Задание за решения на задачи по Модул 2: XML валидация чрез DTD

Изготвил: Марина Дабова, ФН: 62503, група 3, специалност СИ

Коментарите са обозначени с червен цвят

#### **Задача 1: Свържете и валидирайте дадения по-долу XML документ с дадената DTD граматика по следните 2 начина:**

* С вътрешно DTD
* С външно DTD

Решение:

1. С вътрешно DTD

<?xml version="1.0"?>

<!DOCTYPE shiporder[ <! –- слагаме името на корена и след това DTD декларацията -->

<!ELEMENT shiporder (orderperson,shipto,item+)>

<!ATTLIST shiporder orderid CDATA #REQUIRED>

<!ELEMENT orderperson (#PCDATA)>

<!ELEMENT shipto (name,address,city,country)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT address (#PCDATA)>

<!ELEMENT city (#PCDATA)>

<!ELEMENT country (#PCDATA)>

<!ELEMENT item (title,note?,quantity,price)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT note (#PCDATA)>

<!ELEMENT quantity (#PCDATA)>

<!ELEMENT price (#PCDATA)>

]>

<shiporder orderid="889923">

<orderperson>John Smith</orderperson>

<shipto>

<name>Ola Nordmann</name>

<address>Langgt 23</address>

<city>4000 Stavanger </city>

<country> Norway </country>

</shipto>

<item>

<title>Empire Burlesque</title>

<note>Special Edition</note>

<quantity>1</quantity>

<price>10.90</price>

</item>

<item>

<title>Hide your heart</title>

<quantity>1</quantity>

<price>9.90</price>

</item>

</shiporder>

Декларирането на вътрешна DTD декларация става по следния начин :

Следваме синтаксиса за включване на вътрешна DTD граматика: <!DOCTYPE root-element [element-declarations]>

“<”, документен тип, след това заявяваме елемента-корен, който виждаме от XML документа и като аргумент се слага DTD декларацията (слага се след отваряне “[“. След това затваряне аргумента”]>” и го завършваме като отдолу е XML документа.

След това валидираме. Като в случая няма грешки, затова валидатора показва, че документа е валиден.

1. С външно DTD

shiporder.dtd – файла:

<?xml version="1.0"?> <! -- премахва се -->  
<!DOCTYPE shiporder[ <! -- премахва се -->  
<!ELEMENT shiporder (orderperson,shipto,item+)>  
<!ATTLIST shiporder orderid CDATA #REQUIRED>  
<!ELEMENT orderperson (#PCDATA)>  
<!ELEMENT shipto (name,address,city,country)>  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT address (#PCDATA)>  
<!ELEMENT city (#PCDATA)>  
<!ELEMENT country (#PCDATA)>  
<!ELEMENT item (title,note?,quantity,price)>  
<!ELEMENT title (#PCDATA)>  
<!ELEMENT note (#PCDATA)>  
<!ELEMENT quantity (#PCDATA)>  
<!ELEMENT price (#PCDATA)>  
]> <! -- премахва се -->

xml- файл:

<?xml version="1.0"?>

<!DOCTYPE shiporder SYSTEM "shiporder.dtd"> <! –- слагаме името на корена и след ключовата дума слагаме DTD файла -->

<shiporder orderid="889923">

<orderperson>John Smith</orderperson>

<shipto>

<name>Ola Nordmann</name>

<address>Langgt 23</address>

<city>4000 Stavanger </city>

<country> Norway </country>

</shipto>

<item>

<title>Empire Burlesque</title>

<note>Special Edition</note>

<quantity>1</quantity>

<price>10.90</price>

</item>

<item>

<title>Hide your heart</title>

<quantity>1</quantity>

<price>9.90</price>

</item>

</shiporder>

Външна DTD декларация става със следния синтаксис:<!DOCTYPE root-element SYSTEM "file-name"> . Като съответно имаме :“<”, документен тип, след това заявяваме елемента-корен, който виждаме от XML документа, имаме ключовата дума „SYSTEM“ и наименованието да DTD граматиката. Тя се създава в отделен файл с разширение “.dtd” . След това във валидатора се добавя съдържанието на XML документа, натискаме “validate” като валидатора иска от нас да добавим DTD файла -като се маха декларацията на XML и декларацията на вътрешно DTD, защото иначе ще изпише грешка.

