

1. Най-общо маркър езиките включват:

-структурни(layout), функционални(action), семантични(meaning)и стилистични(appearance) описания

2.XML пространството от имена се отнася:

-само за йерархията от елементи (и атрибутите им), за която е дефинирано

3.xml:id е псевдо-атрибут, който задава условие за уникална стойност, като това:

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

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

-елементите book-преки наследници на текущия елемент, които имат под-елемент author с под-елемент last, равен на "пробен изпит"

5.Изберете верния отговор:

-Всеки валиден XML документ е добре конструиран, но обратното не е задължително вярно

6.Кой от следните елементи не е с грешен XML синтаксис:

-<birth>28...</birth >

7.Разгледайте изброените по-долу символи:

(& и < са грешните)

-Между изброените символи има точно два недопустими в PCDATA елементно съдържание

8.Елементите key/unique задават ключ/уникалност в XML Schema, като имат за обхват (т.е. област на действие):

-Само XML елементите, в чиято декларация е зададен key или unique

9.Параметричните единици (Parameter Entities) са декларирани в DTD и се използват:

-единствено в декларациите на DTD

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

-като обикновен текст

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

-вярно

13. При използване на вътрешна и външна DTD дефиниция, вътрешната може да предефинира:

-само ENTITY и ATTLIST на външната дефиниция

14. В XSchema, референцията към тип може да реферира:

Само към глобални типове

15. В примера по-долу:

```
<x xmlns:n1="http...  
  xmlns="http...">  
  <ok a="1" b="2" />  
  <hmmm a="1" n1:a="2" />  
</x>
```

-Дефиницията на елемента <hmmm> е валидна според спецификацията на пространствата от имена

17. При избиране на елемент, наречен MyElem и имащ атрибут Attr със стойност title, в XSLT ние трябва да използваме:

-select="MyElem[@Attr='title']"

18. Атрибутите, специфицирани в DTD посредством ключовата дума IMPLIED:

-могат да бъдат определени (специфицирани) в XML

19. Binary entity може да се използва само като атрибут от тип ENTITY

-Вярно

20. В XML Schema, ние можем да построим производни (derived) types:

- както с extensions, така и с restrictions

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

- първият е валиден, вторият е невалиден

23. `xml:id` е псевдо-атрибут, който задава условие за уникална стойност, като това:

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

24. В XSchema, глобалните типове:

- винаги са директни наследници на корена

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

- ERROR

27. XPath изразът `//book[@pages]` връща

- всички book елементи, които имат атрибут pages

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

- невярно

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

- невярно

52. Редът `<!ATTLIST point honorific(Mr|Ms|Mrs|Rev|DR) ...>` е валиден DTD пример за:

- name group

53. В XML схема, уникалността на key и keyref елементите е:

-само за йерархията на съответните елементи в документа-екземпляр, за които са дефинирани key и keyref

54. Ако сме дефинирали XSL променлива като

```
<xsl:variable name="price">low</xsl:variable>
```

то тя може да се използва в XSL елемент като:

```
<xsl:value-of select="$price"/>
```

59. В XSchema, типът на всеки елемент на еквивалентен клас трябва да бъде:

-същият като типа на елемента-екземпляр, или негов дериват

60. Ако в XSchema дефинираме exact тип, то:

-той може да има дериватни типове, но не могат да бъдат използвани в XML документа вместо този exact тип

62. Общо текстово Entity:

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

64. Какво HE е възможно да бъде описано в DTD

-минимален и максимален брой срещания на елемент

65. XSL кодът, показан на фигурата, ще даде като резултат:

```
<xsl:template match="name...">
  <xsl:element name="{...}">
    My own contents!
  </xsl:element>
</xsl:template>
```

-изходните XML елементи ще бъдат с имена, еднакви със съдържанието на елементи от сорс дървото и със съдържание "My own contents!"

67. Кое е името на подразбиращото се (default) пространство от имена в декларацията по долу:

<?xml version=...

