

# Lucidi Consentiti per lo scritto di Sistemi di Elaborazione in rete

## Socket

- Creare un oggetto **ServerSocket** (lato server), specificando il numero di porta  
**ServerSocket serv\_socket = new ServerSocket(port);**

- Chiamare **accept()** (lato server) per restare in attesa di connessioni  
**Socket cli\_socket = serv\_socket.accept();**

- Creare un oggetto di classe **Socket** (lato client), specificando indirizzo e porta  
**Socket sock = new Socket("myhost.dsi.it", 1234);**  
Oppure specificando l'indirizzo IP:  
**Socket sock = new Socket("150.145.63.148", 1234);**

- Recuperare uno stream associato ad un socket (con variabile chiamata *sock*, come nell'esempio precedente):

```
InputStream istream = sock.getInputStream();  
OutputStream ostream = sock.getOutputStream();
```

- Costruire uno stream di caratteri sullo stream del socket

1) in output

```
PrintWriter out = new PrintWriter (ostream,true);  
out.println( "Ciao a tutti" );  
// si usa esattamente come la System.out.println
```

2) in input

```
BufferedReader in = new BufferedReader (new InputStreamReader(istream));  
String s=in.readLine();// legge una linea di testo dal socket
```

- Costruire uno stream di dati (int, float, ecc.) sullo stream del socket

1) in output

```
DataOutputStream out = new DataOutputStream (ostream);  
out.writeInt(4); // scrive un intero
```

2) in input

```
DataInputStream in = new DataInputStream(istream);  
int x=in.readInt(); // legge un intero
```

- Chiudere un socket  
**sock.close();**

- Chiudere gli stream  
**in.close();**  
**out.close();**

# Thread

1. Creazione di una classe che estende Thread:

```
class Nome extends Thread {  
    public Nome (...) {  
        ...  
    }  
    public void run () {  
        ...  
    }  
}
```

2. Creazione e avvio di un Thread:

```
Nome n=new Nome(...);  
n.start();
```

3. Attesa del thread per un tempo casuale:

```
try {  
    sleep ((int)(Math.random()*1000)); //tra 0 ed 1 secondo  
}catch (InterruptedException e) {}
```

# Sincronizzazione

1. Attesa:

```
try{  
    wait();  
}catch(InterruptedException e){}
```

2. Notifica:

```
notifyAll();  
notify();
```