#### **Задача 2: За дадения по-долу XML документ създайте DTD граматика и го валидирайте спрямо нея.**

<?xml version="1.0"?>

<!DOCTYPE games [

<!ELEMENT games (game)>

<!ELEMENT game (home-team,ex-team,scores\*,yellows\*,reds\*)>

<!ATTLIST game score CDATA #REQUIRED>

<!ELEMENT home-team (#PCDATA)>

<!ELEMENT ex-team (#PCDATA)>

<!ELEMENT scores (score+)>

<!ELEMENT score (player)>

<!ATTLIST score me CDATA #REQUIRED>

<!ATTLIST score type CDATA #IMPLIED>

<!ELEMENT player (#PCDATA)>

<!ELEMENT yellows (player+)>

<!ELEMENT reds (player+)>

]>

<games>

<game score="1-1">

<home-team>Roma</home-team>

<ex-team>Lazio</ex-team>

<scores>

<score me="15">

<player>Klose</player>

</score>

<score me="85" type="penalty">

<player>Tox</player>

</score>

</scores>

<yellows>

<player>Tox</player>

<player>Hernanes</player>

</yellows>

<reds>

<player>Kjaer</player>

</reds>

</game>

</games>

Декларираме вътрешна DTD декларация.Създаваме си DTD граматика.Като изписваме документния тип, след това заявяваме елемента-корен, който виждаме от XML документа и като аргумент се слага DTD граматиката като:

Имаме елемент “games”, който има под-елемент “game” и затова се записва в скоби. След това имаме елемент “game” със под-елементи съответно в скобите “home-team,ex-team,scores\*,yellows\*,reds\*”. Като тук “scores”,”yellows”,”reds” са със “\*”, защото може да ги има 0 или повече пъти.След това изписваме списък с атрибути за “game”- като се изписва името на атрибута в случая “score”, типа и указание за стойността като в случая е задължитилен.По долу имаме списък с атрибути за “score” с атрибут “me”, съответния му тип и указание, че е задължитилен и списък с атрибути за “score” с атрибут “type”, типа и “#IMPLIED”, указващ, че може да бъде зададен,но не е задължителен(това е така, защото “penalty” не е задължително). След това се продължава с изписването на всеки елемент аналогично. Като може да се види че при “scores” имаме под-елемент “score” ,който има и “+”, защото може да се среща 1 или повече пъти. Тази ситуация се наблюдава и при “yellows” и “reds”, където “player” също има “+”, защото може да се среща 1 или повече пъти.

След това валидираме. Като валидатора показва, че документа е без грешки и е валиден.

#### **Задача 3: За дадената по-долу DTD граматика създайте XML документ и го валидирайте спрямо нея.**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE Course\_Catalog [

<!ELEMENT Course\_Catalog (Department+)>

<!ELEMENT Department (Title, Chair, Course+)>

<!ATTLIST Department Code (CS | EE | LING) #REQUIRED>

<!ELEMENT Title (#PCDATA)>

<!ELEMENT Chair (Professor)>

<!ELEMENT Course (Title, Description?, Instructors, Prerequisites?)>

<!ATTLIST Course

Number (CS106A | CS106B | CS107 | CS109 | CS124 | CS143 | CS145 | CS221 | CS228 | CS229 | EE108A | EE108B | LING180) #REQUIRED

Enrollment (1070 | 110 | 130 | 180 | 280 | 320 | 500 | 60 | 620 | 90) #IMPLIED

>

<!ELEMENT Description (#PCDATA)>

<!ELEMENT Instructors ((Lecturer, Professor\*) | (Professor+, Lecturer?))>

<!ELEMENT Prerequisites (Prereq+)>

<!ELEMENT Prereq (#PCDATA)>

<!ELEMENT Professor (First\_Name, Middle\_Initial?, Last\_Name)>

<!ELEMENT Lecturer (First\_Name, Middle\_Initial?, Last\_Name)>

<!ELEMENT First\_Name (#PCDATA)>

<!ELEMENT Middle\_Initial (#PCDATA)>

<!ELEMENT Last\_Name (#PCDATA)>

]>

<Course\_Catalog>

<Department Code="CS">

<Title> Computer Science </Title>

<Chair>

<Professor>

<First\_Name> Ivan</First\_Name>

<Middle\_Initial> Ivanov </Middle\_Initial>

<Last\_Name> Ivanov </Last\_Name>

</Professor>

</Chair>

<Course Number="CS106A"

Enrollment="1070">

<Title> </Title>

<Description> </Description>

<Instructors>

<Lecturer>

<First\_Name> Petar</First\_Name>

<Middle\_Initial> Petrov </Middle\_Initial>

<Last\_Name> Petrov </Last\_Name>

</Lecturer>

<Professor>

<First\_Name> Ivan</First\_Name>

<Middle\_Initial> Ivanov </Middle\_Initial>

<Last\_Name> Ivanov </Last\_Name>

</Professor>

</Instructors>

<Prerequisites>

<Prereq> C++ </Prereq>

<Prereq> math </Prereq>

</Prerequisites>

</Course>

</Department>

</Course\_Catalog>

Първо си разместих дадената граматика, за да мога по-добре да структурирам XML-документа и за да намеря, кой е корена и след това всеки под-елемент и т.н.

