

## 一、选择题

- XML 采用以下哪种数据组织结构？（ ）  
A 星状结构 B 线状结构 C 树状结构 D 网状结构
- 下列关于 XML 文档中根元素的说法不正确的是（ ）  
A 每一个结构完整的 XML 文档中有且只有一个根元素  
B 根元素完全包括了文档中其他所有元素  
C 根元素的起始标记要放在其他所有元素的起始标记之前，而根元素的结束标记要放在其他所有元素的结束标记之后  
D 根元素不能包含属性节点
- 下面哪个公司或组织制定了 XML：（ ）  
A ISO B Oracle C W3C D Microsoft
- 以下哪个是规范的 XML 元素：（ ）  
A -booklist B 3class C Book Case D book-info
- 对于一个 XML 文档，以下哪个名称是错误的？（ ）  
A <Note> B <h1> C <1dollar> D 以上三个都不正确
- <?xml version="1.0"?>  
<!--例子 -->  
<greeting>Hello, World!</greeting>

上面的 XML 文档属于（ ）文档。

- A 无效的 B 有效的 C 格式良好的 D 格式错误的
- 以下的 xml 文档片段中，属性符合 XML 语法规则的是（ ）。  
A <car name="zhang's car"> B <car name='zhang's car'>  
C <car name='jieda car '"> D <car name=jieda >
- >、<、&等在 XML 文档中被赋予了特殊的意义，所以在 XML 文档中不能直接包含这些字符，而是对这些字符采用了特殊编码，< 对应的特殊编码是：（ ）  
A &amp; B &lt; C &lgt; D &quot;
- 在 DTD 声明中使用下列哪个符号可以允许元素不出现或者只出现一次？（ ）  
A ? B \* C | D +
- 在一个 DTD 元素中定义了一个元素“学员”为<!ELEMENT 学员(#PCDATA)>，那么如下的 XML 实例片段中合法的是（ ）  
A <学员>张三</学员> B <学员><姓名>张三</姓名></学员>  
C </学员> D <学员>一班<姓名>张三</姓名>同学</学员>

11. 分析以下在 DTD 文档：

<!ELEMENT doc(title?,chap\*)>

<! ELEMENT title(#PCDATA)>

<! ELEMENT chap(sect+)>

<! ELEMENT sect(para+)>

<! ELEMENT para(#PCDATA)>

下面的 XML 实例（ ）能够通过 DTD 的校验。

- A <doc><chap><para>Text</para></chap></doc>
  - B <doc><chap><sect><para>Text</para></sect></chap></doc>
  - C <doc><title>Text</title></doc>
  - D <doc><title>Text</title><sect><para>Text</para></sect></doc>
12. 以下哪一项是使用参数实体的正确方法？（ ）



22. XML 文档声明: `<?xml version=" 1.0" ?>`, 则文档的 encoding 为 ( )。
- A. ASCII                      B. Unicode                      C. UTF-16                      D. UTF-8
23. DTD 中, 属性用 ( ) 关键字声明。
- A. ENTITY                      B. ELEMENT                      C. DOCTYPE                      D. ATTLIST
24. Schema 文档中的 `<xsd:sequence>` 元素的用途是 ( )。
- A. 强制元素按任意顺序出现一次  
B. 强制在一个数据类型中的元素按特定的顺序出现  
C. 强制属性值按特定顺序出现  
D. 只能用于注释目的
25. DOM 解析中, 下面 ( ) 方法可以获得 XML 文档节点树的根元素节点。
- A. `getDocumentElement()`                      B. `getEntities()`  
C. `getPublicID()`                      D. `getWholeText()`
26. 已知 XML 文档包含如下内容 (list 为根元素):
- ```
<class>
  <stu id="001">
    <age>18</age>
  </stu>
  <stu/>
  <stu id="002">
    <age>15</age>
  </stu>
  <monitor>
    <stu id="003"></stu>
  </monitor>
</class>
```
- 使用 XSLT 进行转换时, 语句 `<xsl:template match="/class/stu[@id]">` 能够匹配到 ( ) 个 item 元素。
- A. 3                                              B. 2  
C. 1                                              D. 0
27. 下列标记定义不合法的是 ( )。
- A. `<price>`                                              B. `<_author>`  
C. `<2title>`                                              D. `<li.name>`
28. 解析下面 XML 文档时, 下列说法正确的是 ( )。
- ```
<book>
  <author>耿祥义</author>
  <ISBN>9787302206491</ISBN>
</book>
```
- A. “耿祥义” 是文本对象  
B. “耿祥义” 是元素对象  
C. `<book>` 是元素对象, 同样也是文本对象  
D. `<ISBN>` 是元素对象, 同样也是文本对象
29. Schema 文档中可以直接指向另一个元素定义模块, 避免在文档中多次定义同一元素的元素属性是 ( )。
- A. form                      B. abstract                      C. block                      D. ref

