

XML_FINALS

Общ брой точки 160/161

✓ Параметрично Entity:

1/1

- ☐ се дефинира в XML документа и може да се използва в който и да е XML документ
- ☐ се дефинира в маркър декларации в DTD, но може да се използва в който и да е XML документ
- ☒ се дефинира в маркър декларации в DTD и може да се използва само в DTD дефиниции ✓

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

1/1

- ☐ същият текстов контекст както когато използваме <xsl:value-of select='.'> елемента
- ☐ текстовия контекст на елемента и текстовия контекст на всички наследници на елемента
- ☒ текстовия контекст само на елемента ✓
- ☐ текстовия контекст на всички наследници на елемента

✓ Чрез XPointer можем да реферираме към повече от един елемента в XML документ. 1/1

- ☐ Неверно
- ☒ верно ✓



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

- ☐ Верно
- ☒ Неверно



✓ Кой(кои) RDF контейнер(и) задава(т) ресурси или литерали, които могат да бъдат евентуално дублирани: 1/1

- ☐ rdf:Seq и rdf:Alt
- ☐ Само rdf:Bag
- ☒ rdf:Bag и rdf:Seq
- ☐ rdf:Bag и rdf:Alt
- ☐ Само rdf:Seq
- ☐ Само rdf:Alt



✓ Разгледайте P:first-letter { font-size: 200% } . Това е: 1/1

- ☐ CSS клас
- ☐ псевдо-атрибут (pseudo-attribute)
- ☐ нито едно от по-горе посочените
- ☒ псевдо-елемент (pseudo-element)



✓ Кой от елементите ISBN по-долу не се описва от дефиницията `<element name="ISBN" type="cat:ISBNTYPE"/>`, където ISBNTYPE е: 1/1

- ☐ 1-23-579321-3
- ☐ 4-256-76435-4
- ☐ 63247-84365-12345
- ☐ 12345-54321-98765
- ☒ 6-32-47843-5



✓ Ако свойство P е за субекта S, който участва в RDF тройката S P O, и имаме `P rdfs:domain C`, то следва, че: 1/1

- ☒ S rdfs:type C
- ☐ S rdfs:type P
- ☐ C rdfs:type S
- ☐ P rdfs:type S
- ☐ P rdfs:type C



✓ `xml:id` е псевдо-атрибут, който задава условие за уникална стойност, 1/1 като това:

- ☐ е необходимо да се декларира в XML схема.
- ☐ е необходимо да се декларира в DTD или XML схема.
- ☐ е необходимо да се декларира в DTD.
- ☒ не е необходимо да се декларира в DTD или XML схема.



✓ 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



✓ RDF Literals могат да бъдат зададени само като обект в RDF тройка. 1/1

- ☐ Неверно
- ☒ Верно



✓ rdf:Bag задава:

1/1

- ☒ група на не подредени ресурси или литерали, с вероятно дублиране
- ☐ група на подредени ресурси или литерали, с вероятно дублиране
- ☐ група от ресурси или литерали, които са алтернативи
- ☐ група на подредени ресурси или литерали, без дублиране
- ☐ група на не подредени ресурси или литерали, без дублиране



✓ creator, title, publisher, contributor, date и format са име на елементи, 1/1
дефинирани от:

- ☐ RDFS
- ☒ DC
- ☐ FOAF
- ☐ OWL



✓ XSL кодът, показан на фигурата, ще даде като резултат: 1/1

- ☒ изходните XML елементи ще бъдат с имена, еднакви със съдържанието на елементите от сорс дървото, и със съдържание "My own contents!" ✓
- ☐ същите елементи в изходния XML файл както в сорс XML файла, но с добавено в края съдържание "My own contents!"
- ☐ същите елементи в изходния XML файл както в сорс XML файла, и с коментар "My own contents!" за всеки един елемент

✓ Кой от следните изрази е валиден пример за дефиниране на избор (choice) от елементи в DTD: 1/1

- ☐ (A ~ B ~ C)
- ☒ (A | B | C)
- ☐ (A or B or C)
- ☐ (A, B, C)



✓ Могат ли екземплярите на класовете да бъдат част от онтологията, описваща тези класове? 1/1

☒ Да



☐ Не

✓ В XML Schema, ние можем да построим производни (derived) types: 1/1

☐ само с restrictions

☒ както с extensions, така и с restrictions



☐ само с extensions

✓ В DOM, неопределена колекция от възли се дефинира чрез интерфейс с име: 1/1

☐ UnorderedNodeCollection

☐ UnorderedNodeSet

☐ NodeAttr

☐ NodeList

☒ NamedNodeMap



✓ Атрибутите на XML са чувствителни към регистъра. 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

✓ Свойството `rdfs:member` е супер-свойство от всички свойства, които са екземпляри на: 1/1

- ☐ `rdfs:ContainerProperty`
- ☐ `rdfs:MemberOfProperty`
- ☒ `rdfs:ContainerMembershipProperty` ✓
- ☐ `rdfs:MembershipProperty`
- ☐ `rdfs:MemberProperty`

✓ Кой от следните елементи не е с грешен XML синтаксис:

1/1

- ☐ `<birth>28.04.1990, Gabrovo</ birth>`
- ☐ `<birth>28.04.1990, Gabrovo< /birth>`
- ☒ `<birth>28.04.1990, Gabrovo</birth >` ✓
- ☐ `<birth>28.04.1990, Gabrovo</bir th>`
- ☐ Всичките останали отговори съдържат елемент с грешен XML синтаксис.



