

```
/* Svolgimento Compito 1, 3, 5, 7 - 21/12/05 */
```

```
import java.io.BufferedReader;
```

```
public class Client {
```

```
    public static void main(String[] args) {
```

```
        final int REGISTRYPORT = 1099;
```

```
        String registryHost = null;
```

```
        String serviceName = "ServerRMI";
```

```
        BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));
```

```
        // Controllo dei parametri della riga di comando
```

```
        if (args.length != 1) {
```

```
            System.out.println("Sintassi: Client NomeHost");
```

```
            System.exit(1);
```

```
        }
```

```
        registryHost = args[0];
```

```
        // Impostazione del SecurityManager
```

```
        if (System.getSecurityManager() == null) {
```

```
            System.setSecurityManager(new RMISecurityManager());
```

```
        }
```

```
        // Connessione al servizio RMI remoto
```

```
        try {
```

```
            String completeName = "/" + registryHost + ":" + REGISTRYPORT + "/" +  
                + serviceName;
```

```
            RemOp serverRMI = (RemOp) Naming.lookup(completeName);
```

```
            System.out
```

```
                .println("ClientRMI: Servizio \"" + serviceName + "\" connesso");
```

```
            System.out.println("\nRichieste di servizio fino a fine file");
```

```
            String service;
```

```
            String nomeStanza, nomeUtente, stato;
```

```
            System.out.print("Servizio (AS=aggiungi stanza,EU=elimina utente, "
```

```
                + "AU=aggiungi utente, VU=visualizza utente, ES=elimina stanza): ");
```

```
            while ((service = stdIn.readLine()) != null) {
```

```
                if (service.equals("AS")) {
```

```
                    System.out.print("Inserisci il nome della stanza : ");
```

```
                    nomeUtente = stdIn.readLine();
```

```
                    System.out
```

```
                        .print("Inserisci il tipo della stanza: (M)ulticast o (P)unto-a-punto");
```

```
                    stato = stdIn.readLine();
```

```
                    // Controllo input
```

```
                    while (!stato.equals("M") && !stato.equals("P")
```

```
                        && !stato.equals("SM") && !stato.equals("SP"))
```

```
                        stato = stdIn.readLine();
```

```
                    try {
```

```
                        if (serverRMI.aggiungi_stanza(nomeUtente, stato)) {
```

```
                            System.out.println("Aggiunta la stanza " + nomeUtente  
                                + " , di tipo " + stato);
```

```
                        } else {
```

```
                            System.out.println("Aggiunta stanza non riuscita");
```

```
                        }
```

```
                    }
```

```
                    catch (RemoteException re) {
```

```
                        System.out.println("Errore remoto: " + re);
```

```
                    }
```

```

} // AS

else if (service.equals("EU")) {
    System.out.print("Inserisci il nome dell'utente da eliminare: ");
    nomeUtente = stdIn.readLine();
    Stanza[] ris = null;
    try {
        ris = serverRMI.elimina_utente(nomeUtente);
        if (ris == null) {
            System.out.println("Nessun utente eliminato");
        } else {
            System.out
                .println("Stampo le stanze nelle quali l'utente e' stato
eliminato:");
            for (int i = 0; i < ris.length; i++)
                System.out.println("" + ris[i]);
        }
    }
    catch (RemoteException re) {
        System.out.println("Errore remoto: " + re);
    }
} // EU

else if (service.equals("AU")) {
    System.out
        .print("Inserisci il nome della stanza nella quale aggiungere l'utente:
");

    nomeStanza = stdIn.readLine();
    System.out.print("Inserisci il nome dell'utente da aggiungere: ");
    nomeUtente = stdIn.readLine();
    boolean ris;
    try {
        ris = serverRMI.aggiungi_utente(nomeStanza, nomeUtente);
        if (ris) {
            System.out.println("Utente aggiunto correttamente");
        } else {
            System.out.println("Aggiunta utente non riuscita");
        }
    }
    catch (RemoteException re) {
        System.out.println("Errore remoto: " + re);
    }
} // AU

else if (service.equals("VU")) {
    System.out.print("Inserisci il nome dell'utente da visualizzare: ");
    nomeUtente = stdIn.readLine();
    Stanza[] ris = null;
    try {
        ris = serverRMI.visualizza_utente(nomeUtente);
        if (ris == null) {
            System.out.println("Nessuna stanza contiene l'utente");
        } else {
            System.out.println("Stampo le stanze che contengono l'utente:");
            for (int i = 0; i < ris.length; i++)
                System.out.println("" + ris[i]);
        }
    }
    catch (RemoteException re) {
        System.out.println("Errore remoto: " + re);
    }
} // VU

else if (service.equals("ES")) {
    System.out.print("Inserisci il nome della stanza da eliminare: ");
    nomeStanza = stdIn.readLine();

```

```

        boolean ris;
        try {
            ris = serverRMI.elimina_stanza(nomeStanza);
            if (ris) {
                System.out.println("Stanza eliminata correttamente");
            } else {
                System.out.println("Eliminazione stanza non riuscita");
            }
        }
        catch (RemoteException re) {
            System.out.println("Errore remoto: " + re);
        }
    } // ES

    else
        System.out.println("Servizio non disponibile");

    System.out.print("Servizio (AS=aggiungi stanza,EU=elimina utente, "
        + "AU=aggiungi utente, VU=visualizza utente, ES=elimina stanza): ");

    } // !EOF

}
catch (Exception e) {
    System.err.println("ClientRMI: " + e.getMessage());
    e.printStackTrace();
    System.exit(1);
}
}
}

```