DIMES UNIVERSITÀ DELLA CALABRIA
Dipartimento di **Ingegneria Informatica, Modellistica, Elettronica e Sistemistica**

Esame di **Reti di Calcolatori** del Corso di Laurea in Ingegneria Informatica DM270

Prova scritta del 24 novembre 2016 – Durata: 2 ore e 30 minuti – Orario di consegna: _____

Cognome e nome:	Matricola:

Esercizio 1

Si realizzi un sistema di calcolo distribuito per l'affitto di risorse di elaborazione, costituito da:

1. *M* applicazioni *Elaboratore* (entità che vuole offrire le proprie risorse di calcolo);
2. *1* applicazione *Gestore*;
3. *N* applicazioni *Client* (entità che è interessata all'utilizzo delle risorse di calcolo).

Una *Risorsa* è caratterizzata da un nome, da un tipo (hardware/software) e da una descrizione testuale. Inizialmente, ogni Elaboratore invia al Gestore un'istanza di *OffertaRisorsa* (contenente il *nome*, il *tipo* e la *descrizione* della risorsa offerta). Il Gestore memorizza in un'opportuna struttura dati tutte le offerte ricevute. Successivamente, ogni Client invia al Gestore un'istanza di *RichiestaRisorsa* (contenente il *tipo* e la *descrizione* della risorsa cercata). A questo punto, il Gestore cerca tra le descrizioni memorizzate quella che soddisfa i criteri specificati. Terminata la ricerca, il Gestore risponderà al Client indicando l'indirizzo di rete del Elaboratore che possiede la risorsa cercata.

Si richiede di realizzare in Java le applicazioni *Elaboratore*, *Gestore* e *Client*, nonché le classi *OffertaRisorsa* e *RichiestaRisorsa*.

Seguono maggiori dettagli sulle operazioni che devono essere eseguite dall'applicazione Gestore:

- Usa la porta TCP 3000 per ricevere una sequenza di *M* connessioni da parte dei server remoti *Elaboratore* (ciascuno comunicante la risorsa offerta mediante l'invio di un'istanza di *OffertaRisorsa*). **Non è** richiesto di gestire connessioni multiple. Si supponga che la costante *M* sia nota all'applicazione Gestore. Il Gestore memorizza opportunamente tutte le istanze di *OffertaRisorsa* ricevute.
- Usa la porta TCP 2000 per ricevere una sequenza di *N* connessioni da parte delle applicazioni *Client* (che comunicano le risorse richieste). In questo caso, **è** richiesto di gestire connessioni multiple. Si supponga che la costante *N* sia nota all'applicazione Gestore.
- Per ogni RichiestaRisorsa *r* ricevuta, verifica quale OffertaRisorsa *o* soddisfa *r* (valutando una corrispondenza esatta tra i tipi e le descrizioni). Nel caso in cui ci siano più offerte che soddisfano una richiesta, si sceglie la prima che viene trovata. Restituisce ad ogni Client l'indirizzo del Elaboratore che possiede la risorsa cercata.

Esercizio 2

Si descriva, anche mediante figure opportunamente commentate, come possono essere indicizzati i file in una rete peer-to-peer.

Esercizio 3

Si descriva, anche mediante figure opportunamente commentate, come funziona il controllo del flusso nel protocollo TCP.

Esercizio 4

Si realizzi un Web Service che permette di ottenere alcune informazioni sugli esami di un corso di laurea.
In particolare, il servizio espone:

1. un metodo che, dati il nome di un esame, e la matricola di uno studente che ha partecipato all'esame, restituisce il voto conseguito da quello studente all'esame (un intero compreso tra 1 e 30).
2. un metodo che, data una data, restituisce il nome dell'esame previsto in quella data (si assuma che si tenga un solo esame al giorno).

come specificato nel file WSDL allegato.

Allegato all'esercizio 4

```
<wsdl:definitions targetNamespace="http://www.examples.com/wsdl/ExamsService">
<wsdl:types>
<schema targetNamespace="http://DefaultNamespace">
<import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
<complexType name="Data">
        <sequence>
                <element name="Giorno" type="xsd:int"/>
                <element name="Mese" type="xsd:int"/>
                <element name="Anno" type="xsd:int"/>
        </sequence>
<complexType>
<schema>
<wsdl:types>
        <wsdl:message name="VotoStudenteRequest">
                <wsdl:part name="in0" type="xsd:string"/>
                <wsdl:part name="in1" type="xsd:int"/>
        </message>
        <wsdl:message name="VotoStudenteResponse">
                <wsdl:part name="in0" type="xsd:int"/>
        </message>
        <wsdl:message name="EsameGiornoRequest">
                <wsdl:part name="in1" type="tns1:Data"/>
        </message>
        <wsdl:message name="EsameGiornoResponse">
                <wsdl:part name="in0" type="xsd:string"/>
        </message>
<wsdl:portType name="ExamsService">
        <wsdl:operation name="VotoStudente" parameterOrder="in0 in1">
        <wsdl:input message="impl:VotoStudenteRequest" name=VotoStudenteRequest/>
        <wsdl:output message="impl:VotoStudenteResponse" name=VotoStudenteResponse/>
</operation>
<wsdl:operation name="EsameGiorno" parameterOrder="in0">
        <wsdl:input message="impl:EsameGiornoRequest" name= EsameGiornoRequest/>
        <wsdl:output message="impl:EsameGiornoResponse" name= EsameGiornoResponse/>
</operation>
</wsdl:portType>
<wsdl:binding ...>
...
</wsdl:binding>
<wsdl:service ...>
...
</wsdl:service>
</wsdl:definitions>
```