

Java Sockets Programming

- The package `java.net` provides support for sockets programming (and more).
- Typically you import everything defined in this package with:

```
import java.net.*;
```

Classes

```
InetAddress  
Socket  
ServerSocket  
DatagramSocket  
DatagramPacket
```

InetAddress class

- static methods you can use to create new `InetAddress` objects.
 - `getByName(String host)`
 - `getLocalHost()`

```
InetAddress x = InetAddress.getByName(  
    "monte.cs.rpi.edu");
```

More InetAddress methods

```
byte[] getAddress();      // raw IP  
String getHostAddress();  // text (dd)  
String getHostName();     // text
```

Sample Code: Lookup.java

- Uses `InetAddress` class to lookup hostnames found on command line.

```
> java Lookup monte www.yahoo.com  
monte:128.213.8.110  
www.yahoo.com:64.58.76.176
```

Socket class

- Corresponds to active TCP sockets only!
 - client sockets
 - socket returned by `accept()`;
- Passive sockets are supported by a different class: `ServerSocket`
- UDP sockets are supported by `DatagramSocket`

Socket Constructors

- Constructor creates a TCP connection to a named TCP server.

– There are a number of constructors:

```
Socket(InetAddress server, int port);
```

```
Socket(InetAddress server, int port,  
      InetAddress local, int localport);
```

```
Socket(String hostname, int port);
```

Socket Methods

```
void close();  
InetAddress getInetAddress(); getpeername  
InetAddress getLocalAddress(); getsockname  
InputStream getInputStream();  
OutputStream getOutputStream();
```

- Lots more (setting/getting socket options, partial close, etc.)

Socket I/O

- Socket I/O is based on the Java I/O support (in the package `java.io`).
- `InputStream` and `OutputStream` are abstract classes
 - common operations defined for all kinds of `InputStreams`, `OutputStreams`...

InputStream Basics

```
// reads some number of bytes and  
// puts in buffer array b  
int read(byte[] b);
```

```
// reads up to len bytes  
int read(byte[] b, int off, int len);
```

Both methods can throw `IOException`.

Both return `-1` on EOF.

OutputStream Basics

```
// writes b.length bytes  
void write(byte[] b);
```

```
// writes len bytes starting  
// at offset off  
void write(byte[] b, int off, int  
          len);
```

Both methods can throw `IOException`.

ServerSocket Class (TCP Passive Socket)

- Constructors:

```
ServerSocket(int port);
```

```
ServerSocket(int port, int backlog);
```

```
ServerSocket(int port, int backlog,  
            InetAddress bindAddr);
```

ServerSocket Methods

```
Socket accept();  
  
void close();  
  
InetAddress getInetAddress();  
  
int getLocalPort();  
  
throw IOException, SecurityException
```

Netprog 2002 Java Sockets

13

Sample Echo Server

TCPEchoServer.java

Simple TCP Echo server.

Based on code from:
[TCP/IP Sockets in Java](#)

Netprog 2002 Java Sockets

14

UDP Sockets

- DatagramSocket class
- DatagramPacket class needed to specify the payload (incoming or outgoing).

Netprog 2002 Java Sockets

15

DatagramSocket Constructors

```
DatagramSocket();  
  
DatagramSocket(int port);  
  
DatagramSocket(int port, InetAddress a);
```

All can throw SocketException or
SecurityException.

Netprog 2002 Java Sockets

16

Datagram Methods

```
void connect(InetAddress, int port);  
  
void close();  
  
void receive(DatagramPacket p);  
  
void send(DatagramPacket p);
```

Lots more!

Netprog 2002 Java Sockets

17

DatagramPacket

- Contain the payload (a byte array).
- Can also be used to specify the destination address (when not using connected mode UDP).

Netprog 2002 Java Sockets

18

DatagramPacket Constructors

For receiving:

```
DatagramPacket( byte[] buf, int len);
```

For sending:

```
DatagramPacket( byte[] buf, int len  
    InetAddress a, int port);
```

Netprog 2002 Java Sockets

19

DatagramPacket methods

```
byte[] getData();  
void setData(byte[] buf);  
  
void setAddress(InetAddress a);  
void setPort(int port);  
  
InetAddress getAddress();  
int getPort();
```

destination address

could be either address (depends on context)

Netprog 2002 Java Sockets

20

Sample UDP code

UDPEchoServer.java

Simple UDP Echo server.

Test using nc as the client (netcat):

```
> nc -u hostname port
```

Netprog 2002 Java Sockets

21