<pers:person xmlns:pers=<http://smth>
xmlns:html="https://else

- няма подразбиращо се (default) пространство от имена

68. Речник с документиран процес за актуализация се нарича:

- контролиран

76. Разгледайте имената на XML елементите <xml-tag> и <tag-xml>. Те са:

- първият е невалиден, вторият е валиден

77. Един XML документ може да има повече от едно DTD

- вярно

79. Кой от следите изрази е валиден пример за дефиниране на избор (choice) от елементи в DTD:

(A | B | C)

80. Всяко IDREF, декларирано в DTD, трябва да:

- избере дадено определено ID

82. Изберете вярното:

- Един URI може да бъде или URL, или URN

84. 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?

- използваме атрибути за валидация на сложна структура

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

- вярно

88. Уникална идентификация на пространството от имена се реализира с:

-URI

90. Регулярният израз $[^0-9]^*$ дефинира XSchema стрингови стойности, които представляват:

-който и да е НЕцифров символ, следван от символа x

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

(A, B, C)

92. Регулярният израз $[1-9]?[0-9]^*$ дефинира XSchema стрингови стойности, които са:

- от 0 до 99

93. XML пространствата от имена се използват за разграничаване:

-и на XML елементи, и на XML атрибути

94. Разгледайте `<xsl:value-of>` елемента. Ако стойността на неговия select атрибут е `select='.'`, тогава ние избираме:

-текстовия контекст на елемента и текстовия контекст на всички наследници на елемента

95. В XSchema, локалните типове:

-не могат да бъдат директни наследници на корена

96. Един XML документ може да има повече от едно вътрешно DTD

-невярно

97. При използването на XPath `text()` функцията, ние избираме

-текстовия контекст само на елемента

98. CDATA съдържанията са:

-игнорирани от XML парсерите

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

-string

106.Параметрично Entity:

-се дефинира в маркъп декларациите в DTD и може да се използва само в DTD дефиниции

107.XSL се използва за:

-трансформиране на XML документ към друг текстов документ

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

-точно един затварящ таг

110.Символите <, > и ' са валидни PCDATA символи

-невярно

110.XML документите могат да съдържат само един елемент-корен

-вярно

113. XML йерархията <a> <c> </c> е:

-некоректна

115.XML пространството от имена се отнася:

-само за йерархията от елементи, за която е дефинирано

118.Изпълняването на XSLT декларации

<xsl:value-of select="."/>

и

<xsl:value-of select="text()"/>

Води:

-до един и същ или до различни резултати в зависимост от типа на съдържанието на текущия елемент

119. 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 в резултатното дърво ще бъдат:

- със съдържанието на елементите в изходното дърво

120. Елементите от прост тип в една XML Schema са от даден предефиниран тип и:

- не могат да имат атрибути и под-елементи

124. Атрибутите на XML са чувствителни към регистъра

- вярно

126. 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.

- Вярно

129. 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 /> .`

N-Triples

130. Creator e:

- един от основните 15 елемента от DC Metadata

134. Ако имаме дефинирани две еднакви CSS свойства с различни стойности за един и същи елемент, то ще се приложи последно дефинираното свойство:

- винаги, освен когато едно от свойствата е по-специфично

135. Изберете едно:

- SAX използва push парсване, а StAX – парсване от тип pull

138. Атрибутът xml:base задава:

- база от относителни URI връзки към външни за документа ресурси

140. "other" и "none" са възможни стойности за XLink атрибутите:

- actuate и show

143. Отдалечените XLink ресурси винаги представляват външни за документа ресурси:

- Невярно

144. Разгледайте изброените по-долу символи:



- Между изброените символи има точно два недопустими в PCDATA елементно съдържание.

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

- edge

148. В XSLT, вземането на решение кои елементи ще бъдат обработени се задава със следния XSLT елемент:

- <xsl:apply-templates>

149. В CSS, задаването на елемент, чийто атрибут attr има стойност, съдържаща на произволно място в себе си стойността xpto, става чрез:

- [attr~=xpto]

150. В XSchema, за глобалните типове:

- minOccurs и maxOccurs може да се дефинират само за глобални типове, които се реферират

151.XSL елементът Apply templates се използва вътре в един шаблон template за извикване на други шаблони:

Активира рекурсивно обработката на всички наследници на елемента за които се отнасят

152.DocumentType::Node Interface се използва за получаване на информация за документ описан в DTD

DOM 1.0 не разрешава редактирането на този възел

153. В XLink, изходящите връзки (outbound links)могат да имат

Произволен краен брой в участващи в тях ресурси

154.SAX Element обектите:

Могат да разграничават обектите дефинирани изрично от тези специфицирани в DTD

155.Кое от твърденията е истина?

