

<kwdGroup lang="ENG">

<keyword>thermal process</keyword>

<keyword>optimal feedback on allocation</keyword>

<keyword>optimal control problem</keyword>

</kwdGroup>

</keywords>

<references>

<reference>Габасов, Р. Принципы оптимального управления / Р. Габасов, Ф.М. Кириллова // Докл. НАН Беларуси. - 2004. - Т. 48, № 1. - С. 15-18.</reference>

<reference>Габасов, Р. Оптимальное управление тепловым процессом / Р. Габасов, Ф.М. Кириллова, Д.С. Кузьменков // Докл. НАН Беларуси. - 2009. - Т. 53, № 1. - С. 5-9.</reference>

</references>

<dateReceived>30.04.10.</dateReceived>

</article>

```

<secTitle lang="RUS">МАТЕМАТИКА</secTitle>
<article>

  <authors>
    <author num="001">
      <individInfo lang="RUS">
        <surname>Лубочкин</surname>
        <initials>А. В.</initials>
        <orgName>&GSU_russian;</orgName>
      </individInfo>

      <individInfo lang="ENG">
        <surname>Lubochkin</surname>
        <initials>A. V.</initials>
        <orgName>&GSU_english;</orgName>
      </individInfo>
    </author>
  </authors>

  <artTitles>
    <artTitle lang="RUS">РЕШЕНИЕ КЛАССИЧЕСКОЙ ЗАДАЧИ
РЕГУЛИРОВАНИЯ ОПТИМАЛЬНЫМИ УПРАВЛЕНИЯМИ
ЛИНЕЙНО-КВАДРАТИЧНЫХ ЗАДАЧ</artTitle>
    <artTitle lang="ENG">SOLUTION OF THE CLASSICAL
REGULATION PROBLEM BY OPTIMAL CONTROLS
OF LINEAR-QUADRATIC PROBLEMS</artTitle>
  </artTitles>

  <abstracts>
    <abstract lang="RUS">В классе ограниченных
управлений рассматривается базовая задача классической
теории регулирования. Методами оптимального управления
строится алгоритм работы регулятора.</abstract>
    <abstract lang="ENG">The basic problem of the
classical regulation theory in the bounded control class
is considered. Optimal control methods are used to
construct the regulator work algorithm.</abstract>
  </abstracts>

  <codes>
    <udk>531.36:62-50</udk>
    <edn>SDFGHJ</edn>
  </codes>

```

```

<keywords>
  <kwdGroup lang="RUS">
    <keyword>динамическая система</keyword>
    <keyword>задача регулирования</keyword>
    <keyword>ограниченная стабилизирующая обратная
связь</keyword>
    <keyword>вспомогательная задача оптимального
управления</keyword>
  </kwdGroup>

  <kwdGroup lang="ENG">
    <keyword>dynamic system</keyword>
    <keyword>regulation problem</keyword>
    <keyword>bounded stabilizing
feedback</keyword>
    <keyword>auxiliary optimal control
problem</keyword>
  </kwdGroup>
</keywords>

<references>
  <reference>Габасов, Р. Оптимизация линейной
системы управления в режиме реального времени /
Р. Габасов, Ф. М. Кириллова, О. И. Костюкова // Известия
РАН. Техническая кибернетика. – 1992. – № 4. –
С. 3–19.</reference>

  <reference>Лубочкин, А. В. Дискретная реализация
позиционного решения в линейно-квадратичной задаче
с ограничениями / А. В. Лубочкин // Известия ГТУ
имени Ф. Скорины, Гомель. – 2003. – № 3 (18). –
С. 32–37.</reference>

  <reference>Габасов, Р. К методам стабилизации
динамических систем / Р. Габасов, Ф. М. Кириллова,
О. И. Костюкова // Известия РАН. Техническая кибернетика.
– 1994. – № 3. – С. 67–77.</reference>
</references>

<dateReceived>02.06.10.</dateReceived>

</article>
</articles>
</journal>

