Prepared By Abhiyan Dev Rimal
https://github.com/abhirimal/be-software-notes

PAGE :
DATE : / /

Chap 7

Network Programming

-> The term network programming refers to writing programs that execute across multiple devices (computers) in which the devices are all connected ~~using~~ to each other using a network.

The java.net package provides support for the two common network protocols.

1) TCP (Transmission Control Protocol)
-> It allows reliable communication between two applications.
-> It is used over the Internet Protocol which is referred to as TCP/IP

2) UDP (User Datagram Protocol)
-> It is a connection-less protocol
-> It allows for packets of data to be transmitted between applications.

Java Networking
-> Java Networking is a concept of connecting two or more computing devices together so that we can share resources.
-> Socket Programming provides facility to share data between the different computing devices.

Scanned with CamScanner

host, servername, IP Address are same.

PAGE:
DATE: / /

Chapter-7 (10) - 8 or 15
Network Programming

1) What is an URL processing? URL Class. ~~Example~~

~~OR~~

~~What is using the URL~~

Ans: The URL class is the gateway to any of the resource available on internet. A class URL represents a Uniform Resource Locator, which is a pointer to a "resource" on the World Wide Web. A resource can point to a simple file or directory, or it can ~~be~~ refer to a more complicated object such as a query to a database or a search engine.

Commonly used URL class

Method	
public String getProtocol()	Returns the protocol of the url.
public String getHost()	Returns the hostname
public String getPort()	Returns the port number of the url
public String getFile()	Returns the file name of the url

Scanned with CamScanner

PAGE :
DATE : / /

2. WAP for reading contents in a given URL.
or
WAP using the URL class to show URL processing in Java.

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

public class URLDemo {
    ~~public class~~
    public static void main(String[] args) {
        try {
        URL url = new URL("https://www.oracle.com/
                                index.html")
        System.out.println("Protocol: " + url.getProtocol());
        System.out.println("Host Name:" + url.getHost());
        System.out.println("Port Number:" + url.getPort());
        System.out.println("File Name :" + url.getFile());
        }
        catch (Exception e) {
            System.out.println(e); }
        }
}
```

Extra: url.getQuery
url.getRef
url.getContent

Output:
Protocol: https
Host Name: www.oracle.com
Port Number: -1
File Name: /index.html

Scanned with CamScanner

PAGE :
DATE : / /

3. URL connection class. Example

Ans: The Java URL connection class represents a communication link between the URL and the application. This class can be used to read and write data to the specified resource referred by the URL.

The openConnection() method of URL class returns the object of URLConnection class.

Syntax:

```
public URLConnection openConnection() throws
                                IOException { }
```

Example:

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

public class URLDemo {
    public static void main(String[] args) {
    try {
        URL url = new URL("https://youtube.com");
        URLConnection urlcon = url.openConnection();
        InputStream input = urlcon.getInputStream();
        int i;
        while ((i = input.read()) != -1) {
            System.out.print((char) i);
        }
    } catch (Exception e) {
        System.out.println(e);
    }
    }
}
```

(# This code displays the content of youtube.com)

Scanned with CamScanner

PAGE :
DATE : / /

4) Difference between URL and URL connection class.

URL class	URLConnection class
1. A URL class is used to represent the location of a resource.	1. URL Connection class is used to represen the communication link between the URL and application.
2. It provides an abstraction of a URL the World Wide Web's pointer.	2. It specifies how to reach a resource.
3.	3.

System.out.println("Local host name" + addr.getHostName());

Scanned with CamScanner

PAGE :
DATE : / /

5) Explain Inet Address class. WAP to find local IP address & hostname of the system

Ans Java Inet class represents IP address. Inet address class provides method to get the IP address of any host name.

There are no constructors for this class but static methods which returns instances of Inet Address class for general use.