✓ Всеки два OWL класа принципно могат да се припокриват, т.е да имат общи екземпляри: 1/1

- ☒ Верно
- ☐ Неверно



✓ rdf:Seq задава: 1/1

- ☒ група на подредени ресурси или литерали, с вероятно дублиране
- ☐ група на неподредени ресурси или литерали, без дублиране
- ☐ група на подредени ресурси или литерали, без дублиране
- ☐ група на неподредени ресурси или литерали, с вероятно дублиране
- ☐ група на алтернативни ресурси или литерали



✓ Изберете верния отговор: 1/1

- ☐ Всеки добре конструиран XML документ е валиден, но обратното не е задължително вярно.
- ☐ Всеки валиден XML документ е добре конструиран, както и обратното - всеки добре конструиран XML документ е валиден.
- ☐ Всички останали твърдения са неверни.
- ☒ Всеки валиден XML документ е добре конструиран, но обратното не е задължително вярно.



✓ CDATA съдържанията са:

1/1

- ☐ или парсвани, или игнорирани от XML парсерите в зависимост от CDATA директивата
- ☒ игнорирани от XML парсерите
- ☐ парсвани от XML парсерите



✓ Ако свойството P е дефинирано с обхват (range) C и ресурсът R е обект в тройка с предикат P, то следва, че:

1/1

- ☐ C rdf:type R
- ☐ P rdf:type R
- ☐ C rdf:type P
- ☒ R rdf:type C
- ☐ R rdf:type P



✓ Всички членове на подклас OWL са членова на супер-класовете от този клас:

1/1

- ☒ Вярно
- ☐ Невярно



✓ Кое от твърденията е истина:

1/1

- ☒ само DOM Element обектите имат атрибути
- ☐ само DOM Node обектите имат атрибути
- ☐ и DOM Element, и DOM Node обектите имат атрибути



✓ Разгледайте имената на XML елементите <xml-tag> and <tag-xml>. Те са: 1/1

- ☐ и двата невалидни
- ☐ и двата валидни
- ☒ първият е невалиден, вторият е валиден
- ☐ нито едно от по-горе посочените
- ☐ първият е валиден, вторият е невалиден



✓ Най-общо маркър езиките включват:

1/1

- ☐ структурни (layout), функционални (action), семантични (meaning) но не и стилистични (appearance) описания
- ☐ Стилистични (appearance), структурни (layout), семантични (meaning) но не и функционални (action) описания
- ☐ Стилистични (appearance), структурни (layout), функционални (action) но не и семантични (meaning) описания
- ☒ Структурни (layout), функционални (action), семантични (meaning) и стилистични (appearance) описания



✓ Описание (конкретизация) на твърдение с използване на RDF речника се извършва чрез: 1/1

- ☐ типа rdf:Consideration и свойствата rdf:subject, rdf:predicate и rdf:object
- ☐ типа rdf:Reification и свойствата rdf:subject, rdf:predicate и rdf:object
- ☐ използване на други средства
- ☒ типа rdf:Statement и свойствата rdf:subject, rdf:predicate и rdf:object ✓
- ☐ типа rdf:Concretisation и свойствата rdf:subject, rdf:predicate и rdf:object

✓ Всяко IDREF, декларирано в DTD, трябва да 1/1

- ☒ избере дадено определено ID ✓
- ☐ избере кое да е ID
- ☐ избере множество ID-та

✓ Изберете верното: 1/1

- ☐ Един URI не може да бъде нито URL, нито URN
- ☐ Един URL може да бъде или URI, или URN
- ☐ Един URI може да бъде URL и URN едновременно
- ☐ Един URN може да бъде или URL, или URI
- ☒ Един URI може да бъде или URL, или URN ✓



✓ XML пространствата от имена се дефинират чрез атрибути и могат да бъдат специфицирани в кой да е XML елемент. 1/1

- ☒ Вярно
- ☐ Невярно



✓ Кое от показаните на фигурата средства не е пряко свързано с изграждане на семантичния Уеб: 1/1

- ☐ Ontologies
- ☐ Logical Support
- ☐ Tools
- ☐ Semantic Annotations
- ☐ Applications / Services
- ☒ Collaboration synchronization



✓ XML пространствата от имена се използват за разграничаване: 1/1

- ☐ нито на XML елементи, нито на XML атрибути
- ☒ и на XML елементи, и на XML атрибути
- ☐ само на XML елементи
- ☐ само на XML атрибути



✓ RDF поддържа описания на групи, съдържащи само определени членове, посредством: 1/1

- ☐ RDF вектори
- ☒ RDF колекции ✓
- ☐ RDF масиви
- ☐ RDF enumeration
- ☐ нито един от изброените начини

✓ Регулярният израз `[1-9]?[0-9]` дефинира XSchema стрингови стойности, които са: 1/1

- ☐ от 1 до 9
- ☒ от 0 до 99 ✓
- ☐ от 1 до 99
- ☐ от 1 до 90

✓ Ако в XSchema дефинираме exact тип, то: 1/1

- ☐ той може да има дериватни типове, но те могат да бъдат използвани в XML документа вместо този exact тип само при определени условия
- ☐ той не може да има дериватни типове
- ☒ той може да има дериватни типове, но те не могат да бъдат използвани в XML документа вместо този exact тип ✓
- ☐ той може да има дериватни типове, и те могат да бъдат използвани в XML документа вместо този exact тип



✓ "Формална, експлицитна спецификация на споделена концептуализация" е дефиниция за:

1/1

- ☒ нещо друго, което не присъства в тези отговори
- ☐ валиден XML документ
- ☐ валиден RDFS документ
- ☐ валиден RDF/XML документ
- ☐ валиден XML документ, представящ онтология в даден жанр



✓ В XML Schema дефиницията за елементно съдържание ... order="all" ... е еквивалентна на ANY в DTD.

