

## Examen terminal

### Architectures n-tiers : Intergiciels à objets et services web

Tous documents autorisés.

Le barème est donné à titre indicatif.

#### 1 Corba (5 points)

Un fichier IDL vous est donné au listing 1. On vous donne aussi un fichier d'implémentation au listing 2, ainsi qu'un serveur et un client aux listings 3 et 4.

Listing 1 – foobar.idl

```
1 module foobar{
2
3 exception ExceptionA { string text; };
4
5 // an object of type Bar has an internal state s of type short
6 interface Bar {
7     // increments the internal state s and returns it
8     short m3();
9 } ;
10
11 interface Foo {
12     // systematically throws an exception ExceptionA
13     void m1 () raises (ExceptionA);
14     // returns a newly created object of type Bar, with internal state initialized with the
15         value i
16     Bar m2(in short i) ;
17 } ;
18 };
```

Listing 2 – Barimpl.java

```
1 package foobar;
2 public class Barimpl extends BarPOA {
3
4     private short s;
5
6     public Barimpl(short s){
7         this.s=s;
8     }
9     public short m3 (){
10         s++;
11         return s;
12     }
13 }
```

## Listing 3 – Server.java

---

```

1 import org.omg.CORBA.*;
2 import org.omg.PortableServer.*;
3 import org.omg.PortableServer.POA;
4
5 public class Server {
6
7     public void run (String args[]) {
8         try {
9             ORB orb = ORB.init(args, null);
10            POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
11            rootpoa.the_POAManager().activate();
12            foobar.Fooimpl foo = new foobar.Fooimpl ();
13            org.omg.CORBA.Object ref = rootpoa.servant_to_reference(foo);
14            foobar.Foo href = foobar.FooHelper.narrow(ref);
15            System.out.println(orb.object_to_string(href));
16            System.err.println("Java Server ready and waiting ...");
17            orb.run();
18        } catch (Exception e) {
19            System.err.println("ERROR: " + e);
20            e.printStackTrace(System.out);
21        }
22    }
23
24    public static void main (String args[]) {
25        Server srv = new Server ();
26        srv.run (args);
27    }
28 }

```

---

## Listing 4 – Client.java

---

```

1 import org.omg.CORBA.*;
2
3 public class Client {
4
5     public void run (String args[]) {
6         try {
7             ORB orb = ORB.init(args, null);
8             org.omg.CORBA.Object ref = orb.string_to_object (args[0]);
9             foobar.Foo href = foobar.FooHelper.narrow(ref);
10
11             short s=0;
12             foobar.Bar bar = href.m2(s);
13             System.out.println(bar.m3());
14             System.out.println(bar.m3());
15
16         } catch (Exception e) {
17             System.err.println("ERROR: " + e);
18             e.printStackTrace(System.out);
19         }
20
21     }
22
23     public static void main (String args[]) {
24         Client cli = new Client ();
25         cli.run (args);
26     }
27 }

```

---

**Question 1.** On veut changer la communication existante de la référence vers l'objet distant du serveur vers le client, en utilisant le service de nommage de Corba.

**a-** Apportez les modifications nécessaires.

**b-** Expliquez comment on doit désormais lancer le serveur et le client ainsi que tout ce qui peut être nécessaire à leur bon fonctionnement.

**c-** On lance le serveur puis le client. Que s'affiche t-il côté client et côté serveur ?

**Question 2.** Donnez en Java une implémentation de l'interface Foo.

**Question 3.** Expliquer précisément à quoi sert la méthode **narrow** des classes Helper. Donnez un exemple précis d'une utilisation d'une telle méthode.

## 2 RMI (4 points)

On veut écrire la même application foobar que celle écrite en Corba précédemment (avec la version utilisant le service de nommage). Notez bien qu'il n'est pas nécessaire d'avoir répondu aux questions sur CORBA pour réussir cette exercice.

**Question 4.** Donner l'ensemble du code Java nécessaire.

**Question 5.** Expliquez comment lancer le serveur et le client ainsi que tout ce qui peut être nécessaire à leur bon fonctionnement.