Само DOM Element обектите имат атрибути

156.Приложения,коита имат нужда от сложни структурни манипулации на много от XML елементите, трябва да се използват за:

DOM

157.В RDF, група от алтернативни свойства се задава чрез:

<rdf:Alt>

158.Квалифицирано XML име (наречено QName) е:

Име от вида namespace_prefix:local_name

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

Колекция

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

до един и същ или до различни резултати в зависимост от типа на съдържанието на текущия елемент

161. Всеки DOM възел (Node) може да има деца

Зависи от децата или невярно

162. Разгледайте P:first-letter { font-size: 200% }. Това е:

псевдо-елемент (pseudo-element)

163. С един DOM Element обект:

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

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

Отг.Невярно

165. В SAX приложните обекти, имащи достъп до XML сорса

трябва да бъдат регистрирани (от програмиста) за callback функциите на парсера

166. XMLReader в SAX 2.0 разширява стандартния Java Reader интерфейс

Отг.Невярно

167. SAX служи както за четене на XML документи, така и за генериране на XML

Отг.Невярно

168. DOMException връща HIERARCHY_REQUEST_ERR при опит за:

вмъкване на възел на неподходящо място в йерархията на DOM дървото

169. В DOM неподредена колекция от възли се дефинира чрез интерфейс с име:

NamedNodeMap

170. При опит да се включи недобре конструиран XML документ в XInclude се достига до фатална грешка

Не и при използване на атрибута "parse" със стойност "text"

171. XPATH изразът /catalog/product(@id='1234']/itemlist/item/@price='12'] адресира: •
d. Стойностите на всички item елементи, чиито атрибут price е равен на 12' и които имат за предшественик елемента product, притежаващ атрибут id равен на 1234

172. Методите за работа с атрибути на DOM Element интерфейса

Могат да разграничат стойността на атрибут, дефинирана изрично в XML документа от стойността по подразбиране, зададена в DTD

173. XQUERY изразът избира:

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

174. XLink използва отдалечени (remote) ресурси, за да представя:

Както външни за документа ресурси така и вътрешни ресурси, достъпни по URI

175. Събитието processingInstruction възниква, когато парсерът достигне до всяка една инструкция за обработка включително и до XML декларацията

Отг.Невярно

176.Методът `getAttribute()` на DOM интерфейса Node връща:

`NamedNodeMap`

177.Събитието `processingInstruction` възниква, когато SAX парсерът достигне до всяка една инструкция за обработка включително и до XML декларацията

Невярно

178. XPATH изразът `/catalog/product[@id='1234']/itemlist/item/@price='12'` адресира: •
d. Стойностите на всички `item` елементи, чиито атрибут `price` е равен на `12` и които имат за предшественик елемента `product`, притежаващ атрибут `id` равен на `1234`

179.MEDIA атрибутите, като `screen,aural,braille,tty` са част от:

CSS2 и CSS3 спецификациите

180. В CSS3 фиксираната (`fixed`) схема за позициониране:

Предизвиква повторение на кутията на всяка страница при `paged media`

181.В CSS абсолютната схема за позициониране

Се характеризира със задаване на явно отместване спрямо съдържащия блок.