30. 关于 DOM 的描述错误的是 ( )。
- A. DOM 使开发者能够以编程方式读取、操作和修改 XML 文档
  - B. W3C 组织公布了 DOM 模型的规范, 然后各个软件厂商 (比如微软) 再根据 W3C 的规范开发 DOM 解析器, 并且提供一系列的编程 API, 这些 API 都是遵守 W3C 规范的
  - C. DOM 模型是 W3C 组织开发出来的一个在内存中表示 XML 数据的线性模型
  - D. 能使用 JavaScript 进行 DOM 开发, 也能使用 Java、C#
31. 实体是一个外部文件, 该外部文件由个人或工作小组所定义并认可的外部普通实体使用的格式是 ( )。
- A. LIST
  - B. ADMIN
  - C. PUBLIC
  - D. SYSTEM
32. 对于下面的 DTD 和 XML 文档, 正确的陈述是 ( )。

exam1.dtd

```
<!ELEMENT root (element1, element2?)>
<!ELEMENT element1 (#PCDATA)>
<!ELEMENT element2 (#PCDATA)>
```

exam1.xml

```
<?xml version="1.0"?>
<!DOCTYPE element SYSTEM "exam1.dtd">
<root>
  <element1>
    <![CDATA[内容]]>
  </element1>
</root>
```

- A. DTD 中的元素名 element2? 定义错误, 元素名不允许出现字符 “?”
  - B. XML 文档中的 DOCTYPE 定义中, SYSTEM 应该改为小写
  - C. XML 文件中遗漏了 element2
  - D. XML 文档中的根元素与 DOCTYPE 中的定义不同
33. W3C XML Schema 文档中, ( ) 元素用于声明只有一个相容元素必须出现, 用于互斥情况。
- A. sequence
  - B. choice
  - C. group
  - D. all
34. XML 名称空间名是指 ( )。
- A. W3C 制定的一个技术规范
  - B. 一个可以访问到的 URI
  - C. 任何 URI
  - D. 紧跟在 xmlns: 后面的前缀名称

35. 关于 XML 命名空间的实例片段

```
<?xml version="1.0" encoding="GB2312"?>
<root>
  <p1:电视机 xmlns:p1="www.China.com">
```

Made In China

```
    <p1:显示器>Made In Guangzhou</p1:显示器>
  </p1:电视机>
  <p1:开关>Made In Beijing</p1:开关>
```

```
</root>
```