```

Представление XML-документа в браузере Google Chrome:

```
▼<journal>
  <!-- Компоненты журнала -->
  ▼<journalInfo lang="RUS">
    <title>проблемы физики, математики и техники</title>
    <number>2</number>
    <dateYear>2010</dateYear>
  </journalInfo>
  ▼<articles>
    <secTitle lang="RUS">ФИЗИКА</secTitle>
    <!-- Определение элемента Статья -->
    ▼<article>
      <!-- Содержимое статьи: Авторы -->
      ▼<authors>
        ▼<author num="001">
          ▼<individInfo lang="RUS">
            <surname>Максименко</surname>
            <initials>Н. В.</initials>
            <orgName>Гомельский государственный университет имени Франциска Скорины</orgName>
          </individInfo>
          ▼<individInfo lang="ENG">
            <surname>Maksimenko</surname>
            <initials>N. V.</initials>
            <orgName>Francisk Skorina Gomel State University</orgName>
          </individInfo>
        </author>
        ►<author num="002">
          ...
        </author>
      </authors>
      <!-- Содержимое статьи: Название статьи -->
      ▼<artTitles>
        <artTitle lang="RUS">СПЕКТРАЛЬНАЯ ЗАДАЧА УРАВНЕНИЯ КЛЕЙНА - ГОРДОНА - ФОКА</artTitle>
        <artTitle lang="ENG">SPECTRAL PROBLEM OF THE KLEIN - GORDON - FOCK EQUATION</artTitle>
      </artTitles>
      <!-- Содержимое статьи: Аннотация -->
      ▼<abstracts>
        <abstract lang="RUS">Определены собственные значения и собственные функции связанной ре
        <abstract lang="ENG">The eigenvalues and eigenfunctions of the relativistic bound syste
      </abstracts>
      <!-- Содержимое статьи: Коды -->
      ▼<codes>
        <udk>539.12</udk>
        <edn>QWERTY</edn>
      </codes>
      <!-- Содержимое статьи: Ключевые слова -->
      ▼<keywords>
        ▼<kwdGroup lang="RUS">
          <keyword>релятивистская связанная система</keyword>
          <keyword>собственные значения и собственные функции</keyword>
        </kwdGroup>
        ▼<kwdGroup lang="ENG">
          <keyword>relativistic bound system</keyword>
          <keyword>eigenvalues and eigenfunctions</keyword>
        </kwdGroup>
      </keywords>
      <!-- Содержимое статьи: Список литературы -->
      ▼<references>
        <reference>Быков, А. А. Потенциальные модели кваркония / А. А. Быков, И. М. Дремин, А. И
        <reference>Дремин, И. М. Уравнения движения частиц с переменной массой и удержание квар
      </references>
      <!-- Содержимое статьи: Дата поступления -->
      <dateReceived>14.06.10.</dateReceived>
    </article>
```

Внешнее DTD для данного XML-документа Journal.dtd:

```
<!ELEMENT journal (journalInfo, articles)>

<!ENTITY % rus_eng "lang (RUS|ENG) 'RUS'">

<!-- Компоненты журнала -->
<!ELEMENT journalInfo (title, number, dateYear)>
<!ATTLIST journalInfo lang CDATA #FIXED "RUS">
<!ELEMENT title (#PCDATA)>
<!ELEMENT number (#PCDATA)>
<!ELEMENT dateYear (#PCDATA)>

<!ELEMENT articles (secTitle, article)+>
<!ELEMENT secTitle (#PCDATA)>
<!ATTLIST secTitle lang CDATA #FIXED "RUS">

<!-- Определение элемента Статья -->
<!ELEMENT article (authors, artTitles, abstracts, codes,
keywords, references, dateReceived)>
<!ELEMENT authors (author+)>
<!ELEMENT author (individInfo+, individInfo+)>
<!ATTLIST author num CDATA #REQUIRED>
<!ELEMENT individInfo (surname, initials, orgName)>
<!ATTLIST individInfo %rus_eng;>
<!ELEMENT surname (#PCDATA)>
<!ELEMENT initials (#PCDATA)>
<!ELEMENT orgName (#PCDATA)>

<!-- Содержимое статьи: Название статьи -->
<!ELEMENT artTitles (artTitle+)>
<!ELEMENT artTitle (#PCDATA)>
<!ATTLIST artTitle %rus_eng;>

<!-- Содержимое статьи: Аннотация -->
<!ELEMENT abstracts (abstract+)>
<!ELEMENT abstract (#PCDATA)>
<!ATTLIST abstract %rus_eng;>

<!-- Содержимое статьи: Коды -->
<!ELEMENT codes (udk, edn)>
<!ELEMENT udk (#PCDATA)>
<!ELEMENT edn (#PCDATA)>
```

```

<!-- Содержимое статьи: Ключевые слова -->
<!ELEMENT keywords (kwdGroup+)>
<!ELEMENT kwdGroup (keyword+)>
<!ATTLIST kwdGroup %rus_eng;>
<!ELEMENT keyword (#PCDATA)>

<!-- Содержимое статьи: Список литературы -->
<!ELEMENT references (reference+)>
<!ELEMENT reference (#PCDATA)>

<!-- Содержимое статьи: Дата поступления -->
<!ELEMENT dateReceived (#PCDATA)>