1/1

- ☐ верно
- ☒ неверно



✓ В RDF един екземпляр може да има няколко типа (т.е. да участва като субект в няколко релации rdf:type).

1/1

- ☐ Неверно
- ☒ Верно



✓ В XSchema, за глобалните типове:

1/1

- ☒ minOccurs и maxOccurs може да се дефинират само за глобални типове, които се реферират ✓
- ☐ minOccurs и maxOccurs не може да се дефинират
- ☐ minOccurs и maxOccurs може да се дефинират винаги
- ☐ minOccurs и maxOccurs може да се дефинират само за глобални типове, които не се реферират

✓ Ако даден атрибут на XML елемент има само локално име (т.е. името му не е QName), то:

1/1

- ☐ Той не се отнася към каквото и да било пространство от имена за документа, в който е дефиниран.
- ☐ Той се отнася към пространството от имена по подразбиране (default namespace).
- ☐ Той не се отнася към пространството от имена на елемента, към който принадлежи.
- ☒ Той се отнася неявно към пространството от имена на елемента, към който принадлежи. ✓

✓ XML документите могат да съдържат само един елемент-корен

1/1

- ☒ Верно ✓
- ☐ Неверно



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

- ☐ Вярно
- ☒ Невярно



✓ На показаната фигура на RDF граф, T-shirt е: 1/1

- ☐ Обект
- ☐ Предикат
- ☐ дериват
- ☒ субект



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

- ☐ активира nereкурсивно обработката на всички наследници на елемента, за който се отнася
- ☒ активира рекурсивно обработката на всички наследници на елемента, за който се отнася



✓ В XLink, входящите дъги (inbound arcs) могат да имат: 1/1

- ☒ произволен краен брой участващи в тях ресурси
- ☐ точно два участващи в тях ресурса



✓ Една XLink дъга (arc), която има локален стартов ресурс и отдалечен 1/1 краен ресурс, се нарича:

- ☒ Outbound
- ☐ сочеща трети (third-party) ресурс
- ☐ inbound



✓ Йерархичната подредба на понятия заедно с информация за допълнителни отношения като по-широко/по-тясно понятие, синоними, еквивалентност и др., задава: 1/1

- ☐ Таксономия
- ☐ Списък от термини
- ☐ Пръстен от синоними
- ☒ Тезаурус
- ☐ Онтология



✓ В XSchema, локалните типове 1/1

- ☐ винаги са директни наследници на корена
- ☐ могат да бъдат директни наследници на корена
- ☐ могат да бъдат или да не бъдат директни наследници на корена
- ☒ не могат да бъдат директни наследници на корена



✓ В примера по-долу: `<x xmlns:n1="http://www.hayde-de.org" xmlns="http://www.hayde-de.org" > <ok a="1" b="2" /> <hmmm a="1" n1:a="2" /> </x>` 1/1

- ☐ Дефиницията на елемента `<hmmm>` е невалидна според спецификацията на пространствата от имена.
- ☒ Дефиницията на елемента `<hmmm>` е валидна според спецификацията на пространствата от имена. ✓

✓ XLink използва отдалечени (remote) ресурси, за да представя: 1/1

- ☐ нито един от останалите отговори не е верен
- ☐ само външни за документа ресурси, но не и вътрешни ресурси, достъпни по URI
- ☐ само вътрешни за документа ресурси, достъпни по URI, но не и външни ресурси
- ☐ Нито външни за документа ресурси, нито вътрешни ресурси, достъпни по URI
- ☒ Както външни за документа ресурси, така и вътрешни ресурси, достъпни по URI ✓

✓ Описанието (конкретизацията) на RDF твърдение чрез използване на 1/1 RDF речника се нарича:

- ☒ reification ✓
- ☐ concretization
- ☐ objectification
- ☐ consideration



✓ RDF литералите могат да се явяват както като субект, така и като обект в RDF тройка 1/1

- ☐ Вярно
- ☒ Невярно



✓ Елементите contributor, creator, publisher и rights са част от: (**има подобен въпрос, но с други възможни отговори**) 1/1

- ☐ DCQ пространството (Dublin Core квалификатори)
- ☐ DCM пространството (Dublin Core метаданни)
- ☒ DC пространството (Dublin Core ядро)
- ☐ DCT пространството (Dublin Core термини)



✓ В XSchema, типът на всеки елемент на еквивалентен клас трябва да бъде: 1/1

- ☐ същият като типа на елемента-екземпляр
- ☐ различен от типа на елемента-екземпляр
- ☐ дериват на типа на елемента-екземпляр
- ☒ същият като типа на елемента-екземпляр, или негов дериват



✓ DOMException връща HIERARCHY_REQUEST_ERR при опит за:

1/1

- ☐ Вмъкване на невалиден възел в йерархията на DOM дървото
- ☐ заявка за получаване на йерархията на възел с дълбочина, по-голяма от съществуващата на възела
- ☒ Вмъкване на възел на неподходящо място в йерархията на DOM дървото ✓
- ☐ Заявка за получаване на йерархията на елемент без наследници
- ☐ Заявка за получаване на йерархията на атрибут