**Question 6.** Il y a un service de nommage en RMI. Expliquez comment le démon correspondant doit être lancé à l'extérieur d'un serveur de manière à ce que le bind d'un serveur fonctionne (et notamment pour ne pas rencontrer d'exception lors du bind).

## 3 .NET remoting et RMI (3 points)

**Question 7.** Expliquez précisément la ou les différences entre RMI et .NET remoting concernant la mise à disposition d'objets distants par un serveur. Il s'agit ici de discuter, non pas de la manière d'écrire des classes d'objets distants, mais de la mise à disposition des instances. On abordera de vrais différences : il ne s'agit pas de pointer des différences syntaxiques par exemple.

**Question 8.** Donnez un avantage et un inconvénient à donner les paramètres d'un serveur Remoting dans un fichier de configuration plutôt que directement dans le code. Il ne s'agit pas ici de pointer la difficulté à écrire un tel fichier ni de discuter de la lisibilité du XML.

## 4 Services web et WCF (9.5 points)

**Question 9.** On veut écrire la même application foobar que celle écrite en Corba et en RMI précédemment. Est-ce possible simplement ? Si oui comment, sinon pourquoi ?

**Question 10.** Expliquez pourquoi on donne comme anti-pattern de signature de service web : `DataSet Query(String requete)` ?

Soit l'extrait de fichier WSDL donné au listing 5.

Listing 5 – extrait d'un fichier WSDL

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tm="http://
  microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/
  encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://services.
  aonaware.com/webservices/" xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http:
  //schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  targetNamespace="http://services.aonaware.com/webservices/" xmlns:wsdl="http://schemas.
  xmlsoap.org/wsdl/">
3   <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Word Dictionary Web
    Service</wsdl:documentation>
4   <wsdl:types>
5     <s:schema elementFormDefault="qualified" targetNamespace="http://services.aonaware.com/
      webservices/">
6       <s:element name="StrategyList">
7         <s:complexType />
8       </s:element>
9       <s:element name="StrategyListResponse">
10        <s:complexType>
11          <s:sequence>
12            <s:element minOccurs="0" maxOccurs="1" name="StrategyListResult" type="
              tns:ArrayOfStrategy" />
13          </s:sequence>
14        </s:complexType>
15      </s:element>
16      <s:complexType name="ArrayOfStrategy">
17        <s:sequence>

```

```

18         <s:element minOccurs="0" maxOccurs="unbounded" name="Strategy" nillable="true" type=
19             "tns:Strategy" />
20     </s:sequence>
21 </s:complexType>
22 <s:complexType name="Strategy">
23     <s:sequence>
24         <s:element minOccurs="0" maxOccurs="1" name="Id" type="s:string" />
25         <s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string" />
26     </s:sequence>
27 </s:complexType>
28 <s:element name="Match">
29     <s:complexType>
30         <s:sequence>
31             <s:element minOccurs="0" maxOccurs="1" name="word" type="s:string" />
32             <s:element minOccurs="0" maxOccurs="1" name="strategy" type="s:string" />
33         </s:sequence>
34     </s:complexType>
35 </s:element>
36 <s:element name="MatchResponse">
37     <s:complexType>
38         <s:sequence>
39             <s:element minOccurs="0" maxOccurs="1" name="MatchResult" type="
40                 tns:ArrayOfDictionaryWord" />
41         </s:sequence>
42     </s:complexType>
43 </s:element>
44 <s:complexType name="ArrayOfDictionaryWord">
45     <s:sequence>
46         <s:element minOccurs="0" maxOccurs="unbounded" name="DictionaryWord" nillable="true"
47             type="tns:DictionaryWord" />
48     </s:sequence>
49 </s:complexType>
50 <s:complexType name="DictionaryWord">
51     <s:sequence>
52         <s:element minOccurs="0" maxOccurs="1" name="DictionaryId" type="s:string" />
53         <s:element minOccurs="0" maxOccurs="1" name="Word" type="s:string" />
54     </s:sequence>
55 </s:complexType>
56 <s:element name="string" nillable="true" type="s:string" />
57 <s:element name="ArrayOfDictionary" nillable="true" type="tns:ArrayOfDictionary" />
58 <s:element name="WordDefinition" nillable="true" type="tns:WordDefinition" />
59 <s:element name="ArrayOfStrategy" nillable="true" type="tns:ArrayOfStrategy" />
60 <s:element name="ArrayOfDictionaryWord" nillable="true" type="tns:ArrayOfDictionaryWord"
61     />
62 </s:schema>
63 </wsdl:types>
64 <wsdl:message name="StrategyListSoapIn">
65     <wsdl:part name="parameters" element="tns:StrategyList" />
66 </wsdl:message>
67 <wsdl:message name="StrategyListSoapOut">
68     <wsdl:part name="parameters" element="tns:StrategyListResponse" />
69 </wsdl:message>
70 <wsdl:message name="MatchSoapIn">
71     <wsdl:part name="parameters" element="tns:Match" />
72 </wsdl:message>
73 <wsdl:message name="MatchSoapOut">
74     <wsdl:part name="parameters" element="tns:MatchResponse" />
75 </wsdl:message>
76 <wsdl:portType name="DictServiceSoap">
77     <wsdl:operation name="StrategyList">
78         <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Return list of all
79             available strategies on the server</wsdl:documentation>
80         <wsdl:input message="tns:StrategyListSoapIn" />
81         <wsdl:output message="tns:StrategyListSoapOut" />
82     </wsdl:operation>
83     <wsdl:operation name="Match">
84         <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Look for matching
85             words in all dictionaries using the given strategy</wsdl:documentation>
86         <wsdl:input message="tns:MatchSoapIn" />
87         <wsdl:output message="tns:MatchSoapOut" />
88     </wsdl:operation>
89 </wsdl:portType>
90 <wsdl:binding name="DictServiceSoap" type="tns:DictServiceSoap">
91     <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
92     <wsdl:operation name="StrategyList">
93         <soap:operation soapAction="http://services.aonaware.com/webservices/StrategyList" style
94             ="document" />

```

