

Проект "Болница"

Тема на проекта: Болница - Пластична хирургия

Изготвил: Стоян Тинчев Тинчев

Факултетен номер: ЗМИ0700193

Съдържание

1. Версия „XML схема“

- [a. Описание на предметната област и на заданието](#)
- [b. Дефиниране на схемата](#)
- [c. Добавяне на примерно съдържание](#)

2. Версия „описание на DOM съответстващ на схемата“

- [a. Екземпляри на документа](#)
- [b. Конструирание и валидиране на всеки екземпляр чрез DOM](#)

3. Версия „набор от XSLT трансформации“

- [a. 2 различни стилови таблици \(XSLT\) за трансформиране на XML документите в друг XML код](#)
- [b. 2 различни стилови таблици \(XSLT\) за трансформиране на XML документите в чист текст](#)

4. Заклучения

- [a. Подобрения с добавяне на допълнителни атрибути, логика итн](#)
 - [b. Доразвиване на схемата с добавяне на повече трансформации](#)
-

Версия „XML схема“

а. Описание на предметната област и на заданието

Проектът моделира информационна система за управление на данни за пластична хирургия в болница. Системата съхранява информация за пациентите, процедурите, лекарите и резултатите. Данните се описват чрез XML схема, която дефинира структурата на документа и отношенията между отделните елементи.

Схемата отразява:

- Пациенти и тяхната медицинска информация (име, възраст, история на заболявания).
- Процедури (име на процедурата, дата, описание).
- Лекарите (име, специалност, години опит).
- Резултати от процедурите (оценка, усложнения, снимков материал).

в. Дефиниране на схемата

- **Основни елементи:**

- `Hospital`: коренен елемент, съдържа всички данни.
- `Patients` и `Procedures`: групи вложени елементи, съответстващи на пациенти и процедури.

- **Минимум 20 елемента:**

- Елементи като `PatientID`, `Name`, `Age`, `MedicalHistory`, `ProcedureID`, `ProcedureName`, `DoctorName` и др.

- **Атрибути (5 броя):**

- Примерни атрибути: `ProcedureID`, `PatientID`, `DoctorID`.

- **Две групи вложени елементи:**

- Група "Пациенти" съдържа елементи за пациент и техните процедури.
- Група "Процедури" съдържа детайлите за всяка процедура, включително лекари и резултати.

- **Честота:**

- Всяка група може да се среща многократно благодарение на `maxOccurs="unbounded"`.

с. Добавяне на примерно съдържание

Примерният XML документ съдържа реалистични данни, описващи конкретни пациенти, техните процедури и резултати.

- **Patients:** Съдържа данни за двама пациенти, техните медицински досиета и списък с извършени процедури.
- **Procedures:** Включва подробности за всяка процедура, като име на процедурата, лекуващ лекар и резултати от процедурата.
- **Results:** Описва резултатите от процедурата, усложнения (ако има) и път към изображенията преди/след.

Версия „описание на DOM съответстващ на схемата“

а. Екземпляри на документа

В тази част на проекта се създават 5 различни екземпляра на XML документа, който е описан в първата част (XML схема). Всеки от тези екземпляри е:

- Добре конструиран и валиден спрямо дефинираната XSD схема (`hospital.xsd`).
- Съдържанието на елементите е на кирилица и на български език.
- Файловете (например `1.xml`, `2.xml`, `3.xml`, `4.xml`, `5.xml`) съдържат различни данни, като всяка процедура, пациент и лекар се различават от предишните.

б. Конструирание и валидиране на всеки екземпляр чрез DOM

За единия от екземплярите (например 5.xml) е необходимо да се демонстрира генерирането му чрез DOM API, използвайки език за скриптове (JavaScript). Този екземпляр се създава динамично от нулата, без ръчно писане на XML, като:

- Се използват DOM методи за създаване на елементи, задаване на текстови възли и вгнездяване на елементите един в друг.
- Генерираният документ е валиден спрямо hospital.xsd.
- Текстовото съдържание отново е на български език.
- Примерен HTML/JavaScript файл (dom-generate.html) показва как DOM може да бъде използван, за да създаде валиден XML документ, който след това да бъде сериализиран и прегледан в браузър.

Версия „набор от XSLT трансформации“

а. 2 различни стилкови таблици (XSLT) за трансформиране на XML документите в друг XML код

В този раздел се представят две XSLT стилкови таблици, които извеждат **само определени части** от данните в нов XML формат:

1. `xslts/xml-to-xml/extract-patients.xsl` – извлича данни за пациентите (ID, име, възраст и др.) и ги записва в нов XML документ.
2. `xslts/xml-to-xml/extract-doctors.xsl` – извлича данни за лекарите (ID, име, специалност и т.н.) и ги записва в нов XML документ.

Така можем да получим опростени XML файлове, съдържащи точно определена информация от основния документ.

Примерна команда с `xsltproc`:

```
xsltproc xslts/xml-to-xml/extract-patients.xsl hospital.xml > patients.xml
```

(Аналогично и за извличане на данни за лекарите.)

б. 2 различни стилкови таблици (XSLT) за трансформиране на XML документите в чист текст

Тук целта е да генерираме **plain text** (обикновен текстов файл), в който да се вижда само текстовата информация от XML документа:

1. `xslts/xml-to-txt/patients-to-text.xsl` – генерира текстов списък на пациенти, тяхната възраст и медицинска история, както и процедурите, които са им извършени.
2. `xslts/xml-to-txt/procedures-to-text.xsl` – генерира текстов списък на всички процедури, лекари и резултати.

Примерна команда с `xsltproc`:

```
xsltproc xslts/xml-to-txt/procedures-to-text.xsl hospital.xml > procedures.txt
```

(Създава текстов файл `procedures.txt`.)

Заклучения

а. Подобрения с добавяне на допълнителни атрибути, логика итн

Проектът би могъл да бъде разширен с:

- Допълнителни атрибути в XSD схемата, напр. `gender="male|female"` при елемента `Patient`, статус на процедурата, ниво на риска и т.н.
- Повече логика в DOM генерирането (напр. динамично добавяне на пациенти според данни от форма).
- По-сложни зависимости между елементите, като съвместяване на множество лекари за една процедура.

б. Доразвиване на схемата с добавяне на повече трансформации

- Създаване на **по-детайлни XSLT** за HTML, включително таблици с обобщени статистики, разширени описания и снимков материал.
 - Добавяне на **XSL-FO** стилови таблици за генериране на PDF от XML.
 - Създаване на **динамични уеб страници**, които автоматично зареждат и визуализират XML документа в различен формат (HTML, JSON и др.).
-