- 名称空间前缀使用错误的是 ( )。

- A. <root>
- B. <p1:开关>
- C. <p1:显示器>
- D. <p1:电视机>

36. 下面哪个标记是调用模板的标记? ( )
- A. <xsl:for-each> B. <xsl:template>  
C. <xsl:apply-templates> D. <xsl:if>
37. 下面定义的 CDATA 段正确的是 ( )。
- A. <![CDATA[  
        &你好&  
    ]]>  
B. <![CDATA  
[  
        你好  
]]>  
C. <![CDATA[  
        &你好&  
        <![CDATA[  
            &你好&  
        ]]>  
    ]]>  
D. <![CDATA[  
        &你好&  
    <![CDATA[  
        &你好&  
    ]>
38. XSL 文档根元素是 ( )。
- A. <xsl:root> B. <xsl:stylesheet>  
C. <xsl:template> D. <xsl:apply-templates>
39. 如果在 Schema 文档中定义了目标命名空间, 要在 XML 文档中使用 ( ) 引用该 Schema 文件。
- A. noNamespaceSchemaLocation B. Location  
C. schemaLocation D. namespaceSchemaLocation
40. 当 SAX 解析器解析到文档中元素的开始标记时, 会调用 ( ) 方法。
- A. public void startElement()  
B. public void startDocument()  
C. public void startPrefixMapping()  
D. public void startDocumentLocator()
41. 以下事件处理函数哪一个不是 SAX 实现会产生的 ( )
- A void startDocument()  
B void characters(char[] ch, int start, int length)  
C void startElement(String uri, String localName, String qName)  
D void endElement(String uri, String localName, String qName)
42. SAX 的默认基类 DefaultHandler 类没有实现那些接口 ( )
- A) Attributes 接口  
B) DTDHandler 接口  
C) ErrorHandler 接口  
D) ContentHandler 接口

43. XSL 不包括以下哪些部分 ( )
- A) XSLT                      B) XPath                      C) CSS                      D) XSL-FO
44. 声明语句<!ATTIBLIST student studentID ID #REQUIRED>的作用是 ( )
- A、声明元素 student 的属性 studentID 是可有可无的；  
B、声明元素 student 的属性 studentID 是唯一标识类型，可以省略；  
C、声明元素 student 的属性 studentID 是默认取值，可以省略；  
D、声明元素 student 的属性 studentID 是唯一标识类型，不可省略；
45. 在 XML 文档的第一条声明语句中，不能使用的属性是( )
- A、version                      B、encoding                      C、standalone                      D、 name
46. 不符合格式良好的 XML 文档要求的是 ( )
- A、文档可以有一个以上的根元素；                      B、有符合规范的声明语句；  
C、每个元素有正确的起始、结束标记；                      D、元素正确嵌套，没有交叉现象出现；
47. 在进行属性声明时，错误的是 ( )
- A 标记中可以使用多个属性  
B 属性可以出现在起始标记或结束标记中；  
C 属性名是大小写敏感的；  
D 属性值必须用引号引上；
48. 以下不属于 XML 语法规定的是 ( )
- A、元素必须正确嵌套  
B、标记任意大小写  
C、所有的属性值必须放在双引号或单引号中；  
D、若文档没有 DTD，属性值的默认类型是“CDATA”；
49. 声明语句<!ELEMENT 学生信息 ((地址，成绩)|#PCDATA)>指明元素学生信息是 ( )
- A、任意子元素组成；  
B、由子元素“地址”“成绩”和字符数据混合组成；  
C、由字符“地址，成绩”和字符“#PCDATA”组成；  
D、由文本类数据组成
50. XML Schema 文档中语句：
- <element type="name" minOccurs="1" maxOccurs="\*" />
- 中属性 type 的作用是( )
- A、指定子元素的属性；  
B、指定子元素的数据类型；  
C、指定子元素是引用元素声明中 name 属性值为 name 的元素；  
D、指定子元素名称；
51. 在 XML 中，下列关于 DOM 的叙述不正确的。
- A DOM 是独立于开发语言和平台的，因此使用 Visnal Basic、Java、Visual C++等开发工具使用的 DOM 编程 API 是一致的  
B XML 文档通过 load 方法被装载进内存后，在内存中形成一个 DOM 文档对象模型树  
C 通过 DOM API，软件开发人员可以控制 XML 文档的结构和内容  
D 通过 DOM 在 XML 文档中只能按照顺序方式导航
52. 在 XML 中，下述关于 XSL 的说法不正确的是。
- A XSL 是一种用来转换 XML 文档的样式表，它包含转换和格式 XML 文档的规则  
B XSL 在转换 XML 文档过程中，首先根据匹配条件修改源文档内容，然后输出修改

后的文档内容

C XSL 包含了 XSLT 和 Xpath 的强大功能,从而可以把 XML 文档转换成任何一种其它格式的文档

D XSL 文件是同一系列模板组成的,任何一个 XSL 文件至少包括一个模板

53. 考虑如下的 XML.Schema 示例

```
<xs:element name="Price">
  <xs:complexType>
    <xs:attribute name="currency" type="xs:string" />
  </xs:complexType>
</xs:element>
```