✓ В SAX, приложните обекти, имащи достъп до XML сорса:

1/1

- ☐ са регистрирани за callback функции или от програмиста, или от парсера
- ☐ 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 ✓



✓ Кой от отговорите по-долу не съдържа име на RDFS свойство:

1/1

- ☐ rdfs:isDefinedBy
- ☐ Rdfs:range
- ☐ rdfs:comment
- ☒ rdfs:subRange
- ☐ Rdfs:label
- ☐ rdfs:subPropertyOf
- ☐ Rdfs:member
- ☐ rdfs:subClassOf



✓ Какво не е възможно да бъде описано в DTD:

1/1

- ☐ къде типът на елементите може да се среща
- ☐ какви имена могат да бъдат използвани за тип на елементите
- ☐ йерархия на документа и грануларност
- ☐ имена и типове на атрибутите на елементите
- ☒ минимален и максимален брой срещания на елемент



✓ MEDIA атрибутите като screen, aural, braille, tty, и т.н., са част от: 1/1

- ☐ само от CSS2 спецификацията
- ☐ само от CSS3 спецификацията
- ☐ CSS1 и CSS2 спецификациите
- ☒ CSS2 и CSS3 спецификациите
- ☐ само от CSS1 спецификацията



✓ С един DOM Element обект: 1/1

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



✓ Атрибутът xml:base задава: 1/1

- ☒ база на относителни URI връзки към външни за документа ресурси
- ☐ база за задаване на 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, ние трябва да използваме 1/1

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

✓ Кое е името на подразбиращото се (default) пространство от имена в 1/1 декларацията подолу: <?xml version="1.0" encoding="ISO-8859-1"?>
<pers:person xmlns:pers="http://sernaferna.com/pers" xmlns:html="http://www.w3.org/1999/xhtml">

- ☐ и pers, и html
- ☐ Html
- ☐ Pers
- ☐ зависи от XML парсера
- ☒ няма подразбиращото се (default) пространство от имена ✓

✓ Редът <!ATTLIST point honorific (Mr|Ms|Mrs|Rev|Dr) ... > е валиден DTD 1/1 пример за:

- ☒ name group ✓
- ☐ CDATA
- ☐ NMTOKENS
- ☐ notation



✓ В RDF, група от алтернативни стойности се задава чрез:

1/1

- ☐ <rdf:Opt>
- ☐ <rdf:Switch>
- ☐ <rdf:Choice>
- ☐ <rdf:Case>
- ☒ <rdf:Alt>



✓ Записът @prefix rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>. @prefix contact: <<http://www.w3.org/2000/10/swap/pim/contact#>>. <<http://www.w3.org/People/EM/contact#me>> rdf:type contact:Person; contact:fullName "Eric Miller"; contact:mailbox <mailto:em@w3.org>; contact:personalTitle "Dr.". представя по-долния RDF граф в:

1/1

- ☐ RDF/XML
- ☒ Turtle
- ☐ RDFa
- ☐ N3



✓ Следният XML документ е добре конструиран (well-formed): <?xml version='1.0'?> <!--<http://www.w3.org> is bound to both ref1 and default namespace--><myDoc xmlns:ref1="<http://www.w3.org>" xmlns="<http://www.w3.org>" xmlns:ref2="<http://www.hmmm.bg>"> <myElem a="1" b="2" /> <myElem a="1" ref1:a="2" /> <myElem ref1:a="1" ref2:a="2" /> </myDoc>

1/1

- ☒ Истина
- ☐ Неистина



✓ 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"/>`

✓ DTD валидацията ограничава елементите и атрибутите, които могат да участват в документа, само по техните локални имена, ако в DTD дефиницията на елемента или атрибута той е участвал с локално име вместо с QName. 1/1

- ☒ Вярно ✓
- ☐ Невярно

✓ За представяне на стойности в XML файлове, ние можем да използваме XML атрибути или XML елементи. Кой от следващите отговори не е аргумент за употреба на атрибути вместо на елементи? 1/1

- ☐ използваме атрибут, когато той е свойство на елемента
- ☐ използваме атрибути за валидация на прост (simple) тип данни
- ☒ използваме атрибути за валидация на сложна структура ✓
- ☐ използваме атрибут, когато информацията е присъща за елемента, а не за под-елементите



✓ Кой от отговорите по-долу не съдържа име на RDFS клас:

1/1

- ☐ rdfs:Resource
- ☐ rdfs:Datatype
- ☐ rdfs:ContainerMembershipProperty
- ☐ rdfs:Class
- ☐ rdfs:Container
- ☒ rdfs:Property



✓ В OWL, също както и в ООП, не може да съществуват два класа с общи екземпляри.

