

Software Architectures



Networking
Mauro Migliardi Ph. D.



A word of wisdom

- ▶ The word *networking* strikes fear in the hearts of many programmers. Fear not! Using the networking capabilities provided in the Java environment is quite easy.
- ▶ Ci credete? Incredibile, ma **vero**.



Networking

- ▶ Package java.net
- ▶ Networking in Java e' basato sullo stack TCP/IP
 - ▶ Internet oriented
 - ▶ OSI? Chi era costui?
- ▶ Accesso a TCP e UDP
- ▶ TCP == Stream
- ▶ UDP == datagram



java.net

- ▶ non da' accesso a protocolli di livello piu' basso
 - ▶ (e.g. IP)
- ▶ da' accesso a protocollo di livello piu' alto
 - ▶ (e.g. HTTP)
- ▶ supporto SSL con package presenti in distribuzione standard
 - ▶ Javax.net.ssl



Classi del package java.net

- ▶ Le principali sono
 - ▶ InetAddress
 - ▶ Inet4Address, Inet6Address
 - ▶ SocketAddress
 - ▶ InetSocketAddress
 - ▶ Socket
 - ▶ ServerSocket
 - ▶ DatagramSocket
 - ▶ DatagramPacket
 - ▶ MulticastSocket
 - ▶ **DatagramSocketImpl**
 - ▶ **SocketImpl**



Classe InetAddress

- ▶ La classe incapsula le informazioni necessarie a identificare un computer su Internet
 - ▶ nome
 - ▶ indirizzo IP
- ▶ non ha costruttore
 - ▶ metodi static `getByName` e `getAllByName`



Classe InetAddress

▶ **Constructor Summary**

- ▶ *InetAddress(InetAddress addr, int port)*
Creates a socket address from an IP address and a port number. *InetAddress(int port)*
Creates a socket address where the IP address is the wildcard address and the port number a specified value.
- ▶ *InetAddress(String hostname, int port)*
Creates a socket address from a hostname and a port number.

▶ **Method Summary**

- ▶ *static InetAddress createUnresolved(String host, int port)*
Creates an unresolved socket address from a hostname and a port number.
- ▶ *boolean equals(Object obj)*
Compares this object against the specified object.
- ▶ *InetAddress getAddress()*
Gets the InetAddress.
- ▶ *String getHostName()*
Gets the hostname.
- ▶ *int getPort()*
Gets the port number.
- ▶ *boolean isUnresolved()*
Checks whether the address has been resolved or not.



Classe SocketImpl

- ▶ classe astratta che definisce le funzionalita' richieste ad un socket
- ▶ praticamente la libreria socket del C
- ▶ le classi concrete non danno accesso a tutte
- ▶ solo ai sottoinsiemi specifici per l'implementazione
 - ▶ e.g. Socket != DatagramSocket



Classe Socket

- ▶ e' la classe concreta che incapsula il socket di tipo stream
 - ▶ solo AF_Inet, non ci sono socket AF_UNIX
- ▶ e' solo socket client
- ▶ permette di iniziare una connessione ad un host su un port
- ▶ permette di specificare interfaccia e port locali



Classe Socket (Cont.)

- ▶ Inizialmente supportava solo opzioni
 - ▶ dimensione buffers
 - ▶ opzione linger
 - ▶ opzione no-delay
 - ▶ opzione bloccante con timeout
- ▶ La versione corrente supporta molte piu' opzioni
 - ▶ [JavaDocs](#)



ServerSocket

- ▶ e' il socket che permette di accettare connessioni
 - ▶ supporta accept
- ▶ permette di specificare port e interfaccia locale
 - ▶ binding
- ▶ permette di specificare il numero di connessioni "appese"
 - ▶ listen (backlog)



DatagramSocket

- ▶ socket per il protocollo UDP
 - ▶ connectionless
- ▶ permette di specificare il port locale
- ▶ permette di specificare l'interfaccia locale
- ▶ supporta connect
 - ▶ per limitare la destinazione cui si inviano o da cui si ricevono datagram



MulticastSocket

- ▶ estende il DatagramSocket aggiungendo la capacita' di ascoltare su gruppi
 - ▶ join
 - ▶ leave
 - ▶ indirizzi multicast (classe D da 224.0.0.1 a 239.255.255.255)
- ▶ aggiunge i metodi per gestire Time To Live dei datagram



Connessione di sockets

- ▶ due socket (stream) connessi costituiscono un canale bidirezionale
- ▶ coppia `InputStream/OutputStream`
- ▶ agli stream di basso livello si possono sovrapporre tutti gli stream del package di I/O
 - ▶ XDR
 - ▶ stream di caratteri
 - ▶ Oggetti!!!



Serializzazione su rete

- ▶ `ObjectInputStream`
- ▶ `ObjectOutputStream`
 - ▶ ATTENZIONE: l'ordine di apertura e' importante
- ▶ Due coppie di `ObjectStream` costituiscono un canale full duplex su cui e' possibile inviare oggetti "completi"
- ▶ overhead, performance inferiori agli stream non OO
 - ▶ molto dipendente dalla banda di memoria
 - ▶ asimmetrico: ricezione piu' pesante della trasmissione



Programmazione Client/Server

- ▶ datagram
- ▶ byte stream
- ▶ XDR
- ▶ caratteri
- ▶ oggetti