

面向服务的软件工程选择题整理

1. 关于 SOA，以下哪些描述是正确的？ A

- A. 三角形 SOA 操作模型涉及的角色是：服务请求者，服务提供者和服务注册表
- B. 服务可以非正式地分类为任务服务，实体服务和实用服务。其中，只有任务服务可以作为服务组合实现
- C. 在面向服务的情况下，软件开发分为三个阶段：平台开发、服务开发和应用开发。应用程序开发可以由业务专家使用服务组合来完成
- D. Web 服务是一种面向服务的编程语言，用于构建 SOA 中的服务
- E. 只要自包含、平台/实现无关且可重用，本地或网络软件组件都可以被视为服务

解析：

- A. 三角形 SOA 操作模型涉及三个角色：服务请求者、服务提供者和服务注册表。服务请求者向服务提供者发送特定服务的请求，而服务注册表提供可供服务请求者使用的可用服务目录。
- B. 服务可以被分类为不同类型，包括任务服务、实体服务和实用程序服务。然而，所有类型的服务都可以作为服务组合实现。
- C. 在面向服务的体系结构中，软件开发被划分为两个阶段，服务开发 and 应用程序开发。应用程序开发可以由业务专家通过服务组合来完成。
- D. Web 服务不是一种编程语言，而是在互联网上不同软件系统之间进行通信的标准化方式。
- E. 服务是自治、开放、自描述、与实现无关的网络构件。

2. 以下哪个选项是在给定 XML 应用中不使用文档对象模型（DOM）的最佳原因？ C

- A. 应用需要进行复杂处理
- B. 应用可以是非验证的
- C. 应用受到内存使用的重大限制
- D. 架构需要使用属性

解析：

- A. 如果应用需要进行复杂处理，则 DOM 可能是最好的选择，因为它提供了广泛的 API 和功能。
- B. 是否对应用进行验证不会影响 DOM 的使用。
- C. 如果应用受到内存使用的重大限制，则 DOM 可能不是最好的选择，因为它需要将整个文档加载到内存中进行处理。
- D. DOM 可以处理具有属性的 XML 文档，因此这不是不使用 DOM 的原因。

3. XSL 样式表是 XML 文档吗？ A

- A. 是
- B. 不是
- C. 取决于头部
- D. 只有当它被应用于 XML 文档时

解析：XSL 样式表（XSL Stylesheet）是一种基于 XML 语言的文档，用于定义 XML 文档的显示方式和转换方式。它是一种 XML 文档，其中包含有关如何显示或转换 XML 文档的指令和规则。

XSL 样式表包含了用于生成另一种 XML 文档的模板和指令。通过使用 XSL 样式表，可以将 XML 文档转换为其他格式，例如 HTML、PDF 等。XSL 样式表使用 XSLT（XSL 转换）语言来转换 XML 文档，同时使用 XPath 语言来选择和处理 XML 文档中的数据。

4. 在 SOAP 消息中，以下哪个部分在 `env:Envelope` 中是必须的？ (`xmlns:env="http://www.w3.org/2003/05/soap-envelope"`) B

- A. env: Header B. env: Body C. env: Fault D. A 和 B 都是必须的

解析: Envelope 元素表示 SOAP 消息结构的根, 其包括一个强制的 Body 结构和一个可选的 Header 结构。

5. 关于 SOAP, 以下哪些说法是不正确的? D

- A. SOAP 是一种消息交换协议, 提供无状态、单向消息交换范式, 但可以用于创建更复杂的交互模式。
B. 在 SOAP 处理模型中, 定义了两种类型的中介: 转发中介和主动中介。
C. 在 SOAP 错误消息中, env:body 的直接子元素应为 env:fault。
D. 在 SOAP 处理模型中, SOAP 头用于携带增值特性的信息, 可以由最终接收者和/或中介处理。SOAP 正文用于携带从初始发送方到最终接收方的有效负载, 可以被最终接收者和/或中介访问, 但只能由最终接收者修改。