1/1

- ☐ Верно
- ☒ Неверно



✓ Ако няма специфицирано кодиране за един XML документ и документът не е нито в UTF-8, нито в UTF-16, тогава резултатът е:

1/1

- ☐ ISO 8859-1
- ☒ ERROR
- ☐ UTF-8
- ☐ UTF-16



✓ Атрибутите, специфицирани в DTD посредством ключовата дума IMPLIED: 1/1

- ☐ имат само една допустима стойност
- ☒ могат да бъдат определени (специфицирани) в XML ✓
- ☐ трябва да бъдат определени (специфицирани) в XML
- ☐ имат стойност по подразбиране, ако не са определени (специфицирани) в XML

✓ Регулярният израз [[^]0-9]x дефинира XSchema стрингови стойности, 1/1
които представляват:

- ☒ който и да е не-цифров символ, следван от символа x ✓
- ☐ символът [^], последван от произволна цифра следвана от символа x
- ☐ x пъти повторение на който и да е цифров символ
- ☐ x пъти повторение на който и да е не-цифров символ

✓ Един XML документ може да има повече от едно вътрешно DTD. 1/1

- ☒ Неверно ✓
- ☐ Верно



✓ В какъв формат е следният запис: `<http://www.w3.org/TR/rdf-syntax-grammar> <http://xml.org/1.1/title> "XML Basics" .`
`<http://www.w3.org/TR/rdf-syntax-grammar>`
`<http://example.org/stuff/xml> _:bnode . _:bnode`
`<http://example.org/stuff/1.0/fullname> "D. D. Vass" . _:bnode`
`<http://example.org/stuff/1.0/homePage> <http://www.aemon.net/> .` 1/1

- ☐ друг формат, различен от останалите
- ☐ RDF/XML
- ☐ N3
- ☒ N-Triples
- ☐ Durtle



✓ Методите за работа с атрибути на DOM Element интерфейса: 1/1

- ☐ Разграничават стойността на атрибут дефинирана изрично в XML документа, от стойността по подразбиране, зададена в DTD, само ако Node:setAttrDistinction(Boolean param) е извикан предварително с параметър param със стойност True
- ☒ не могат да разграничават стойността на атрибут, дефиниран изрично в XML документа, от стойността по подразбиране, зададена в DTD
- ☐ могат да разграничават стойността на атрибут, дефинирана изрично в XML документа, от стойността по подразбиране, зададена в DTD
- ☐ зависи от настройките на DOM парсера



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

- ☒ Истина
- ☐ Неистина



✓ 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

✓ Единственото изискване един XML документ да бъде валиден е този документ да е структуриран (формиран) съгласно правилата за XML синтаксиса. 1/1

- ☐ Верно
- ☒ неверно ✓

✓ RDF графът от фигурата задава: 1/1

- ☐ Лекцията е посетена от Джон, Мери или Крис.
- ☐ Лекцията е посетена от Джон, Мери и Крис - в този ред.
- ☒ Лекцията е посетена от Джон, Мери и Крис. ✓
- ☐ Лекцията е посетена от Джон, Мери, Крис и от никой друг.



✓ Описанието `ex:adrian foaf:knows ex:ger ex:adrian foaf:age "41"^^xs:int.` 1/1
представя RDF Triples в:

- ☐ RDF/XML формат.
- ☒ N3 формат.
- ☐ Turtle формат.
- ☐ RDFa формат.



✓ XML йерархията `<a> <c> </c> ` е: 1/1

- ☐ Коректна или некоректна в зависимост от DTD/XSchema валидацията
- ☐ Нито един от останалите отговори не е верен
- ☒ Некоректна
- ☐ Коректна



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

- ☐ с името "name"
- ☐ с имената на елементите в изходящото дърво
- ☒ със съдържанието на елементите в изходящото дърво
- ☐ със съдържанието на атрибутите в изходящото дърво
- ☐ с имената на атрибутите в изходящото дърво



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

- ☒ `NamedNodeMap`
- ☐ `NodeList`
- ☐ `Attr`
- ☐ `Text`



✓ Resource Description Framework (RDF) представя информация за ресурси, които: 1/1

- ☐ не могат да бъдат достъпни в Уеб
- ☐ трябва да бъдат достъпни в Уеб
- ☒ могат да бъдат или да не бъдат достъпни в Уеб



✓ В CSS3, фиксираната (`fixed`) схема за позициониране: 1/1

- ☒ предизвиква повторение на кутията на всяка страница при `paged media`
- ☐ се характеризира със задаване на явно отместване (`explicit offset`) спрямо съдържащия блок
- ☐ предефинира `top` и `left` позициите на кутията
- ☐ се контролира от браузера с цел по-бързо показване на съдържанието



✓ Кои от изброените по-долу качества НЕ СА ПРИСЪЩИ на спецификация на онтология:

1/1

☐ Споделена

☒ Неявна



☐ Концептуална

☐ Машинно-четима

☒ Неформална



✓ При включването на документа "хpto.xml" с използване на `<xi:include href="хpto.xml" parse="text"/>`, документът "хpto.xml" ще бъде включен:

1/1

☐ като XML документ, но само ако той е добре конструиран

☐ винаги като XML документ

☐ като XML документ, но само ако той е добре конструиран и валиден спрямо схема или DTD

☒ като обикновен текст



✓ Описанието в RDF на група, съдържаща само зададените в описанието членове, става чрез използване на:

1/1

☐ Множество

☒ Колекция



☐ Алтернатива

☐ Bag

☐ последователност



✓ В SAX 2, управляваме събитията относно съдържанието на документа чрез:

1/1

- ☒ имплементиране на ContentHandler или разширяване на DefaultHandler ✓
- ☐ имплементиране на EventHandler или разширяване на BaseHandler //wrong
- ☐ имплементиране на HandlerBase или разширяване на DocumentHandler
- ☐ имплементиране на DocumentHandler или разширяване на HandlerBase
- ☐ имплементиране на DefaultHandler или разширяване на ContentHandler

✓ XSchema шаблонен фасет (pattern facet) е фасет за данни от тип:

1/1

- ☒ String ✓
- ☐ Integer
- ☐ Binary
- ☐ decimal



✗ XPath изразът `./book[author/last="пробен изпит"]` връща:

0/1

- ☐ Всички book елементи, които имат елемент author с атрибут last равен на "пробен изпит"
- ☐ Елемент last със стойност "пробен изпит", който има 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 author с поделемент last равен на "пробен изпит" ✗

Правилен отговор

- ☒ Всички book елементи, които имат елемент author с поделемент last равен на "пробен изпит"

✓ RDF реификация се задава чрез:

1/1

- ☐ Типа `rdf:Reification` и свойствата `rdf:about`, `rdf:property` и `rdf:object`
- ☒ Типа `rdf:Statement` и свойствата `rdf:subject`, `rdf:predicate` и `rdf:object` ✓
- ☐ Типа `rdf:Description` и свойствата `rdf:subject`, `rdf:predicate` и `rdf:object`
- ☐ Типа `rdf:Reification` и свойствата `rdf:subject`, `rdf:predicate` и `rdf:object`
- ☐ Типа `rdf:Statement` и свойствата `rdf:about`, `rdf:property` и `rdf:object`
- ☐ Типа `rdf:Description` и свойствата `rdf:about`, `rdf:property` и `rdf:object`



✓ Символите <, > и ' са валидни PCDATA символи

1/1

☒ Невярно



☐ Вярно

✓ В DOM, възли-деца могат да имат единствено възлите от тип:

1/1

☐ Node, Element и Attribute

☐ Root, Parent, и Ancestor

☒ Document, DocumentFragment и Element



☐ CharacterData, Text и Comment

☐ Notation, Entity, EntityReference

✓ Кой от следните типове не се среща в описанието на разширена връзка в XLink:

1/1

☐ Title

☐ Locator

☐ Arc

☒ Edge



☐ resource



✓ **Общо текстово Entity:**

1/1

- ☐ Не може да бъде използвано рекурсивно и не може да се появи в съдържанието на елемент и/или в стойност на атрибут
- ☐ Може да бъде използвано рекурсивно, но не може да се появи в съдържанието на елемент и/или в стойност на атрибут
- ☐ Може да бъде използвано рекурсивно и може да се появи в съдържанието на елемент и/или в стойност на атрибут
- ☐ Може да бъде използвано рекурсивно и може да се появи в съдържанието на елемент, но не и в стойност на атрибут
- ☒ Не може да бъде използвано рекурсивно, но може да се появи в съдържанието на елемент и/или в стойност на атрибут ✓

✓ **XML Schema разрешава да се прави разлика между уникална стойност (unique) и ключ (key).**

1/1

- ☒ Верно ✓
- ☐ неверно

✓ **В XML схема, уникалността на key и keyref елементите е:**

1/1

- ☐ винаги за целия документ
- ☒ само за йерархията на съответните елементи в документа-екземпляр, за които са дефинирани key и keyref ✓
- ☐ според стойността на атрибута "context"
- ☐ според стойността на атрибута "range"



✓ При използване на вътрешна и външна DTD дефиниция, вътрешната DTD може да предефинира: 1/1

- ☐ само ENTITY и NOTATION на външната дефиниция
- ☐ само ATTLIST и NOTATION на външната дефиниция
- ☒ само ENTITY и ATTLIST на външната дефиниция ✓
- ☐ всички дефиниции на външната DTD
- ☐ само ENTITY, ATTLIST и NOTATION на външната дефиниция

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

- ☐ <xsl:process-templates>
- ☐ <xsl:value-of>
- ☐ <xsl:template>
- ☐ <xsl:for-each>
- ☒ <xsl:apply-templates> ✓



✓ Ако имаме дефинирани две еднакви CSS свойства с различни стойности за един и същи елемент, то ще се приложи последнодефинираното свойство: 1/1

- ☐ винаги
- ☐ винаги, освен когато правилото за прилагане на едно от свойствата взима под внимание кой е предходният елемент
- ☐ винаги, освен когато едно от свойствата е по-специфично или правилото за прилагането му взима под внимание кой е предходният елемент
- ☒ винаги, освен когато едно от свойствата е по-специфично ✓

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

- ☐ Вярно
- ☒ Невярно ✓

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

- ☐ Вярно
- ☒ Невярно ✓

✓ Един XML документ може да има повече от едно DTD. 1/1

- ☒ Верно ✓
- ☐ неверно



✓ RDF твърдението "dbpedia:Mount_Etna rdf:type my-pref:Mountain, my-pref:Volcano" е допустимо и валидно. 1/1

☒ Верно



☐ Неверно

✓ Person, title, familyName, knows, age, Document и Organization са име на елементи, дефинирани от: 1/1

☐ OWL

☒ FOAF



☐ DC

☐ RDFS

✓ Кой два от изброените по-долу атрибути НЕ МОГАТ да участват в елемент от XLink тип arc: 1/1

☒ Role



☐ From

☐ Show

☒ Href



☐ Title

☐ Type

☐ To

☐ Actuate



✓ Как можем да потиснем запазването на допълнителни (следващи един след друг) празни интервали в съдържанието на текстови елемент? 1/1