属性 currency 声明等同于下面选项的 DTD 声明。( )

A <!ATTLIST Price currency CDATA #REQUIRED>

B <!ATTLIST Price currency CDATA #FIXED>

C <!ATTLIST Price currency CDATA #IMPLIED>

D <!ATTLIST Price currency PCDATA #IMPLIED>

54. 对于 XML 文档实例片段<image height=" 50" width=" 50" />, 最恰当的描述其结构的 DTD 片段是。( )

A

```
<!ELEMENT image ANY>
<!ATTLIST image
height CDATA #REQUIRED
width CDATA #REQUIRED>
```

B

```
<!ELEMENT image EMPTY>
<!ATTLIST image
height CDATA#REQUIRED
width CDATA#REQUIRED>
```

C

```
<!ELEMNET image(#PCDATA)>
<!ATTLIST image
height CDATA#REQUIRED
width CDATA# REQUIRED>
```

D

```
<!ELEMNET image(height,width)>
<!ATTLIST image
height CDATA#REQUIRED
width CDATA# REQUIRED>
```

55. 下面的 XML 片断中结构完整的是。( )

A <customer name="xml.con"><address>123 MainStreet</address></customer>

B <customer><name>Joe's XML Works</name><address>New York</costomer>

C < customer type=extemal><name>Partners Unlimited</name></customer>

D <customer name="John Doe"><address>123 Main Street</address> <zip code="01837"/></customer>

56. 以下不正确的是。( )

- A DTD 定义了 XML 文档中包含的标记、元素、元素类型以及属性
- B 一个结构完整的 XML 文档一定是合法的 XML
- C 命名空间能够区别不同来源的元素、属性的定义
- D 文档 DTD 可以包含在 XML 文档内，也可以在 XML 文档外定义

57. 下列说法错误的是。( )

- A XSL 在转换 XML 文档时分为明显的两个过程：首先转换文档结构，然后将文档格式化输出
- B XSLT 包含 XSL 和 XPath 的强大功能，从而可以把 XML 文档转换成任何一种其它格式的文档
- C 如果将 XML 文档看成 DOS 目录结构，XPath 就是 cd、dir 等目录操作命令的集合
- D 如果将 XML 文档看作一个数据库，XPath 就是 SQL 查询语言

58. XML 中，( ) 是文档对象模型 DOM 中的基本对象，元素、属性、注释、处理指令等都可以认为是它。

- A Document
- B Node
- C NodeList
- D Element

59. 阅读下面 XML 文档，然后判断下列说法正确的是。

```
<book>
<author>tom</author>
<bookcode>12</bookcode>
</book>
```

- A <book>是元素对象，同样也是文本对象
- B <author>是元素对象，同样也是文本对象
- C tom 是文本对象
- D tom 是元素对象

60. 对于如下的 XML 文档实例 (people.xml) ;

```
<?xml version="1.0" ?>
<PEOPLE>
  <PERSON PERSONID="E01">
    <NAME>Tony</NAME>
    <ADDRESS>Beijing</ADDRESS>
  </PERSON>
  <PERSON PERSONID="E02">
    <NAME>Bill</NAME>
    <ADDRESS>USA</ADDRESS>
  </PERSON>
</PEOPLE>
```

引用如下的 XSL 文件：

```
<?xml version="1.0" encoding="gb2312" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" ?>
  <xsl:template="">
    <xsl:value-of select="PEOPLE"/>
  </xsl:template:>
```



</xsl:stylesheet>

在浏览器中的输出结果为。

A 无任何输出

B E01 Tony Beijing Bill USA

C Tony Beijing Bill USA

D <PERSON PERSONID="E01">

<NAME>Tony</NAME>

<ADDRESS>Bijing</ADDRESS>

</PERSON>

<PERSONPERSONID="E02">

<NAME>Bijing</NAME>

<ADDRESS>USA</ADDRESS>

</PERSON>

61. <?xml version="1.0"?>

<!--例子 -->

<greeting>

Hello, World!