```

88     <wsdl:input>
89         <soap:body use="literal" />
90     </wsdl:input>
91     <wsdl:output>
92         <soap:body use="literal" />
93     </wsdl:output>
94 </wsdl:operation>
95 <wsdl:operation name="Match">
96     <soap:operation soapAction="http://services.aonaware.com/webservices/Match" style="
97         document" />
98     <wsdl:input>
99         <soap:body use="literal" />
100     </wsdl:input>
101     <wsdl:output>
102         <soap:body use="literal" />
103     </wsdl:output>
104 </wsdl:operation>
105 </wsdl:binding>
106 <wsdl:service name="DictService">
107     <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Word Dictionary Web
108         Service</wsdl:documentation>
109     <wsdl:port name="DictServiceSoap" binding="tns:DictServiceSoap">
110         <soap:address location="http://services.aonaware.com/DictService/DictService.asmx" />
111     </wsdl:port>
112 </wsdl:service>
113 </wsdl:definitions>

```

**Question 11.** Donnez sous forme de signatures de méthodes les opérations décrites dans l'extrait de ce fichier.

**Question 12.** Le service reçoit ce message SOAP :

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.
3     org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
4     <soap:Body>
5         <Match xmlns="http://services.aonaware.com/webservices/">
6             <word>wsdl</word>
7             <strategy>suffix</strategy>
8         </Match>
9     </soap:Body>
10 </soap:Envelope>

```

Quel message SOAP renvoie-t-il sachant que le mot *wsdl* n'apparaît avec la stratégie *suffix* que dans le dictionnaire d'identifiant *Vera* et y a pour définition : *WSDL Web Services Description Language (WWW)*.

On ne donnera pas l'entête http mais juste le contenu de l'enveloppe.

**Question 13.** Comment écrire un client en C# pour ce service qui affiche sur la sortie standard la liste de toutes les stratégies puis interroge le service de manière à ce que le message SOAP de la question précédente soit émis ?

**Question 14.** Supposons que ce service ait été développé avec WCF. Comment aurait-on procédé ? Donnez le genre de contrats de service et de données à développer, ainsi que les bindings à utiliser.

**Question 15.** Expliquez comment est gouvernée l'instanciation des objets de service en WCF.

**Question 16.** Les classes de proxy se mettent-elles en place statiquement ou dynamiquement quand on consomme des services web en C# ? Même question avec des services WCF. Justifiez vos réponses.