- ☐ посредством псевдо-атрибута xml:lang
- ☐ посредством псевдо-атрибута xml:interval
- ☐ посредством псевдо-атрибута xml:blank
- ☒ посредством използване на друг псевдо-атрибут, различен от изброените в останалите отговори на въпроса ✓
- ☐ посредством псевдо-атрибута xml:empty_space
- ☐ запазването на допълнителни (следващи един след друг) празни интервали в съдържанието на текстови елемент не може да се потиска

✓ SAX Element обектите: 1/1

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



✓ Разгледайте имената на XML празните елементи `<emptyTag />`; и `<emptyTag/ >`. Те са: 1/1

- ☐ и двата валидни
- ☒ първият е валиден, вторият е невалиден ✓
- ☐ нито едно от по-горе посочените
- ☐ първият е невалиден, вторият е валиден
- ☐ и двата невалидни

✓ В RDF, класовете могат да бъдат екземпляри на други класове. 1/1

- ☐ Неверно
- ☒ Верно ✓

✓ Използването на пространства от имена в XPointer изрази е: 1/1

- ☒ Разрешено ✓
- ☐ Забранено



✓ Кой от следните изрази е валиден пример за дефиниране на последователност (sequence) от елементи в DTD:

1/1

- ☐ (A and B and C)
- ☐ (A ^ B ^ C)
- ☒ (A, B, C)
- ☐ (A | B | C)



✓ В RDF твърдението "S rdf:type O"

1/1

- ☐ S задава клас (категория), а O задава екземпляр на този клас
- ☒ O задава клас (категория), а S задава екземпляр на този клас
- ☐ S и O задават класове на екземпляри
- ☐ S и O задават екземпляри на класове



✓ В RDFS, Property е подмножество на RDFS Resources и има за домейн 1/1 (rdfs:domain):

- ☐ типа на стойностите на това Property
- ☐ класа rdfs:Resource, на който принадлежат всички екземпляри на ресурса
- ☐ множество от отделни стойности на това Property
- ☒ класа, асоцииран с това Property



✓ “other” и “none” са възможни стойности за XLink атрибутите:

1/1

- ☐ arcrole и role
- ☒ actuate и show
- ☐ actuate и arcrole
- ☐ how и role
- ☐ show и arcrole



✓ creator е:

1/1

- ☐ елемент-деквалификатор от DC Metadata
- ☒ един от основните 15 елемента от DC Metadata
- ☐ елемент, който не принадлежи на DC Metadata
- ☐ елемент-квалификатор от DC Metadata



✓ Даден CSS стил може да бъде inlined (а не embedded) в документ посредством

1/1

- ☐ елемента INLINE
- ☐ атрибута INLINE
- ☒ атрибута STYLE
- ☐ елемента STYLE



✓ XML пространството от имена се отнася:

1/1

- ☐ Само за елемента, за който е дефинирано
- ☒ Само за йерархията от елементи, за която е дефинирано ✓
- ☐ Само за елемента, за който е дефинирано, както и за неговите атрибути
- ☐ За всички оставащи от мястото на дефинирането му до края на документа елементи

✓ В OWL, един екземпляр може да бъде

1/1

- ☒ член на множество класове ✓
- ☐ единствено член на само един клас

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

1/1

- ☐ StAX Direct Mapping API
- ☐ iterator API
- ☒ cursor API ✓
- ☐ StAX Events API



✓ Според XML синтаксиса, за всеки непразен XML елемент, за отварящия му таг (маркер) трябва да има:

1/1

- ☐ Един или повече затварящи тагове
- ☐ Един затварящ или незатварящ таг
- ☒ Точно един затварящ таг
- ☐ Повече от един затварящи тагове



✓ В CSS3, абсолютната (absolute) схема за позициониране:

1/1

- ☐ задава липса на движение на кутията при движение на документа за continuous media
- ☒ се характеризира със задаване на явно отместване (explicit offset) спрямо съдържащия блок
- ☐ предефинира top и left позициите на кутията
- ☐ предизвиква повторение на кутията на всяка страница при paged media



✓ Resource Description Framework (RDF) служи за описания, предназначени:

1/1

- ☐ за потребителско визуализиране на метаданни
- ☐ както за обработка от софтуерни приложения, така и за потребителско визуализиране на метаданни
- ☐ нито за обработка от софтуерни приложения, нито за потребителско визуализиране на метаданни
- ☒ за обработка от софтуерни приложения



✓ В CSS, задаването на елемент, чийто атрибут attr завършва с дадена стойност xpto, става чрез:

- ☒ [attr\$=xpto]
- ☐ [attr|=xpto]
- ☐ [attr^=xpto]
- ☐ [attr=xpto]
- ☐ [attr~=xpto]



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

- ☐ SAX
- ☒ DOM
- ☐ StAX API
- ☐ XSLT
- ☐ CSS



✓ В XLink, изходящите връзки (outbound links) могат да имат: 1/1

- ☐ не по-малко от два участващи ресурса
- ☒ точно два участващи в тях ресурса
- ☐ произволен краен брой участващи в тях ресурси



✓ Дефинирането на референция (Reference) към ключ (Key) в XSD може да има полета, които са: 1/1

- ☐ нито атрибути, нито елементи
- ☐ само елементи
- ☐ само атрибути
- ☐ както атрибути, така и елементи, като всеки един атрибут (или елемент) трябва да съответства на атрибут (или на елемент) в ключа
- ☒ както атрибути, така и елементи, като всички полета трябва да съответстват на типа и на позицията на тези в ключа ✓