</greeting>

上面的 XML 文档属于文档。

A 无效的

B 有效的

C 格式良好的

D 格式错误的

62. <?xml version="1.0"?>

<!DOCTYPE customer[

<!ENTITY location "jianwai street">

<customer>

<name>jack</name>

<address>&location;</address>

</customer>

关于上述 Xml 文档，以下描述正确的是（ ）。

A 该 Xml 文档中使用了外部 DTD。

B 通过浏览器打开上述 Xml 文档时，字符串 jianwai street 代替了& location;

C 该 Xml 文档不是有效的 XML 文档。

D 该 Xml 文档中的 location 为参数实体。

63. <customer\_summary xmlns:addr="http://www.xyz.com/addresses/"

xmlns:books="http://www.zyx.com/books"/>

<addr:name><title>Mrs.</title></addr:name>

<books:title>Lord of the Rings</books:title>

上述是一段关于 XML 命名空间的实例片段，以下说法错误的是。

A 本例中，命名空间的前缀分别是 addr、books

B 第一个 <title> 元素属于 addr 名称空间，因为其父元素 <addr:name> 属于该名称空间。

- C 名称空间中定义的字符串必须是可以访问的 URL，比如：XML 解析器会到 [url]http://www.zyx.com/books/[url] 去搜索 DTD 或 schema
64. 以下关于 xslt 说法不正确的是（ ）。（选择一项）
- A Xslt 是 xsl 的一部分，有独立的命名空间。
- B Xslt 可以把 xml 文档转换为 xhtml。
- C 要执行 xslt 转换，只需要 xml 文档，xslt 样式表和浏览器就可以了。
- D Xslt 使用 xpath 对 xml 文档进行导航。
65. 属性指定元素最多出现的次数。（选择一项）
- A minOccurs
- B maxOccurs
- C minExclusive
- D maxExclusive
66. 已知 xml 文档包含如下内容（list 为根元素）：
- ```
<list>
    <item id="0001">
        <price>18</price>
    </item>
    <item id="0002">
        <price>15</price>
    </item>
    <item/>
    <product>
        <item id="01"></item>
    </product>
</list>
```
- 使用 xslt 进行转换时，语句<xsl:template match="/list/item[@id]">能够匹配到（ ）个 item 元素。（选择一项）
- A 3
- B 2
- C 1
- D 0
67. XML( )提供了一种避免元素命名冲突的方法。
- A.命名空间 B.DTD C.XSD D.XSL
68. 由( )元素定义的模板规则是 XSL 样式表的最重要的部分
- A.xsl:value-of B.xsl:template C.xsl:apply-templates D.xsl:choose
69. 以下关于 XML 文档的声明的语句正确的是 。
- A <xml version="1.0" encoding="UTF-8">
- B <?XML version="1.0" encoding="UTF-8"?>
- C <?xml Version="1.0" encoding="UTF-8"?>
- D <?xml version="1.0" encoding="UTF-8"?>
70. 使用外部 DTD，在 XML 文档声明中 standalone 的值应为（ ）。
- A yes
- B no
- C 0

