

Cognome e Nome		Matricola	
----------------	--	-----------	--

Esercizio 1

Si consideri un sistema costituito da:

- **N Client**;
- **1 Registro**.

Il *Registro* fornisce un servizio di risoluzione di *nomi di servizi* in *indirizzi IP* e *numeri di porta*. A tal fine, il *Registro* memorizza un insieme di triple *<nome servizio, indirizzo IP, porta>*. Il *Registro* fornisce il proprio servizio tramite il protocollo UDP sulla porta 2301. L'applicazione *Client* invia al *Registro*, tramite datagrammi, una richiesta contenente il nome del servizio (es. "MeteoService"), ed il *Registro* risponde con una stringa contenente indirizzo IP e la porta del servizio separati da un trattino (es. "160.97.24.200 – 1500"). Se il servizio richiesto non è presente in elenco, il *Registro* restituisce la stringa "unknown service requested".

Il *Client* deve effettuare le seguenti operazioni:

1. Legge da tastiera un *servicename*.
2. Chiede all'applicazione *Registro* (che si suppone essere in esecuzione sull'host "registro.server.it") tramite protocollo UDP (come specificato sopra) di risolvere il *servicename* in *indirizzo IP* e *porta*.
3. Si connette col protocollo TCP all'indirizzo ip e alla porta restituita dal *Registro* ed invia il messaggio "Test message"; una volta ricevuta una stringa di risposta dal servizio, la stampa su video e si disconnette.

Si chiede di implementare la classe *Client*, la classe *Registro* ed un main che avvia un Registro e 10 Client su localhost. Non è richiesto di implementare i servizi, né di inizializzare l'insieme di triple memorizzate nel Registro.

Esercizio 2

Si descriva, anche mediante figure opportunamente commentate, il funzionamento del **protocollo HTTP**.

Esercizio 3

Si descriva, anche mediante figure opportunamente commentate, come funziona la **firma digitale**.

Esercizio 4

Si consideri il file WSDL allegato relativo ad un *Web Service* che fornisce informazioni meteorologiche. In particolare, il Web Service espone un metodo che riceve il CAP di una località (es. "87036") e restituisce le condizioni meteorologiche attuali, rappresentate da una coppia *<condizione, temperatura>*, dove condizione è una stringa (es. "sereno", "nuvoloso", "piovoso"). Si implementi in Java una classe che implementa il servizio e si descrivano le principali fasi necessarie alla sua messa in opera.

Allegato all'esercizio 4

```
- <wsdl:definitions targetNamespace="http://localhost:8080/axis/services/PrevisioneMeteo">
-   <!--
-       WSDL created by Apache Axis version: 1.2.1
-       Built on Jun 14, 2005 (09:15:57 EDT)
-   -->
-   <wsdl:types>
-       <schema targetNamespace="http://DefaultNamespace">
-           <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
-           <complexType name="CondizioneMeteorologica">
-               <sequence>
-                   <element name="condizione" nillable="true" type="xsd:string"/>
-                   <element name="temperatura" type="xsd.double"/>
-               </sequence>
-           </complexType>
-       </schema>
-   </wsdl:types>
-   <wsdl:message name="condizioneRequest">
-       <wsdl:part name="in0" type="soapenc:string"/>
-   </wsdl:message>
-   <wsdl:message name="condizioneResponse">
-       <wsdl:part name="condizioneReturn" type="tns1:CondizioneMeteorologica"/>
-   </wsdl:message>
-   <wsdl:portType name="PrevisioneMeteo">
-       <wsdl:operation name="condizione" parameterOrder="in0">
-           <wsdl:input message="impl:condizioneRequest" name="condizioneRequest"/>
-           <wsdl:output message="impl:condizioneResponse" name="condizioneResponse"/>
-       </wsdl:operation>
-   </wsdl:portType>
-   <wsdl:binding name="PrevisioneMeteoSoapBinding" type="impl:PrevisioneMeteo">
-       <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
-       <wsdl:operation name="condizione">
-           <wsdlsoap:operation soapAction=""/>
-           <wsdl:input name="condizioneRequest">
-               <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace" use="encoded"/>
-           </wsdl:input>
-           <wsdl:output name="condizioneResponse">
-               <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/services/PrevisioneMeteo" use="encoded"/>
-           </wsdl:output>
-       </wsdl:operation>
-   </wsdl:binding>
-   <wsdl:service name="PrevisioneMeteoService">
-       <wsdl:port binding="impl:PrevisioneMeteoSoapBinding" name="PrevisioneMeteo">
-           <wsdlsoap:address location="http://localhost:8080/axis/services/PrevisioneMeteo"/>
-       </wsdl:port>
-   </wsdl:service>
- </wsdl:definitions>
```