✓ 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

✓ Binary entity може да се използва само като атрибут от тип ENTITY. 1/1

- ☐ Неверно
- ☒ Верно ✓



✓ XML Reader в SAX 2.0 разширява стандартния Java Reader интерфейс:

1/1

- ☐ Вярно
- ☒ Невярно



✓ 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



✓ Квалифицирано XML име (наречено QName) е:

1/1

- ☒ име от вида `namespace_prefix:local_name`
- ☐ всяко име на XML елемент
- ☐ всяко валидно XML име
- ☐ всяко XML име на пространство от имена



✓ Свойството `rdfs:isDefinedBy` е специален (под)тип на:

1/1

- ☐ `rdfs:is`
- ☒ `rdfs:seeAlso`
- ☐ `rdfs:isCreatedBy`
- ☐ `rdfs:isDescribedBy`
- ☐ `rdfs:isDefined`



✓ Създаване на специфична метаинформация и схема за употреба с цел предоставяне на възможност за нови методи за достъп до информация се нарича:

2/2

- ☒ Семантично аотиране
- ☐ Семантично разглеждане
- ☐ Семантично препоръчване
- ☐ Автоматично предсказване на потребителски код
- ☐ Семантично визуализиране



✓ Задаването на мрежа от хора чрез FOAF твърдения става посредством релацията:

1/1

- ☐ `foaf:linkedTo`
- ☐ `foaf:relatedTo`
- ☐ нито една от изброените
- ☐ `foaf:follows`
- ☒ `foaf:knows`



✓ OWL клас, за който са описани както задължителните, така и достатъчните свойства за членство в класа, се нарича:

1/1

- ☐ disjoint клас
- ☐ примитивен клас
- ☐ анонимен клас
- ☒ дефиниран клас



✓ В RDFS, Property е подмножество на RDFS Resources и има за обхват (rdfs:range) 1/1

- ☐ Типа на стойностите на това Property
- ☒ Множество от отделни стойности на това Property
- ☐ Класа rdfs:Resource, на който принадлежат всички екземпляри на ресурса
- ☐ Класа, асоцииран с това Property



✓ В XSchema, референцията към тип може да реферира:

1/1

- ☐ както към локални, така и към глобални типове
- ☒ само към глобални типове
- ☐ нито към локални, нито към глобални типове
- ☐ само към локални типове



✓ Осъществяването на embedding (а не inlining) на Style Sheet според CSS правилата е възможно посредством: 1/1

- ☐ елемента EMBED
- ☒ елемента STYLE
- ☐ елемента LINK
- ☐ атрибута STYLE



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

- ☐ DOM 1.0 разрешава редактиране на този възел
- ☒ DOM 1.0 не разрешава редактиране на този възел



✓ В RDFS един ресурс може да е екземпляр на няколко класа. 1/1

- ☒ Вярно
- ☐ Невярно



✓ В XSchema, глобалните типове: 1/1

- ☐ могат да не бъдат директни наследници на корена
- ☐ могат да бъдат или да не бъдат директни наследници на корена
- ☒ винаги са директни наследници на корена
- ☐ не могат да бъдат директни наследници на корена



✓ Уникална идентификация на пространство от имена се реализира с: 1/1

☐ URN

☒ URL



☐ URI (и URL, и URN)

✓ XLink не може да дефинира фрагментни идентификатори за URI, сочещи към възли или части от тях в XML ресурси. 1/1

☒ верно



☐ неверно

✓ Елементите от прост тип в една XML Schema са от даден предефиниран тип и: 1/1

☒ не могат да имат атрибути и под-елементи



☐ могат да имат под-елементи, но не и атрибути

☐ могат да имат атрибути и под-елементи

☐ могат да имат атрибути, но не и под-елементи

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

☐ Верно

☒ Неверно



✓ Множественното наследяване (multiple inheritance) не е допустимо в RDFS. 1/1

☒ Неверно



☐ Верно

✓ Всеки DOM възел (Node) може да има деца. 1/1

☐ Верно

☐ зависи от децата

☒ неверно



✓ XQUERY изразът For \$o in catalog/cd Let \$l := \$o/tracklist Where \$l/@num="1" Return \$l/track Избира: 1/1

☐ Стойността на първия елемент track, който е под-елемент на tracklist, имащ атрибут num равен на 1 и който е наследник на елемент cd

☐ стойността на всички елементи track, които са под-елементи на tracklist, наследник на елемент cd с атрибут равен на 1

☐ Нито един от останалите въпроси не е верен

☒ стойността на всички елементи track, които са под-елементи на tracklist, имащ атрибут num равен на 1 и който е наследник на елемент cd



☐ Стойността на първия елемент track, който е под-елемент на tracklist, наследник на елемент cd с атрибут num равен на 1



✓ Annotation Property в OWL се използва за добавяне на метаданни: 1/1

- ☐ само към свойства (properties)
- ☒ към класове, екземпляри и свойства
- ☐ към класове и към екземпляри
- ☐ само към класове
- ☐ към класове и към свойства
- ☐ към екземпляри и към свойства
- ☐ само към екземпляри



✓ При автоматичното разпознаване на анотации, отношението броя на 1/1 правилно разпознатите анотации към броя на всички съществуващи правилни анотации се нарича:

- ☐ пълнотата на връщане (recall)
- ☐ устойчивост (стабилност)
- ☐ достоверност на резултата (reliability)
- ☒ точност (precision)



Това съдържание не е нито създадено, нито одобрено от Google. - [Условия за ползване](#) - [Декларация за поверителност](#)

Google