Корена е „Course\_Catalog“ като той има 1 или повече под-елементи “Department” . Всеки “Department” има “Title”, “Chair” и 1 или повече “Course”. “Department” има и атрибут “Code” като той може да е със стойност “CS” или “EE”, или “LING” и е задължителен. Всеки “Course” има под-елементи “Title”, 0 или един “Description”, “Instructors”, 0 или едно “Prerequisites”. Всеки “Course” има атрибути “Number” със стойности – CS106A или CS106B или CS107 или CS109 или CS124 или CS143 или CS145 или CS221 или CS228 или CS229 или EE108A или EE108B или LING180 (задължителни) и “Enrollment” със стойности 1070 или 110 или 130 или 180 или 280 или 320 или 500 или 60 или 620 или 90 (незадължителни). Под-елемента “Instructors” може да има под-елементи или “Lecturer” и 0 или повече “Professor”, или 1 или повече “Professor” и 0 или един “Lecturer”. Под-елемента “Prerequisites” може да има 1 или повече под-елемента “Prereq”. Всеки “Professor” и “Lecturer” имат съответни под-елементи – “First\_Name”, 0 или един “Middle\_Initial” и “Last\_Name”.

След направата на XML документа, валидираме и валидатора показва, че няма грешка и всичко е както трябва.

#### **Задача 4: За DTD граматиката, намираща се на адрес: http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd, създайте XML документ. Включете в XML документа дадената DTD граматика като публична и го валидирайте спрямо нея.**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<title></title>

</head>

<body></body>

</html>

Използваме синтаксиса за включване на публична външна DTD граматика: <!DOCTYPE rootname PUBLIC FPI URL> като в нашия случай елемента-корен е html и спрямо него си пишем <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" <http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>"> след това си пишем и самия XML- документ.

#### **Задача 5: За дадената по-долу схема създайте DTD граматика и XML документ и го валидирайте спрямо нея. DTD граматиката трябва да изпълнява следните условия:**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE channel [

<!ELEMENT channel (item+,title,link,image\*,language?,description)>

<!ATTLIST channel version CDATA #IMPLIED>

<!ELEMENT item ((title,link,description?) | (title,description?,link) | (link,title,description?)|(link,description?,title)|(description?,link,title)|(description?,title,link))>

<!ELEMENT title (#PCDATA) >

<!ELEMENT link (#PCDATA)>

<!ELEMENT language (#PCDATA)>

<!ELEMENT image ((title,link,url) | (title,url,link) | (link,title,url)|(link,url,title)|(url,link,title)|(url,title,link))>

<!ELEMENT description (#PCDATA)>

<!ELEMENT url (#PCDATA)>

]>

<channel>

<item>

<title></title>

<link></link>

<description></description>

</item>

<title></title>

<link></link>

<image>

<title></title>

<link></link>

<url></url>

</image>

<language></language>

<description></description>

</channel>

По таблицата си създаваме първо DTD граматика. DTD граматиката създаваме като:

Първо виждаме по таблицата, че „channel“ е корена.  Елементът channel има атрибут с име version като той не е задължителен.Записваме го като <!ATTLIST channel version CDATA #IMPLIED>

Под-елементите на channel имат следния ред на подреждане: item, title, link, image, language и description и ги записваме в скобите на елемента channel.Като следим по условие -елементите item и image да се срещат много пъти и item, title, link и description са задължителни(добавяме на item “+”, а останалите елементи image и language – не(съответно им добавяме „\*”и “?”. Трябва да се отбележи и че под-елементите на item (т.е. title, link, description) и image (т.е. title, link, url) нямат определена последователност тоест записваме различните вариации, които за всяко са 6 на брой.Трябва да се отбележи и че под-елементът description на item е незадължителен- това го правим като му добавим „?“.След това по нея създаваме XML файл-а. Валидираме и имаме успешна валидация-т.е. няма грешки.

