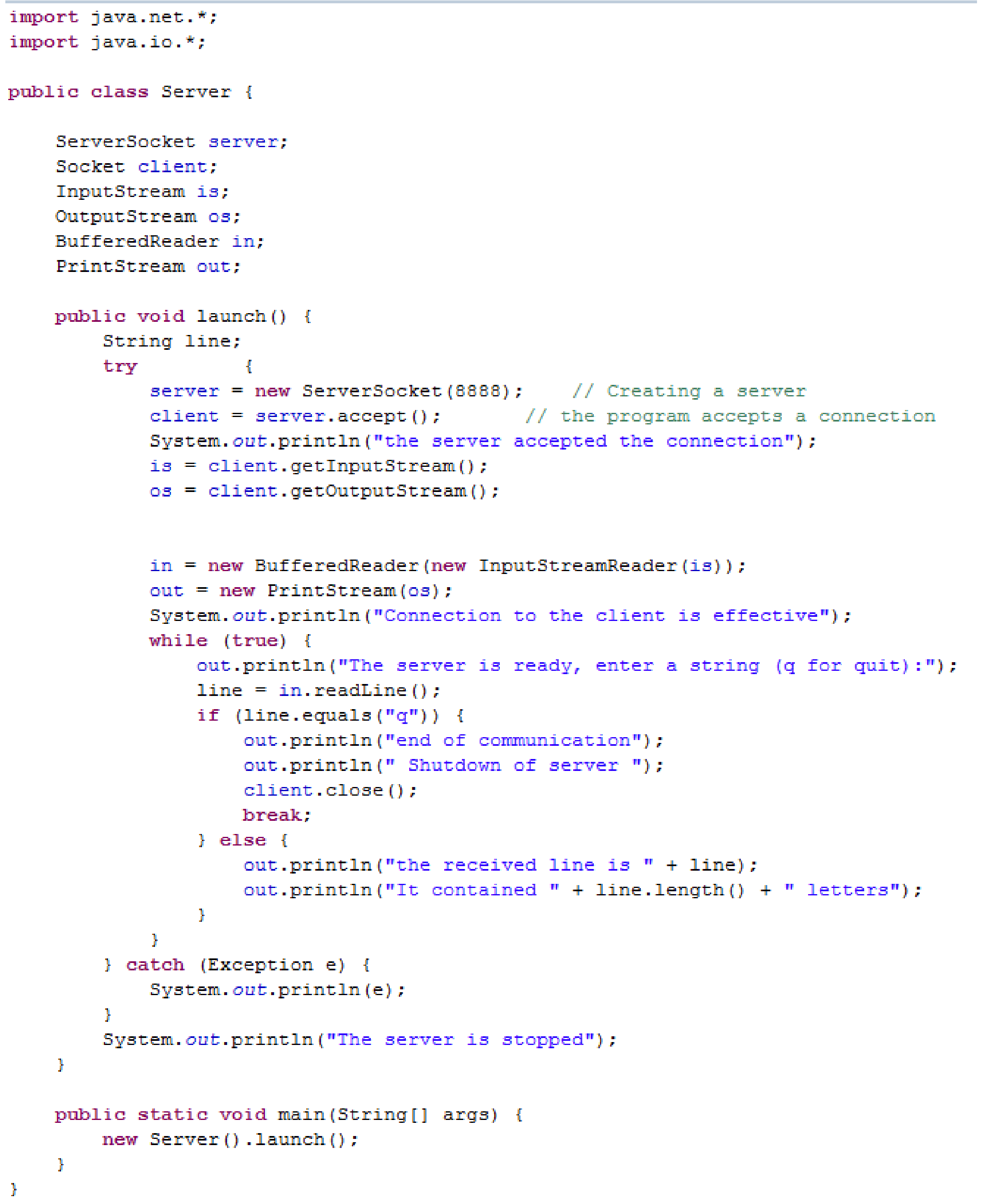
Topic: Network

Slides: network

Resources: http://patriot.net/~tvalesky/easyrmi.html

Code: echo.zip, timeserver.zip

Hand-in None



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

public class Server {

    ServerSocket server;
    Socket client;
    InputStream is;
    OutputStream os;
    BufferedReader in;
    PrintStream out;

    public void launch() {
        String line;
        try         {
            server = new ServerSocket(8888);    // Creating a server
            client = server.accept();        // the program accepts a connection
            System.out.println("the server accepted the connection");
            is = client.getInputStream();
            os = client.getOutputStream();


            in = new BufferedReader(new InputStreamReader(is));
            out = new PrintStream(os);
            System.out.println("Connection to the client is effective");
            while (true) {
                out.println("The server is ready, enter a string (q for quit):");
                line = in.readLine();
                if (line.equals("q")) {
                    out.println("end of communication");
                    out.println(" Shutdown of server ");
                    client.close();
                    break;
                } else {
                    out.println("the received line is " + line);
                    out.println("It contained " + line.length() + " letters");
                }
            }
        } catch (Exception e) {
            System.out.println(e);
        }
        System.out.println("The server is stopped");
    }

    public static void main(String[] args) {
        new Server().launch();
    }
}
```

Explanation

The java.net package includes everything concerning the access to networks. The most commonly used classes are as follows:

`URL`: represents an Internet URL

`Socket` and `ServerSocket`: enable TCP/IP connections according to the Socket interface

`DatagramPacket` and `DatagramServer`: determine type UDP connections.

`InetAddress`: represents a TCP/IP address

A server program expects that the client program requests a connection, and then, when the connection is established by the Socket class object, the server program responds to requests from the client.

The `ServerSocket` class

The `java.net package` contains the `ServerSocket` class that represents a server. It is associated with a port greater than 1024 identification number (the other numbers are reserved).

Constructor

Have constructed an object of class `ServerSocket` indicating its port number.

```
Server = new ServerSocket (port);
```

Method `accept()`

After the creation of the server, the accept method is used to wait for connection attempts by clients. When a connection is detected, the accept method returns an object of the Socket class that will establish dialogue through input and output streams.

Exercise 1

Understand and test the server program using a telnet client

(`telnet MachineName port`).

Exercise 2

Write the class `Server2` so that the server waits for another client connection after each closing connection.

Exercise 3

Write the class `Client` that will replace the `telnet`. You can use the `Socket` class to connect to the server.

Exercise 4

Start 2 clients at the same time. What is going on? Write the class `Server4` which can handle several clients at once (using `Threads`).

Exercise 5

Using previous classes, write a program (`Server5` and `Client5`) communication with several clients at the same time. (There will be a client and a server program).

A message sent to the server will be sent to all connected clients.