JAVA COM SOCKETS - Complementos

Quando estamos trabalhando com sockets, na verdade estamos implementando um fluxo de dados entre a aplicação cliente e a aplicação servidor. A seguir, vamos estudar diversos conceitos necessários a respeito de fluxos de dados necessários para a boa implementação de sockets em Java.

1. Fluxos

Quando abrimos um socket, deve-se associar um objeto e associar um fluxo. Um fluxo é uma seqüência de bytes. O pacote java.io trata de fluxos de dados

Três Objetos de fluxos são criados automaticamente quando inicia-se a execução de um programa:

- System.out (PrintStream, FilterOutputStream, OutputStream) – saída padrão: console
- System.in (InputStream) - entrada padrão: teclado
- Sytem.err (PrintStream, FilterOutputStream, OutputStream) - saída de erro: console
- Entradas e saídas padrão podem ser redirecionadas

Algumas classes que são interessantes de serem utilizadas:

- InputStream e OutputStream - classes abstratas que definem métodos para realizar entrada e saída respectivamente.
- ByteArrayInputStream e ByteArrayOutputStream - trata as entradas e saídas como arrays de bytes.
- DataInputStream e DataOutputStream - manipular tipos primitivos

Alguns métodos a serem considerados

Métodos de InputStream

- int read() - retorna um byte lido ou -1 indicando fim de arquivo
- int read(byte[]) - como parâmetro retorna os bytes lidos em um array. O valor inteiro retornado é o número de bytes lidos
- void close() - fecha o fluxo
- int avaiable () - retorna o número de bytes que estão disponíveis para leitura
- void skip(long) - permite descartar um número específico de bytes

Métodos de OutputStream

- void write (int) - escreve um byte na saída
- void write (byte[]) - escreve um vetor de bytes na saída
- void close() - fecha o fluxo
- void flush () - força a gravação de tudo

DataInputStream e DataOutputStream

• permite ler e escrever dados primitivos de JAVA. Existem vários métodos.

- Exemplos de métodos:
    - byte readByte() //lê um byte
    - long readLong() //lê um número long
    - double readDouble() //lê um número double
    - int readInt() //lê um número inteiro
    - char readChar() //lê um caracter (char)
    - String readString() //lê uma string
    - String readUTF() //lê uma string
    - void writeByte(byte) //escreve um byte
    - void writeLong(long) //escreve um número long
    - void writeDouble(double) //escreve um número double
    - void writeInt(int) //escreve um número inteiro
    - void writeChar(char) //escreve um caracter (char)
    - void writeString(string) //escreve uma string
    - void writeUTF(string) //escreve uma string