## 二、填空题

1. `<?xml version="1.0" encoding="gb2312"?>`是\_\_\_\_\_。
2. \_\_\_\_\_是解决 XML 元素多义性和名字冲突问题的方案。
3. XML 元素由\_\_\_\_\_、结束标记和两者之间的内容三部分组成。
4. XML 通常使用\_\_\_\_\_或\_\_\_\_\_来验证内容是否有效。
5. DTD 中可以定义的实体种类包括\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
6. DTD 描述的基本部件是\_\_\_\_\_和\_\_\_\_\_,它们负责确定 XML 文档的逻辑结构。
7. DTD 全称为\_\_\_\_\_,它分为外部 DTD 和\_\_\_\_\_两种。
8. DTD 文件中,表示元素能出现零次或多次的限定符号为\_\_\_\_\_,表示元素出现至少一次的限定符号位\_\_\_\_\_,表示元素出现零次或一次的限定符号为\_\_\_\_\_。
9. DTD 中%字符用于声明\_\_\_\_\_实体。实体定义`<!ENTITY MJ "Michael Jordan">`,则 XML 文档中引用“MJ”实体的表达式为\_\_\_\_\_。
10. 在 DTD 中,元素类型是通过\_\_\_\_\_标记声明的。
11. 属性类型设为 ID,表明该属性的取值\_\_\_\_\_。
12. 在 DTD 文档中下面语句的作用是\_\_\_\_\_。  
`<!ELEMENT 教师(姓名,性别)>`
13. 在 DTD 文档中下面语句的作用是\_\_\_\_\_。  
`<!ENTITY picl SYSTEM "fil.jpg">`
14. 声明语句`<!ATTLIST 姓名 身份证号 ID #REQUIRED>`的作用是为\_\_\_\_\_元素声明\_\_\_\_\_属性。
15. DTD 中指定,属性是必选属性要使用的关键字是:\_\_\_\_\_;指定属性是可选属性要使用的关键字是\_\_\_\_\_。
16. 能够通过 DTD 验证的 XML 文档我们称为\_\_\_\_\_的文档。
17. 声明语句`<!ATTLIST 姓名 身份证号 ID #REQUIRED>`的作用是\_\_\_\_\_。
18. 所有 Schema 文档都必须使用\_\_\_\_\_作为其根元素。
19. 声明用户自定义的简单数据类型使用关键字\_\_\_\_\_,声明复杂类型元素要使用的关键字是\_\_\_\_\_。
20. Schema 的数据类型分为\_\_\_\_\_和\_\_\_\_\_。
21. 选择 catalog 元素下的所有名为 cd 的子元素,要使用的 XPath 表达式是\_\_\_\_\_。
22. `<xsl:sort>`元素一般作为`<xsl:apply-templates>`或\_\_\_\_\_的子元素出现。
23. XSL 是目前用来设计 XML 文档显示样式的主要技术之一,它主要由\_\_\_\_\_,\_\_\_\_\_和 XSL-FO 三部分组成。
24. 在 XSL 转换中,原始 XML 文档中的元素和元素的内容组成的文档树被称为\_\_\_\_\_,转换之后,结果文档中的元素和元素内容组成的文档结构树被称为\_\_\_\_\_。
25. XSL 文档的命名空间 URI 为:\_\_\_\_\_。
26. 要引用 XSL 转换样式表文档,xml-stylesheet 处理指令的 type 属性值必须设为\_\_\_\_\_。
27. 在 XSL 样式表中可以应用匹配模板的元素是\_\_\_\_\_。

28. XML 文档利用 DOM 技术加载到内存中时，会被封装成一个\_\_\_\_\_对象。
29. XSL 文件中，主模板的 match 属性值为\_\_\_\_\_。
30. match= “/” 语句的含义是\_\_\_\_\_。
31. 使用\_\_\_\_\_标记可以对多个同名的节点访问，并且该节点可以设置在显示数据的时候，可以按升序或降序显示。
32. XML 文档中下条语句的作用是：\_\_\_\_\_。  
`<?xml-stylesheet type="text/xsl" href="mysub/file1.xsl"?>`
33. 可以在 XSL 文件中使用<xsl:element>创建一个标记，设置标记的名称，通过\_\_\_\_\_标记可以为创建的设置相应的属性。
34. XML 文档中下条语句的作用是：\_\_\_\_\_。  
`<?xml-stylesheet type="text/xsl" href="mysub/file1.xsl"?>`
35. DOM 全称为\_\_\_\_\_，DOM 解析采用\_\_\_\_\_驱动，SAX 解析使用\_\_\_\_\_驱动。
36. DOM 解析中，最基本的数据类型是\_\_\_\_\_接口类型，它有很多子接口类型，其中文档使用\_\_\_\_\_接口类型，元素使用\_\_\_\_\_接口类型，文本使用\_\_\_\_\_接口类型，属性使用\_\_\_\_\_接口类型。
37. SAX 解析器是否报告 XML 声明给事件处理器？\_\_\_\_\_（填“是”或“否”）
38. XML 文档片段为：  
`<book xmlns:tup=" 清华大学出版社" >  
 <tup:bookName>XML 程序设计</tup:bookName>  
</book>`
- 如果 SAX 解析器设置为支持名称空间，写出解析<tup:bookName>元素结束标记时 endElement(String uri, String localName, String qName)方法中各个参数的取值。  
uri=\_\_\_\_\_；localName=\_\_\_\_\_；qName=\_\_\_\_\_。
39. DOM 解析中，使用\_\_\_\_\_接口类型的 getElementElement()方法可以获取根元素节点。而元素 Element 节点的 getElementsByTagName(String tagName)方法返回类型为\_\_\_\_\_接口类型。
40. DOM 解析支持对 XML 文档的\_\_\_\_\_，SAX 解析支持对 XML 文档的\_\_\_\_\_。