182.Осъществяването на `embeddng`, а не (`inlining`) на `StyleSheet` според CSS правилата е възможност посредством:

Елемента `STYLE`

183. Даден CSS стил може да бъде `inliened` (а не `embedded`)стил в документ посредством

Атрибута `STYLE`

184. Xlink не може да дефинира фрагментни идентификатори за URI сочещи към възли или части от тях в XML ресурси

Вярно

185.В XLink информацията за това как се преминава през двойка ресурси се нарича

Дъга

186. Чрез `XPointer` можем да реферираме към повече от един елемента в XML документ.
Вярно

187.При включването на документа `"xpto. xml"` с използване на `<:inckwle href="xpto.xml parse="texr">`, документът `"xpto.xml"` ще бъде включен: Изберете едно като обновен текст

188.RDF литералите, могат да се явяват, както като субект така и като обект в RDF тройка

Невярно

189.Кои RDF контейнери задават ресурси на литерали, които могат да бъдат евентуално дублирани

rdf:Bag rdf:Seq

190. OWL клас, за който са описани както задължителните, така и достатъчните свойства за членство в клас се нарича

Дефиниран клас

191.Задаването на начина на преминаване през една XLink връзка става чрез:

Дъга(arc)

192.Речник с дефинирани и предлагани процедури за актуализирането му, тоест документиран процес на актуализация с речника се нарича:

Контролиран речник

193.Кои от изброените по-долу качества не са присъщи на спецификацията по онтология

Неформална

Неявна

194.Йерархичната подредба на понятия заедно с информацията за допълнителни нейерархични отношения представлява

Тезаурус

195.Използването на пространства от имена в XPointer изрази е

разрешено

196.Осъществяването на вграждане на CSS стилове в документа е възможно посредством

Елемента STYLE

197.В OWL един екземпляр може да бъде

член на множество класове

198.Кое от следните свойства е невярно?

Ако свойството T е дефинирано с обхват (range) C и ресурсът S е субект в тройка с предикат T, то следва че S rdf:type C

199.Елементите contributor,creator,publisher и rights са част от

DC пространството

200.RDF реификация се задава чрез: типа rdf:Statement и свойствата rdf:subject, rdf predicate и rdf object

201.Литералите в RDF тройките могат да бъдат:

Всеки от останалите отговори е грешен

202. Кое не е цел на RDF

Намаляването на разликата между интерпретацията на уеб страницата от потребителя и програмите или разширяване на XHTML атрибутите

203. Всички членове на подклас OWL са членове на супер-класовете на този клас

Вярно

204. В RDF класовете могат да бъдат екземпляри само на други класове

Вярно

205. Множественото наследяване не е допустимо в RDFS

Невярно

206. RDF твърдението `dbpedia:Mount_Etna rdf:type my-pref:Mountain, my-pref:Vulcano` е допустимо и валидно

Вярно

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

Вярно

208. Описанието (конкретизацията) на RDF твърждение чрез използване на RDF речника се нарича

Reflication

209. Свойството `rdfs:member` е супер-свойство на всички свойства, които са екземпляри на `Rdfs:ContainerMembershipProperty`

210. Как може да потиснем запазването на допълнителни празни интервали в съдържанието на текстови елементи?

Посредством използване на друг псевдо-атрибут, различен от изброените

211. В RDF твърдението `S rdf:type O`

`O` задава клас, категория, а `S` задава екземпляр на този клас

212. Елементите от прост тип в една XML Schema са от даден предефиниран тип и:

Не могат да имат атрибути и под-елементи

213. Задаването на мрежа от хора чрез FOAF твърдения става посредством релацията

`foaf:knows`

214. 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>
```

Истина

215.XML елементите може да съдържат само един елемент корен

Вярно

216.Символите < > са валидни PCDATA символи

Невярно

217.XSL се използва за

Трансформиране на XML документ към друг текстов документ

218.Формална , експлицитна спецификация на споделена концептуализация е дефиниция за

Нещо друго което не присъства в тези отговори

219.Annotation Property в OWL се използва за добавяне на метаданни:

Към класове, екземпляри и свойства

220.Могат ли екземплярите на класовете да бъдат част от онтологията описваща тези класове

Да

221.OWL класовете могат да се припокриват, да имат общи екземпляри

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

Вярно

222.В OWL също както в ооп не може да съществуват два класа с общи екземпляри

Невярно

223.Person, title, family, Name, knows, age, Document и Organizations са име на елементи
дефинирани от:

FOAF

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

Множество от отделни стойности на това Property

225. В RDFS Property е подмножество на RDFS Resources и има домейн (rdfs:domain)

Клас асоцииран с това property

226. Кое от отговорите по-долу не съдържа име на RDFS клас

rdfs:Property

227. Кое от отговорите по-долу не съдържа име на RDFS свойство

rdfs:subRange

228. Свойството rdfs:member е супер-свойство на всички свойства, които са екземпляри на

Rdfs:ContainerMembershipProperty

229. Описанието

ex:adrian foaf:knows ex:ger

ex:adrian foaf:age "41"^^xs:int.

представя RDF Triples в :

N3 формат

230. RDF Literals литералите могат да бъдат създадени зададени само, като обект тройка

Вярно

231. В RDF един екземпляр може да има няколко типа

Вярно

232.

rdf:Seq задава

група на подредени ресурси или литерали с вероятно дублиране

233. rdf:bag

група на не подредени ресурси или литерали с вероятно дублиране

234. Описането (конкретизацията) на RDF твърдение чрез използване на RDF речника се нарича:

Reflication

235. Описанието на група съдържаща само зададените в описанието членове става чрез използване на:

колекция

236. RDF Resource Description Framework служи за описания предназначени:

За обработка на софтуерни приложения

237. creator, title, publisher, contributor, date и format са име на елементи, дефинирани от:

DC

238. RDF графът от фигурата задава:

Лекцията е посетена от Джон, Мери и Крис.

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

6-32-47843-5

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

cursor API

241. Една XLink дъга (arc), която има локален стартов ресурс и отдалечен краен ресурс, се нарича:

Outbound

242. В XLink, входящите дъги (inbound arcs) могат да имат:

произволен краен брой участващи в тях ресурси

243. Resource Description Framework (RDF) представя информация за ресурси, които: могат да бъдат или да не бъдат достъпни в Уеб

244. 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:

Turtle

245. При парсане на XML документи посредством StAX, можем да се придвижваме само напред в XML документа.

Истина

246. Кое от показаните на фигурата средства не е пряко свързано с изграждане на семантичния Уеб:

Collaboration synchronization

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

Той се отнася неявно към пространството от имена на елемента, към който принадлежи.

248. На показаната фигура на RDF граф, T-shirt е:

Субект

249. rdf:Bag задава:

група на неподредени ресурси или литерали, с вероятно дублиране

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

[attr\$=xpto]

251. Описание (конкретизация) на твърдение с използване на RDF речника се извършва чрез:

типа rdf:Statement и свойствата rdf:subject, rdf:predicate и rdf:object

252. Свойството rdfs : isDefinedBy е специален (под)тип на:

rdfs:seeAlso

253. При автоматичното разпознаване на анотации, отношението броя на правилно разпознатите анотации към броя на всички съществуващи правилни анотации се нарича:

точност (precision)

254. Дефинирането на референция (Reference) към ключ (Key) в XSD може да има полета, които са:

както атрибути, така и елементи, като всички полета трябва да

съответстват на типа и на позицията на тези в ключа

255. RDF поддържа описания на групи, съдържащи само определени членове, посредством:

RDF колекции

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

Неверно

257. Описанието (конкретизацията) на RDF твърдение чрез използване на RDF

речника се нарича:

Reification

258. В RDFS един ресурс може да е екземпляр на няколко класа.

Верно

259. Кой два от изброените по-долу атрибути НЕ МОГАТ да участват в елемент от

XLink тип arc:

Role

Href

260. Кой(кои) RDF контейнер(и) задава(т) ресурси или литерали, които могат да

бъдат евентуално дублирани:

rdf:Bag и rdf:Seq

261. RDF реификация се задава чрез:

Типа rdf:Statement и свойствата rdf:subject, rdf:predicate и rdf:object

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

Document, DocumentFragment и Element

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

имплементиране на ContentHandler или разширяване на DefaultHandler

263. Създаване на специфична метаинформация и схема за употреба с цел

предоставяне на възможност за нови методи за достъп до информация се нарича:

Семантично анотиране

264. Ако свойство P е за субекта S, който участва в RDF тройката S P O, и имаме

S rdf:type C

265. Събитието processingInstruction възниква, когато SAX парсерът достигне до

всяка една инструкция за обработка, включително и до XML декларацията

Невярно