解析: 在 SOAP 处理模型中, 定义了两种中介类型:

- 转发中介: 它们负责在 SOAP 消息的路由路径上将消息转发到下一个接收者, 而不对消息进行处理或修改。
- 主动中介: 它们负责对 SOAP 消息进行处理和/或修改, 然后将其转发到下一个接收者。它们可以根据消息内容执行复杂的操作, 如安全处理、路由决策、负载均衡等。

6. 在 WSDL 中, 以下哪个元素不能是 description 元素的直接子元素 E

- A. types B. interface C. binding D. service E. endpoint F. import

解析: endpoint 元素是服务端点的定义, 是 service 元素的子元素。

7. 关于 WSDL, 哪一个说法是不正确的? C

- A. WSDL 文档用于定义服务, 从某个 WSDL 文档派生的服务具有相似的功能。
B. 接口 (interface) 和操作 (operation) 元素用于定义服务的抽象能力, 而绑定 (binding)、端点 (endpoint) 和服务 (service) 元素用于将接口链接到具体的消息和部署信息。
C. 在一个接口中, 操作可以在一个命名空间下共享相同的名称, 只要它们具有不同的输入/输出消息的定义。这被称为“重载”。
D. 由于错误消息被接口中的操作共享, 因此定义了元素 fault 作为元素 Interface 的直接子元素, 并通过名称引用。
E. 除了 XML 模式, 我们还可以在 WSDL 中使用一些其他模式定义语言。

解析: 在 WSDL 中, 操作名称必须在接口中唯一, 无论它们的输入/输出消息定义是否相同。这是因为操作名称将用作 Web 服务的公共 API, 不允许使用重载。

8. 关于 Code 1, 以下哪个说法不正确? B¹

- A. Header block 事务只需要由最终接收方处理。如果最终接收方失败, 则应生成一个 SOAP 故障消息。
B. 由于 Header block 事务定义了事务 ID, 因此响应消息 (如果有) 必须包含该 Header block, 否则我们无法实现 SOAP 消息的相关性, 因此应生成 SOAP 故障消息。
C. 可以添加多个 Header block 以支持更多特性。
D. Element Body 中可以包含多个直接子元素。

9. Code 1 中显示的 SOAP 消息, 最有可能是以下哪种类型的消息? B

¹答案给的是 B, 但查了半天的说法都是“如果 SOAP 消息包含带有事务 ID 的标头块, 则它用于关联事务中的请求和响应消息。因此, 响应消息必须包含具有相同事务 ID 的相同标头块。如果响应消息不包含相同的头块, 则意味着响应与原始请求无关, 无法实现关联。在这种情况下, 生成 SOAP 错误消息适合指示响应无效或意外。”

- A. 基于 XML 的内容交换消息
B. RPC 请求消息
C. RPC 响应消息
D. 错误消息

code 1. SOAP message Example

```

1 POST /Reservations?code=FT352BQ HTTP/1.1
2 Host: travelcompany.example.org
3 content-Type: application/soap+xml; charset="utf-8"
4 Content-Length: nnnn
5 <?xml version='1.0'?>
6 <env: Envelope xmlns: env="http://www.w3.org/2003/05/soap-envelope"/>
7   <env: Header>
8     <t:transaction
9       xmlns:t="http://thirdparty.example.org/transaction"
10      env:encodingstyle="http://example.com/encoding"
11      env:mustUnderstand="true">5</t:transaction>
12   </env: Header>
13
14   <env: Body>
15     <m:chargeReservation
16       env:encodingstyle="http://www.w3.org/2003/05/soap-encoding"
17       xmlns:m="http://travelcompany.example.org/">
18       <m:reservation
19         xmlns:m="http://travelcompany.example.org/reservation">
20         <m:code>FT35ZBQ</m:code>
21       </m:reservation>
22       <o:creditCard xmlns:o="http://mycompany.example.com/financial">
23         <n:name xmlns:n="http://mycompany.example.com/employees">
24           Åke Jógvan dyvind
25         </n:name>
26         <o:number>123456789099999</o:number>
27         <o:expiration>2005-02</o:expiration>
28       </o:creditCard>
29     </m:chargeReservation>
30   </env:Body>
31 </env:Envelope>

