Втори тест по XML

Общ брой точки 30/31

✓ Събитието processingInstruction възниква, когато SAX парсерът достигне до всяка една инструкция за обработка, включително и до XML декларацията 1/1

- ○ Вярно
- ◉ Невярно ✓

✓ SAX служи както за четене на XML документи, така и за генериране на XML 1/1

- ◉ Невярно ✓
- ○ Вярно

✓ XPath изразът //book[@pages] връща 1/1

- ◉ всички book елементи, които имат атрибут pages ✓
- ○ първият book елемент, който има атрибут pages
- ○ първият book елемент, който има непразен атрибут pages
- ○ всички book елементи, които имат непразен атрибут pages

✓ XSL се използва за: 1/1

- (●) трансформиране на XML документ към друг текстов документ ✓
- ( ) трансформиране на XML документ към друг документ само в HTML формат
- ( ) трансформиране на XML документ към друг документ само в XML или HTML формат
- ( ) трансформиране на XML документ към друг документ само в XML формат

✓ Отдалечените XLink ресурси винаги представяват външни за документа ресурси. 1/1

- ( ) Верно
- (●) Неверно ✓

✓ XSL елементът Apply-Tempates се използва вътре в един шаблон (template) за извикване на други шаблони. Той: 1/1

- (●) активира рекурсивно обработката на всички наследници на елемента, за който се отнася ✓
- ( ) активира нерекурсивно обработката на всички наследници на елемента, за който се отнася

✓ DocumentType::Node Interface се използва за получаване на информация за документ описан в DTD 1/1

- (●) DOM 1.0 не разрешава редактиране на този възел ✓
- ( ) DOM 1.0 разрешава редактиране на този възел

✓ При парсване на XML документи посредством StAX, можем да се придвиждваме само напред в XML документа 1/1

- (●) Истина ✓
- ( ) Лъжа

✓ В SAX приложените области, имащи достъп до XML сорса 1/1

- ( ) не трябва да бъдат регистрирани от програмиста за callback функции, тъй като те са listeners
- (●) трябва да бъдат регистрирани (от програмиста) за callback функциите на парсера ✓
- ( ) са регистрирани за callback функции или от програмиста, или от парсера
- ( ) Option 4

✓ При използването на XPath text() функцията, ние избираме: 1/1

- ( ) същият текстов контекст както когато използваме <xsl:value-of select='.'> елемента
- ( ) текстваия контекст на елемент и текстовия контекст на всички наследници на елемента
- (●) текстовия контекст само на елемент ✓
- ( ) текстовия котекст на всички наследници на елемента

✓ С един DOM Element обект: 1/1

- ○ може да направите разлика между подразбиращаз се (default) стойност, определена в DTD, и стойността, дадена в XML файла
- ◉ не може да направите разлика между подразбиращаз се (default) стойност, определена в DTD, и стойността, дадена в XML файла ✓

✓ XMLReader в SAX 2.0 разширява стандартния Java Reader интерфейс 1/1

- ○ Вярно
- ◉ Невярно ✓

✗ Всеки DOM възел (Node) може да има деца 0/1

- ○ вярно
- ◉ невярно ✗
- ○ зависи от децата

Правилен отговор

- ◉ вярно

✓ Разгледайте <xsl:value-of> елемента. Ако стойността на неговия select атрибут е select='.' тогава ние избираме 1/1

- текстовия контекст само на елемента
- същия текстов контекст, както когато използваме text() функцията
- ✓ текстовия контекст на елемента и текстовия контекст на всички наследници на елемента
- текстовия контекст на всички наследници на елемента

✓ SAX Element обектите 1/1

- ✓ могат да разграничават атрибутите, дефинирани изрично, от тези специфицирани в DTD
- не могат да разграничават атрибутите, дефинирани изрично, от тези специфицирани в DTD
- зависи от SAX парсера

✓ XPath изразът ./book[author/last = "пробен изпит"] връща 1/1

- всички book елементи, коит имат елемент author с атрибут last равен на "пробен изпит"
- елемент last със стойност "пробен изпит", който има за баща елемент author с поделемент book - наследник на текущия елемент
- всички book елементи, които имат елемент author с поделемент last равен на "пробен изпит"
- всички last елементи със стойност "пробен изпит", които имат за баща елемент author с поделемент book
- ✓ елемент book - наследник на текущия елемент, който има елемент author с поделемент last равен на "пробен изпит"