```

Проверить документ на валидность можно, например, на сайте <https://www.xmlvalidation.com/>, загрузив сначала XML-документ Journal.xml, а затем внешнее описание типа документа Journal.dtd.

Практическое задание

Для заданной предметной области создать не менее трех описаний типов документа (DTD) и на их основании разработать соответствующие XML-документы.

Описание типа документа (DTD) должно включать в себя требования на наличие:

- всех элементов в XML-документе;
- части элементов в XML-документе (различные варианты использования символов «*, ?, +, |»). Причем обязательными являются комбинации данных символов. Хотя бы в одном из DTD должно присутствовать сложное описание (например, «((имя_элемента| имя_элемента*) +, имя_элемента*) +, имя_элемента?»).

Правила для атрибутов должны быть объявлены с учетом их значений (например, FIXED, IMPLIED, REQUIRED, атрибуты по умолчанию, перечисление допустимых значений).

Использовать не менее двух сущностей.

ЛИТЕРАТУРА

1. XML / К. Кэгл [и др.]. – М. : Лори ; Санкт-Петербург : Питер : Питер Пресс, 2006. – 638 с.
2. Extensible Markup Language (XML) [Electronic resource]. – Mode of access: <https://www.w3.org/XML/>. Date of access: 07.07.2025.
3. Document Type Definition [Электронный ресурс]. – Режим доступа: <https://genberm.narod.ru/xml/lections/dtd/dtd.html>. Дата доступа: 07.07.2025.
4. DTD Tutorial [Electronic resource]. – Mode of acces: https://www.w3schools.com/xml/xml_dtd_intro.asp. Date of access: 07.07.2025.

Производственно-практическое издание

Ружицкая Елена Адольфовна,
Лубочкин Александр Васильевич

**WEB-ТЕХНОЛОГИИ: ИСПОЛЬЗОВАНИЕ DTD
ДЛЯ ОПИСАНИЯ СТРУКТУРЫ XML-ДОКУМЕНТА**

Практическое пособие

Редактор Е. С. Балашова
Корректор В. В. Калугина

Подписано в печать 10.10.2025. Формат 60x84 1/16.

Бумага офсетная. Ризография.

Усл. печ. л. 2,56. Уч.-изд. л. 2,80.

Тираж 10 экз. Заказ 538.

Издатель и полиграфическое исполнение:

учреждение образования

«Гомельский государственный университет имени Франциска Скорины».

Специальное разрешение (лицензия) № 02330 / 450 от 18.12.2013 г.

Свидетельство о государственной регистрации издателя, изготовителя,

распространителя печатных изданий в качестве:

издателя печатных изданий № 1/87 от 18.11.2013 г.;

распространителя печатных изданий № 3/1452 от 17.04.2017 г.

Ул. Советская, 104, 246028, Гомель.