```

10. 关于 Code 1, 以下哪个说法不正确? C

- A. 发送此 SOAP 消息的发送方应该期望一个 HTTP 响应消息。由于 HTTP 是面向连接的, 所以在 HTTP 响应消息中不需要地址信息。
B. 寻址方法是由 HTTP 规范定义的, 而不是 SOAP 规范。
C. 如果在 HTTP 地址 URI 中使用参数 code=FT352BQ 来标识资源, 则该 SOAP 消息是 Web 架构兼容的。SOAP 规范为 URI 计算指定了一个默认算法。
D. 属性 encodingstyle 用于定义元素的序列化方案。它的值是特定于应用程序的决定, 并且假设互操作的能力已经“带外 (out-of-band)”解决

解析: SOAP 规范没有指定用于 URI 计算的默认算法。相反, 它提供了构建与 Web 体系结构兼容的 URI 的指南。计算 URI 的实际算法取决于特定的实现和被识别的资源。

11. 关于 Code 2, 以下哪个说法不正确? B

- A. 元素操作中的属性模式给出了此操作的 MEP（消息交换模式）。由于采用 URI 标识，MEP 不受 WSDL 2.0 预定义的 8 种类型的限制。
- B. 如果元素操作中没有属性模式，则该操作的 MEP 由子元素的存在和顺序确定（输入，输出，故障，故障输出）。
- C. 符号“#any”表示操作“检索 (retrieve)”接收任何单个元素作为输入。
- D. “reservationDetails.xsd”定义消息类型“list:reservationList”。如果更改消息类型，则 WSDL 文档可能仍然相同。
- E. 元素描述可以包含多个接口元素，只要它们具有不同的名称。

解析：如果元素操作中没有属性模式，则该操作的 MEP 由正在使用的 WSDL 版本的默认消息交换模式决定，对于 WSDL 1.1 是请求-响应，对于 WSDL 2.0 是输入-输出。