### 三、简答题

1. 什么是 XML？简述 XML 与 HTML 的区别。
2. 什么是 DTD？它可以分为几类？
3. 简述 Schema 与 DTD 的区别。
4. 什么是 DOM？简述 DOM 的工作原理。
5. 什么是 SAX？简述 SAX 的工作原理。
6. 什么是 XSL？简述使用 XSL 定义 XML 文档显示方式的基本思想。
7. 分别写出以下路径对应得简化路径。

child::root

self::node()/root

descendant-or-self::id

//id[position()=1]

/child::root/child::student/attribute::id

child::student/attribute::id[text()='101']

8. 有 XML 文档:

```
<students>
  <student id="20090534">
    <name>小王</name>
  </student>
  <student id="20090535">
    <name>小张</name>
  </student>
</students>
```

编写读取“20090534”这个属性值得 XPath 路径;

编写读取“小张”这个元素内容得 XPath 路径;

编写得到 students 元素下的子元素 student 元素个数的 Xpath 路径。

9. 写出下列缩略词对应的中文意思。

XML, DTD, XSD, CSS, XSL, XSLT, DOM

10. 考虑如下的 XML Schema 示例

```
<xs:element name="库存">
  <xs:complexType>
    <xs:attribute name="类型" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
```

写出上述 Schema 中元素“库存”及其属性“类型”在 DTD 中的等价声明。

四、改错题。下列代码是否正确? 若不正确, 为什么?

1、 ( )

```
<?xml version="1.0" encoding="UTF-8"?>
<Students>
  <学生>
    <NAME>jojo</name>
    学号>941003</学号>
  </学生>
</Students>
```

2、 ( )

```
<?xml version="1.0" encoding="UTF-8"?>
<SCENE>Enter Pozzo and Lucky. </SCENE>
<VLADIMIR>Gogo!</VLADIMIR>
<POZZO>
  <ACTION>clutching onto Lucky who staggers.</ACTION>
</POZZO>
```

3、 ( )

```
<?xml version="1.0" encoding="GB2312" standalone="yes"?>
<tt>
  <TITLE>My test!</TITLE>
  <GREETING>Hello, World!</GREETING>
  <SPECIAL>It is written in XML.</SPECIAL>
```

```

        <AUTHOR>Feelbed</AUTHOR>
    </tt>
4、(

```

<学生信息>

```

    <person>
        <name>王花花</name>
        <sex>女</sex>
        <age>23</age>
    </person>
    <person>
        <name>李黑黑</name>
        <sex>男</sex>
        <age>20</age>
    </person>

```

</学生信息>

## 五、程序填空题

1. <?xml \_\_\_\_\_ encoding=" \_\_\_\_\_ " ?>

```

<_____ [
    <!ELEMENT UNIVERSITY (_____)>
    <!ELEMENT NAME (#PCDATA)>
    <!ELEMENT ADDRESS (#PCDATA)>
    <!ENTITY name "青岛大学">
    <!ENTITY address "宁夏路 308 号">
]

```

>

<!--this is a comment-->

< UNIVERSITY >

<!--NAME 元素中的内容为实体 name 中的内容-->

<NAME>\_\_\_\_\_</NAME>

<!-- ADDRESS 元素中的内容为实体 address 中的内容-->

<ADDRESS>\_\_\_\_\_</ADDRESS>

</ UNIVERSITY >

2. <\_\_\_\_\_ PRODUCTDATA \_\_\_\_\_ " 1.dtd"> <!--外部 DTD-->

<PRODUCTDATA>

<PRODUCT PRODUCTID="P001" >

<PRODUCTNAME>Gone With the Wind</PRODUCTNAME>

<PRICE>25.00</PRICE>

<QUANTITY>35</QUANTITY>

</PRODUCT>

</PRODUCTDATA>

3. 有如下 XML 模式:

<xsd:element name=" \_\_\_\_\_ ">

```

<xsd:simpleType>
  <xsd:restriction base="_____">
    <xsd:pattern value="_____">
  </xsd:restrictions>
</xsd:simpleType>
</xsd:element>

```

设 XML 文档中的<language>的内容为：English 或 France。完成上面的 XML 模式。

4. 根据上一题 XML 文档，将下面的模式文件的内容补充完整

```

<?xml version="1.0" encoding="UTF-8"?>
<____ xmlns:xs="____">
  <____ name="PRODUCTDATA" type="prdata"/>
  <____ name="prdata">
    <xs:sequence>
      <xs:element name="PRODUCT" type="prdt"/>
    </xs:sequence>
    _____
  <xs:complexType name="prdt">
    <xs:sequence>
      <xs:element name="PRODUCTNAME" type="xs:string"/>
      <!--元素 PRICE 中存放的内容为实数，并且此元素出现的最小次数是 1 次，最多是 3 次-->
      <xs:element name="PRICE" type="____" _____="1"
      _____/>
      <xs:element name="QUANTITY" type="xs:integer"/>
    </xs:sequence>
    <____ name="PRODUCTID" type="xs:ID" use="required"/>
  </xs:complexType>
  _____

```

5. <?xml version="1.0" encoding="gb2312" ?>

```

<xsl:stylesheet version="____" xmlns:xsl="____">
  <xsl:template _____="/">
    <html>
      <body>
        <h2 align="center">学生成绩单</h2>
        <table border="1" cellpadding="0" align="center">
          <tr><th>姓名</th><th>性别</th><th>生日</th><th>成绩</th></tr>
          <____ select="roster/student">
            <tr>
              <td><xsl:____ select="name"/></td>
              <td><xsl:____ select="sex"/></td>
              <td><xsl:____ select="birthday"/></td>
              <td><xsl:____ select="score"/></td>
            </tr>

```

```
</table>
</body>
</html>
```

6. 使用 DOM 解析 XML 文件代码如下，读程序完成填空。

exa.xml 代码如下：

```
<?xml version="1.0" encoding="GB2312"?>
<school>
<student ID="s101" sex="女">
<name>李华</name>
<age>12</age>
<score>92</score>
</student>
</school>
```

Test.java 代码如下：

```
import org.w3c.dom.*;
import javax.xml.parsers.*;
import java.io.*;

public class Test {
    public static void main(String args[]) throws Exception{
        DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        //DOM 解析返回文档对象
        _____ doc = builder.parse(new File("exa.xml"));
        //获得文档根元素节点
        _____ root = doc.getDocumentElement();
        //获得根节点下元素标记名为 student 的所有的元素节点的集合
        _____ nl = root.getElementsByTagName("student");
        for (int i = 0; i < nl.getLength(); i++) {
            //获取 nl 集合中的第 i 个节点对象
            Element stu = (Element) nl.item(i);
            //获得元素 student 属性的集合
            _____ map = stu.getAttributes();
            //循环输出 map 中所有的属性
            for (int k = 0; k < map.getLength(); k++) {
                _____ attrNode = (_____)map.item(k);
                System.out.print(attrNode.getName() + "="
                    + attrNode.getValue() + " ");
            }
            //获得 student 标记下所有子节点集合
        }
    }
}
```



```

NodeList eNl = stu.getChildNodes();
//遍历所有子节点
for (int j = 0; j < eNl.getLength(); j++) {
    _____ child = eNl.item(j);
    if (child.getNodeType() == child.ELEMENT_NODE) {
        Element eNode = (Element)child;
    }
    if (child.getNodeType() == child.TEXT_NODE) {
        _____ tNode = (_____)child;
    }
}
}
}
}

```