Methods	Description
getHostName()	returns the host name of ip address.
getHostAddress()	returns the ip address in string format
getLocalHost()	returns the IP address of local host

```
// Program to find local IP address & hostname of
   the system
import java.net.*;
public class Example {
    public static void main (String[] args)
    throws Exception {
        InetAddress addr = InetAddress.getLocalHost();
        System.out.println("Local Host Address " +
                                addr.getHostAddress());
        String hostname = addr.getHostName();
        System.out.println("Local host name: " +hostname);
    }
}
```

Output: Local Host Address: 192.168.1.4
Local host name: RamSingh

PAGE :
DATE : / /

Not asked

Q) WAP to display Hostname and IP address in Java.

```
import java.net.*;
public class Demo {
    public static void main(String[] args) {
        try {
        InetAddress address = InetAddress.getByName
                ("www.youtube.com");
        System.out.println(address);
        System.out.println("Host name :" + ip.getHostName());
        System.out.println("IP address :" + ip.getHostAddress());
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Output:

Host Name: www.youtube.com
Host Address: 206.51.231.148

Scanned with CamScanner

PAGE :
DATE : / /

6) Define socket programming & its usage in java with eg.

Ans: ⇒ Java Socket Programming is used for communication between ~~two~~ the applications running on different JRE.
→ Java socket programming can be connection-oriented or connection-less.

Socket provides the communication mechanism between two computers using TCP. A client program creates a socket on its end of the ~~program~~ communication and attemps to connect the socket to a server. When the connection is made, the server creates a socket object on its end of the communication.
The client and the server can now communicate by writing to and reading from the socket.

After the connections are ~~made~~ established communication can ~~be~~ occurs using I/O streams. Each socket has both an Output Stream and an Input Stream. The client's Output Stream is connected to the server's Input Stream, and the client's Input Stream is connected to the Server's Output Stream.
TCP is a two-way communication protocol, hence data can be sent across both streams at the same time.

Scanned with CamScanner

PAGE :
DATE : / /

Port	Socket
1. Port is a numeric value that is assigned to an application in an end point of communication.	1. Socket is an internal end point for sending & receiving data within a node on a computer network.
2. Helps to identify a specific application or a process.	2. It work as an interface to send and receive data through specific port.

TCP/IP ~~Server~~ Client/Server Steps

1. Create a socket. ~~to the~~
2. Bind an IP address and port to the socket
3. Listen for connections if the application is a server using a stream socket.
4. Establish connection if the application is client using a stream socket.
5. Exchange data
6. Close the socket when done.

Scanned with CamScanner

PAGE :
DATE : / /

7) Create a TCP client/Server programs which client & server communicate by sending a message to each other.

- Server.java

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

public class Server {
    public static void main(String[] args) {
    try {
    ServerSocket ss=new ServerSocket(4000);
    System.out.println("Waiting for Connection...");

    Socket s=ss.accept();
    System.out.println("Client Connected");

    InputStreamReader inr= new InputStreamReader
                        (s.getInputStream());
    BufferedReader bfr= new BufferedReader(inr);

    String str= bfr.readLine();
    System.out.println("Client:" +str);

    PrintWriter pr=new PrintWriter(s.getOutputStream());
    pr.println("Hey Client");

    pr.flush();
```

Scanned with CamScanner

PAGE :
DATE : / /

```
bf.close();
s.close();
ss.close();
} catch (IOException e) {
        System.out.println(e);
    }
}
}
```

- Client.java

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

public class Client {
  public static void main (String[] args) {
  try {
    Socket s = new Socket ("localhost", 4000);
    PrintWriter pr = new PrintWriter (s.getOutputStream());
    pr.println ("Hello Server");
    pr.flush();
  InputStreamReader inr = new InputStreamReader
                        (s.getInputStream());
  BufferedReader bfr = new BufferedReader (inr);
    String str = bfr.readline();
    System.out.println ("Server:" +str);
    s.close();
  } catch (IOException e) {
        System.out.println (e);
    }
  }
}
```

Scanned with CamScanner

PAGE :
DATE : / /

8) Difference between TCP & UDP

	TCP	UDP
Acronym	Transmission Control Protocol.	User Datagram Protocol.
Connection	TCP is connection oriented.	UDP is connectionless protocol.
Usage	TCP is suited for applications that require ~~by~~ high reliability and transmission time is less critical.	UDP is suitable for applications that need fast efficient transmission such as games.
Used by	HTTP, HTTPS, FTP	DNS
Speed	TCP is slower than UDP.	UDP is faster because error recovery is not attempted.
Reliability	TCP absolute guarantee that the data transferred remains intact and arrives in the same order in which it was sent.	There is no guarantee that the message or packets sent would reach at all.
Weight	It is heavy weight.	It is light weight.

Scanned with CamScanner