code 2. WSDL Example

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <description xmlns="http://www.w3.org/ns/wsdl"
3     targetNamespace="http://greath.example.com/2004/services/reservationList"
4     xmlns:tns="http://greath.example.com/2004/services/reservationList"
5     xmlns:list=mhttp://greath.example.com/2004/schemas/reservationList" ...>
6     <types>
7         <xs:import
8             namespace=mhttp://greath.example.com/2004/schemas/reservationList"
9             schemalocation="reservationlist.xsd" />
10    </types>
11    <interface name="reservationListInterface">
12        <operation name="retrieve"
13            pattern="http://www.w3.org/ns/wsdl/in-out" wsdlx:safe="true">
14            <input messageLabel="In" element="#any" />
15            <output messageLabel="out" element="list:reservationList" />
16        </operation>
17    </interface>
18    <binding name="reservationListSOAPBinding"
19        interface="tns:reservationListInterface"
20        type="http://www.w3.org/ns/wsdl/soap"
21        wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">
22        <operation ref="tns:retrieve"
23            wsoap:mep="http://www.w3.org/2003/05/soap/mep/request-response" />
24    </binding>
25    <service name="reservationListService"
26        interface="tns:reservationListInterface">
27        <endpoint name="reservationListEndpoint"
28            binding="tns:reservationListSOAPBinding"
29            address="http://greath.example.com/2004/reservationList" />
30    </service>
31 </description>

```

12. 关于 Code 2，以下哪个说法不正确？ B

- A. 在 WSDL 扩展中定义的属性 wsdlx:safe="true" 表示此操作不会以任何方式强制客户端。在此示例中，我们将使用 HTTP POST 调用这些操作。
- B. 元素 service 定义了 interface 和 endpoint 间的一对一对应关系。元素 service 只能有一个 endpoint 作为子元素。

- C. 元素 `service` 中的某些信息（例如属性 `address`）只能在服务部署后确认。
- D. 在此示例中，元素 `binding` 是不可重用的。在可重用的绑定中，应省略属性 `interface` 和特定于操作和故障的绑定详细信息。

解析：虽然元素 `service` 确实定义了 `interface` 和 `endpoints`² 之间的一对一关系，但它不限于只有一个 `endpoint` 作为子元素。`service` 元素可以有多个 `endpoint`，每个 `endpoint` 对应于同一接口的不同绑定。这允许客户根据他们的要求选择合适的 `endpoint`。

13. 关于 Code 2，以下哪个说法是正确的？ C

- A. 属性 `schemaLocation="reservationList xsd"` 用于定位 schema。如果找不到 schema，将生成一个 SOAP 故障。
- B. 由于未在接口 `reservationListInterface` 中定义故障，因此服务请求者将不会接收任何 SOAP 故障消息。
- C. 此示例中的所有属性 `messageLabel` 都可以省略。
- D. 由于在操作中 `retrieve` 使用了“in-out” MEP，因此在 SOAP-HTTP 绑定中使用 SOAP 请求-响应 MEP 是唯一的选择。

解析：

- A. 如果找不到 `schemaLocation` 属性中指定的 XSD 文件，Web 服务可能无法正常运行，但不一定会生成 SOAP 错误。
- B. 如果在 WSDL 中没有为特定操作定义故障，服务仍将能够生成 SOAP 故障消息以响应错误，但客户端将无法理解故障，因为在 WSDL。因此，最好在 WSDL 中为所有操作定义故障，以便客户端能够正确理解 and 处理错误。
- C. 正确，对于只有一个输入/输出的情况，`messageLabel` 可以省略。
- D. 也可以使用 SOAP 响应 MEP，使用 HTTP GET 方法在 HTTP 请求中返回一个包含 SOAP 消息的 HTTP 响应的正文。

14. UDDI 中不包括下列哪个元素？ D

- A. `bussinessEntity` C. `bindingTemplate` E. `publisherAssertion`
B. `businessService` D. `TechnicalService`

解析：UDDI 包括五个主要元素，并使用 XML Schema 来正式表达其数据结构

- `businessEntity`: 商业实体信息及其提供的服务
- `businessService`: 商业实体所提供的服务
- `bindingTemplates`: 如何调用一个服务
- `tModel (Technical Models)`: 特定概念和结构（主要是对于绿页的抽象）
- `publisherAssertion`: 表达商业关系

15. 考虑以下 XML 代码 `<paragraph> Click <link> here </link> to apply </paragraph>`，如果使用 SAX 解析器（一种流式 XML 解析器），以下事件将按照什么顺序发生？ B

- | | | |
|------------------------------------|-------------------------------|-------------------------------|
| 1. 开始元素-名称: <code>paragraph</code> | 4. 结束元素-名称: <code>link</code> | 7. 字符串: <code>to apply</code> |
| 2. 结束元素-名称: <code>paragraph</code> | 5. 字符串: <code>Click</code> | |
| 3. 开始元素-名称: <code>Link</code> | 6. 字符串: <code>here</code> | 8. 结束字符串 |
- A. 1、5、7、3、6、4、2 C. 1、5、6、7、3、4、2
B. 1、5、3、6、4、7、2 D. 1、5、8、3、6、8、4、7、8、2

²`endpoint` 是指将服务绑定到特定网络地址的实例。它定义了如何访问服务，包括网络协议、网络地址和消息格式等。`Endpoint` 需要指定一个绑定（`binding`），以确保客户端能够正确地与服务进行交互。在一个服务中，可以定义多个 `endpoint`，每个 `endpoint` 都指向服务的不同实例。