#### **Задача 6: Съставете DTD граматика, която позволява да бъдат представени в XML документ резултатите от футболните мачове и включва следната информация:**

* 1. Футболните отбори участващи в един мач
* 2. Крайния резултат за всеки мач
* 3. Играчите отбелязали гол в мача
* 4. Времето, в което е отбелязан всеки гол
* 5. Играчите получили наказателни картони (жълти или червени)

<?xml version="1.0"?>

<!DOCTYPE games [

<!ELEMENT games (game)>

<!ELEMENT game (home-team,ex-team,scores\*,yellows\*,reds\*)>

<!ATTLIST game score CDATA #REQUIRED>

<!ELEMENT home-team (#PCDATA)>

<!ELEMENT ex-team (#PCDATA)>

<!ELEMENT scores (score+)>

<!ELEMENT score (player)>

<!ATTLIST score me CDATA #REQUIRED>

<!ATTLIST score type CDATA #IMPLIED>

<!ELEMENT player (#PCDATA)>

<!ELEMENT yellows (player+)>

<!ELEMENT reds (player+)>

]>

Създаваме DTD граматиката : Имаме елемент “games”, който има под-елемент “game” и затова се записва в скоби. След това имаме елемент “game” със под-елементи съответно в скобите “home-team,ex-team,scores\*,yellows\*,reds\*”. Като тук “scores”,”yellows”,”reds” са със “\*”, защото може да ги има 0 или повече пъти(наказателните картони не са задължителни).След това изписваме списък с атрибути за “game”- като се изписва името на атрибута в случая “score”, типа и указание за стойността като в случая е задължитилен.По долу имаме списък с атрибути за “score” с атрибут “me”, съответния му тип и указание, че е задължитилен и списък с атрибути за “score” с атрибут “type”, типа и “#IMPLIED”, указващ, че може да бъде зададен,но не е задължителен(това е така, защото “penalty” не е задължително). След това се продължава с изписването на всеки елемент аналогично. Като може да се види че при “scores” имаме под-елемент “score” ,който има и “+”, защото може да се среща 1 или повече пъти. Тази ситуация се наблюдава и при “yellows” и “reds”, където “player” също има “+”, защото може да се среща 1 или повече пъти.

#### **Задача 7: Да се включи в DTD файла от задача 5 и да се използва в XML файл:**

* 1. Една вътрешна (System) и една външна декларация (Public) на DTD нотация за някои от MIME типовете image/jpeg, image/png или image/gif. След това да се декларира entity използващо тези 2 нотации
* 2. Общо entity задаващо стойността на елемента link на image
* 3. Параметризирано entity със стойност "title" и да се използва навсякъде където тази дума се среща в DTD файла
* 4. Вложено еntity в entity
* 5. Рекурсивно entity

channel.dtd-файл:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE channel [

<!ELEMENT channel (item+,%Title;,link,language,image\*,description)>

<!ATTLIST channel version CDATA #IMPLIED>

<!ELEMENT item ((%Title;,link,description?)|(link,%Title;,description?)|(%Title;,description?,link)| (link,description?,%Title;)|(description?,%Title;,link)| (description?,link,%Title;))>

<!ELEMENT link (#PCDATA)>

<!ELEMENT language (#PCDATA)>

<!ELEMENT image ((%Title;,link,url)|(%Title;,url,link)|(link,url,%Title;)|(link,%Title;,url)|(url,%Title;,link)| (url,link,%Title;))>

<!ELEMENT description (#PCDATA)>

<!ELEMENT url (#PCDATA)>

<!NOTATION jpg SYSTEM "image/jpeg"> <!-- вътрешна (System) декларация на DTD нотация-->

<!NOTATION gif PUBLIC "image/gif"> <!—външна (Public) декларация на DTD нотация-->

<!ENTITY JPGent SYSTEM "http://www.tur.org.uk/аа.jpg" NDATA jpg> <!—декларираме entity използващо нотацията-->

<!ENTITY GIFent PUBLIC "https://www.ox.ac.uk/display\_images/logo.gif" NDATA gif> <!—декларираме entity използващо нотацията-->

<!ENTITY link link.com> <!-- Общо entity задаващо стойността на елемента link на image -->

<!ELEMENT %Title (#PCDATA)>

<!ENTITY % Title "title" > <!-- Параметризирано entity със стойност "title" -->

]>

xml- файл:

<!DOCTYPE channel SYSTEM "channel.dtd">

<channel>

<item>

<title></title>

<link></link>

<description></description>

</item>

<title></title>

<link></link>

<image>

<title></title>

<link></link>

<url></url>

</image>

<language></language>

<description></description>

</channel>

Използваме: Синтаксис за дефиниране на нотации: <!NOTATION Name SYSTEM SystemLiternal> или <!NOTATION Name PUBLIC PublicID>; Синтаксис за дефиниране на общо entity: <!ENTITY name definition>;Синтаксис за дефиниране на външно общо entity: <!ENTITY name SYSTEM uri> или <!ENTITY name PUBLIC FPI uri>;Синтаксис за дефиниране на общо entity, което използва нотации: <!ENTITY name SYSTEM/PUBLIC uri NDATA notationName>;Синтаксис за дефиниране на параметрично entity: <!ENTITY % name definition> или <!ENTITY % name SYSTEM uri> или <!ENTITY % name PUBLIC FPI uri>