✓ При прилагане на XSLT трансформацията: <xsl:template match="name"> <xsl:element name="{.}"> Very nice </xsl:element> </xsl:template> за документ <names> <name> Bob </name> <name> Steve </name> </names> имената на създадените елементи в резултатното дърво ще бъдат 1/1

- ○ с имената на атрибутите в изходящото дърво
- ○ с имената на елементите в изходящото дърво
- ○ със съдържанието на атрибутите в изходящото дърво
- ○ с името "name"
- ◉ със съдържанието на елементите в изходящото дърво ✓

✓ Кое от твърденията е истина 1/1

- ◉ само DOM Element обектите имат атрибути ✓
- ○ и DOM Element и DOM Node обектите имат атрибути
- ○ само DOM Node обектите имат атрибути

✓ Приложения, които имат нужда от сложни структурни манипулации на много от XML елементите, трябва да използват 1/1

- ○ StAX API
- ○ SAX
- ○ XSLT
- ○ CSS
- ◉ DOM ✓

✓ В XSLT, вземането на решение кои елементи ще бъдат обработени се задава със следния XSLT елемент 1/1

- <xsl:process-template>
- <xsl:value-of>
- <xsl:template>
- <xsl:for-each>
- (●) <xsl:apply-templates> ✓

✓ Изпълнението на XSLT декларациите <xsl:value-of select="."/> и <xsl:value-of select="text()"/> води 1/1

- винаги до различни резултати
- (●) до един и същ или до различни резултати в зависимост от типа на съдържанието на текущия елемент ✓
- винаги до един и същ резултат

✓ Целта на валидация на XML документ от XML парсера е да се провери дали XML документът е добре структуриран (well-formed). 1/1

- верно
- (●) неверно ✓

✓ В един DOM Element обект 1/1

- ○ може да направите разлика между подразбираща се (default) стойност определена в DTD, и стойността, дадена в XML файла
- ◉ не може да направите разлика между подразбираща се (default) стойност определена в DTD, и стойността, дадена в XML файла ✓

✓ Методът getAttributes() на DOM интерфейса Node връща 1/1

- ◉ NamedNodeMap ✓
- ○ Attr
- ○ Text
- ○ NodeList

✓ Изберете едно 1/1

- ○ както SAX, така и StAX използват pull парсване
- ◉ SAX използва push парсване, а StAX - парсване от тип pull ✓
- ○ както SAX, така и StAX използват push парсване
- ○ SAX използва pull парсване, а StAX - парсване от тип push

✓ За разлика от SAX, при използване на StAX можем да се движим както напред, така и назад в XML документа. 1/1

- ◉ Неверно ✓
- ○ Верно

✓ Атрибутът xml:base задава: 1/1

- (●) база за относителни URI връзки към външни за документа ресурси ✓
- ( ) база за задаване ан други мета-атрибути
- ( ) базов URI за дефиниране на пространство от имена
- ( ) база за сливане на XML документи

✓ При избиране на елемент, наречен MyElem и имащ атрибут Attr със стойност title, в XSLT ние трябва да използваме 1/1

- ( ) select="MyElem(@Attr='title')"
- ( ) select="MyElem{@Attr='title'}"
- (●) select="MyElem[@Attr='title']" ✓
- ( ) select="MyElem[Attr='title']"

✓ За постигане на по-малък, ефикасен и бърз код с използване на StAX се препоръчва 1/1

- ( ) iterator API
- ( ) StAX Direct Mapping API
- ( ) StAX Events API
- (●) cursor API ✓

✓ DOMException връща HIERARCHY_REQUEST_ERR при опит за 1/1

- заявка за получаване на йерархията на атрибут
- заявка за получаване на йерархията на възел с дълбочина, по-голяма от съществуващата за възела
- заявка за получаване на йерархията на лемент без наследници
- (●) вмъкване на възел на неподходящо място в йерархията на DOM дървото ✓

✓ Ако сме дефинирали XSL променлива като <xsl:variable name="price"> low </xsl:variable>, то тя може да се използва в XSL елемент като: 1/1

- (●) <xsl:value-of select="$price"/> ✓
- <xsl:value-of select="@price"/>
- <xsl:value-of select="{$price}"/>
- <xsl:value-of select="price"/>