**XML Тест 1**

1. XML документите могат да съдържат само един елемент-корен – Вярно
2. Разгледайте имената на XML празните елементи <emptyTag /> и <emptyTag/ >. Те са: - първият е валиден, втория е невалиден
3. Кой от следните елементи не е с грешен XML синтаксис

<birth> 28.04.1990 </birth >

1. Разгледайте имената на XML елементите <xml-tag> и <tag-xml>. Те са: първия е невалиден, втория е валиден
2. Атрибутите на XML са чувствителни към регистъра – Вярно
3. Символите <, > и ‘ са валидни PCDATA символи – Невярно
4. Разгледайте изброените по-долу символи … - Между изброените символи има точно два недопустими в PCDATA елементно съдържание
5. CDATA съдържанията са – игнорирани от XML парсерите
6. Ако няма специфицирано кодиране за един XML документ и документът не е нито UTF-8, нито в UTF-16, тогава резултатът е – ERROR
7. За представяне на стойности в XML файлове, ние използваме XML атрибути и XML елементи. Кой от следващите отговори не е аргумент за употреба на атрибути вместо елементи? – Използваме атрибути за валидация на сложна структура
8. xml:id е псевдо-атрибут, който задава условие за уникална стойност, като това: не е необходима да се декларира в DTD или XML схема
9. Как можем да потиснем запазването на допълнителните (следващи един след друг) празни интервали в съдържанието на текстови елемент? – посредством използване на друг псевдо-атрибут, различен от изброените в останалите отговори на въпроса
10. Атрибутът xml:base задава – база за относителни URI връзки към външни за документа ресурси
11. XML йерархията <a> <b> <c> </b> </c> </a> e: некоректна
12. Според XML синтаксиса, за всеки непразен XML елемент, за отварящия му таг трябва да има – точно един затварящ таг
13. Единственото изискване един XML документ да бъде валиден е този документ да е структуриран (формиран) съгласно правилата за XML синтаксиса – Невярно
14. Целта на валидацията на XML документ от XML парсера е да се провери дали XML документът е добре структуриран (well-formed) – Невярно
15. Изберете верния отговор – Всеки валиден XML документ е добре конструиран, но обратното не е задължително вярно
16. Следният XML документ е добре конструиран ……. – Истина
17. XML пространствата от имена се дефинират чрез атрибути и могат да бъдат специфициране в кой да е XML елемент - Вярно
18. XML пространството от имена се отнася – само за йерархията от елементи, за която е дефинирано
19. XML пространства от имена се използват за разграничаване – и на XML елементи, и на XML атрибути
20. Уникална идентификация на пространство от имена се реализира с – URL
21. Квалифицирано XML име (наречено QName) е – име от вида namespace\_prefix:local\_name
22. 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
23. В примера по-долу …. – Дефиницията на елемента <hmmm> е валидна според спецификацията на пространствата от имена
24. Кое е името подразбиращо се (default) пространство от имена в декларацията по-долу ……. – няма подразбиращо се пространство от имена
25. Един XML документ може да има повече от едно DTD – Вярно
26. Един XML документ може да има повече от едно вътрешно DTD – Невярно
27. При използване на вътрешна и външна DTD дефиниция, вътрешната DTD може да предефинира – само ENTITY и ATTLIST на външната дефиниция
28. 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
29. Какво не е възможно да бъде описано в DTD – минимален и максимален брой срещания на елемент
30. Атрибутите, специфицирани в DTD посредством ключовата дума IMPLIED – могат да бъдат определени (спацифицирани) в XML
31. Редът <!ATTLIST point honorific (Mr|Ms|Mrs|Rev|Dr) … > е валиден DTD пример за – name group
32. Общо текстово Entity – Не може да бъде използвано рекурсивно, но може да се появи в съдържанието на елемент и/или в стойност на атрибут
33. Параметрично Entity – се дефинира в маркъп декларациите в DTD и може да се използва само в DTD дефиниции
34. Binary entity може да се използва само като атрибут от тип ENTITY – Вярно
35. Всяко IDREFF, декларирано в DTD, трябва да – избере дадено определено ID
36. Кой от следните изрази е валиден пример за дефиниране на избор (choice) от елементи в DTD – (A|B|C)
37. Кой от следните изрази е валиден пример за дефиниране на последователност (sequence) от елементи в DTD – (A,B,C)
38. В XSchema референцията към тип може да реферира – само към глобални типове
39. В XSchema, локалните типове – не могат да бъдат директни наследници на корена
40. В XSchema, глобалните типове – винаги са директни наследници на корена
41. В XSchema, за глобалните типове – minOccurs и maxOccurs може да се дефинират само за глобални типове, които се реферират
42. Елемнтите от прост тип в една XML Schema са от даден предефиниран тип и – не могат да имат атрибути и под-елементи
43. В XML Schema, ние можем да построим производни (derived) types – както с extension, така и с restrictions
44. В XSchema, типът на всеки елемент на еквивалентен клас трябва да бъде – същият като типа на елемента-екземпляр, или негов дериват
45. Ако в XSchema дефинираме exact тип, то – той може да има деривантни типове, но те не могат да бъдат използвани в XML документа вместо този exact тип
46. XSchema шаблонен фасет (pettern facet) е фасет за тип данни от тип – string
47. XML Schema разрешава да се прави разлика между уникална стойност (unique) и ключ (key) – Вярно
48. 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
49. Регулярният израз [1-9]?[0-9] дефинира XSchema стрингови стойности, които са – от 0 до 99
50. Регулярният израз [^0-9]x дефинира XSchema стрингови стойности, които представляват – който и да е не-цифров символ, следван от символа х
51. Кой от елемнтите ISBN по-долу не се описва от дефиницията <element name=“ISBN“ type=“cat:ISBNType“/>, където ISBNType e – 6-32-47843-5
52. Най-общо маркъп езиците включват: - структурни(layout), функционални(action), семантични(meaning) и стилистични(appearance) описания
53. XPath изразът ./book[author/last=”пробен изпит“] върща: - елементите book – преки наследници на текущия елемент, които имат под-елемент author с под-елементи last, равен на “пробен изпит”
54. 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
55. Параметричните единици (Parameter Entities) са декларирани в DTD и се използват: - единствено в декларациите на DTD