

```
// Implementazione del Client RMI
```

```
import java.rmi.*;
```

```
class RMI_Client {
```

```
    // Avvio del Client RMI
```

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

```
        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 && args.length != 2) {
```

```
            System.out.println("Sintassi: ClientFile NomeHost [registryPort]");
```

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

```
        } else {
```

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

```
            if (args.length == 2) {
```

```
                try {
```

```
                // Aggiungere anche controllo porta valida, nel range di quelle usabili
```

```
                registryPort = Integer.parseInt(args[1]);
```

```
            } catch (Exception e) {
```

```
                System.out
```

```
                    .println("Sintassi: ClientFile NomeHost [registryPort],
```

```
registryPort intero");
```

```
                System.exit(2);
```

```
            }
```

```
        }
```

```
    }
```

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

```
    try {
```

```
        String completeName = "/" + registryHost + ":" + registryPort + "/"
```

```
            + serviceName;
```

```
        RMI_interfaceFile serverRMI = (RMI_interfaceFile)
```

```
Naming.lookup(completeName);
```

```
        System.out.println("Client RMI: Servizio \" + serviceName
```

```
            + "\" connesso");
```

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

```
        String service;
```

```
        System.out
```

```
            .print("Servizio (C=conta il numero delle righe, L=lista nomi file
```

```
con parola): ");
```

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

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

```
                String nomeFile;
```

```
                String parola;
```

```
                System.out.print("Nome file? ");
```

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

```
                System.out.print("Parola? ");
```

```
                // Prendo il primo della linea
```

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

```
                // Invocazione remota
```

```
                try {
```

```
                    int numeroRighe = serverRMI.conta_numero_linee(nomeFile,
```

```
parola);
```

```
                    if( numeroRighe<0 ) System.out.println("File non esistente o parola non trovata");
```

```

        else System.out.println("Il file " + nomeFile + " contiene la parola
cercata,"
        + " ed e' composto da " + numeroRighe + " righe");
    } catch (RemoteException re) {
        System.out.println("Errore remoto: " + re.toString());
    }

    } // C

    else if (service.equals("L")) {
        String nomeDir;
        String parola;
        System.out.print("Nome direttorio? ");
        nomeDir = stdin.readLine();
        System.out.print("Parola? ");
        // Prendo il primo della linea
        parola = stdin.readLine();

        String[] res;
        try {
            res =
serverRMI.lista_nomi_file_contenenti_parola_in_linea(nomeDir, parola);
            if( res==null ) System.out.println("Problemi: il file richiesto non e' un
direttorio o non esiste");
            else{
                System.out.println("Ho trovato "+res.length+" file contenenti la parola
cercata, i seguenti:");
                for(int i=0; i<res.length; i++)
                    System.out.println(res[i]);
            }
        } catch (RemoteException re) {
            System.out.println("Errore remoto: " + re.toString());
        }

    } // L

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

    System.out
        .print("Servizio (C=conta occorrenze, L=lista nomi file con
carattere): ");
    } // !EOF

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

```