

NWK (MI2)

Vorlesung 5

Inhalt

Wiederholung/Verständnisfragen

Thema

LaboraAufgabe war das erstellen eines einfachen Client-Server Programm
Was immer im Internet gemacht wird ist Client-Server Kommunikation

InputStram

- FileInputStream(Byte-Streams in/aus Dateien)
- BufferedInputStream(für gepufferte Ein-/Ausgabe)

Übung 1 enthält einen Link, den man sich als Lesezeichen setzen sollte!

Zwischen den Klassen InputStreamReader und OutputStreamWriter muss eine Konvertierung stattfinden von ByteStream und CharStream

Beispiel:

```
...newBufferedReader(new  
InputStreamReader(newFileInputStream("fileName"),  
Charset.forName("ASCII"));
```

```
newFileInputStream("fileName") → ByteStream  
Charset.forName("ASCII"); → Konvertierung von ByteStream zum  
CharStream
```

Ausführung der EchoServer und EchoClient-Programme zur Demonstration
//EinEchoserver wird z.B. bei der ersten Verwendung von Skype
//verwendet

Vorgehensweise:

Server wartet darauf, dass jemand ihn "anspricht" und wenn das

geschieht, dann gibt er eine Nachricht zurück.

Auf jedem Rechner gibt es ein Loopback-Netz, wo sich nur der eine Rechner drinnen befindet. Die IP-Adresse ist 127.0.0.1 und wird auch als Localhost bezeichnet

Die meisten Internetprotokolle schließen die Zeile mit "\r\n" ab. Um zu überprüfen, ob ein Port noch frei ist, kann man sich sämtliche benutzten Ports in der Konsole mit "netstat -a" anzeigen lassen.

Referenzen

Anmerkungen

Es gibt Bytestreams und CharakterStreams

Lesen und schreiben - InputStream ist eine abstrakte Klasse

Es reicht zu wissen, wie lese ich aus Dateien und wie schreibe ich sie. - mittels gepufferter Ein- und Ausgabe.

Wichtige Streams sind File-Input/Output und BufferedInput/Output - Streams

Beispiel von der Tafel:

Ein gepufferter Stream soll erstellt werden:

```
int b;
```

zuerst wird der InputStream allokiert:

```
FileInputStream fin = new FileInputStream("fileName");
```

Dann wird der BufferedInputStream allokiert und der Fileinputstream als initialisiert:

```
BufferedInputStream bin = new BufferedInputStream(fin);
```

in "b" wird der Rückgabewert von bin.read geschrieben:

```
b = bin.read();
```

Die Allokierung des FileInputStream und des BufferedInputStream wird in üblicher Programmierschreibweise einer Zeile zusammengefasst (chaining):

```
BufferedInputStream bin = new BufferedInputStream(new
```

```
FileInputStream("fileName");
```

Beispiel für eine weitere Zusammenfassung:

Ausgangscode:

```
int a,b;
```

```
b=3;
```

```
a=MathLib.quadrat(b);
```

oder/aber besser ...

```
a= MathLib.quadrat(3);
```

Bei der Verwendung der Portnummer 1234 kann es, zumindest auf WinXP, zu Schwierigkeiten kommen

Alle Ports unter 1024 sind für das System reserviert und daher tabu!