Client

<<Java Class>>
Client
(default package)

- △ sendPacket: DatagramPacket
- △ receivePacket: DatagramPacket
- △ sendReceiveSocket: DatagramSocket
- S F TIMEOUT: int
- □ S input: Scanner
- △ filename: String
- S F fileDirectory: String
- △ mode: String
- □ in: BufferedInputStream
- S F MAX_DATA: int
- △ addr: InetAddress
- △ port: int
- □ req: byte

- ● C Client()
- ● S main(String[]):void
- ● ui():void
- ● connection():void
- ● readReq():void
- ● writeReq():void
- ● S createRequest(byte,String,String):byte[]
- ● createAck(byte):byte[]
- ● createData(byte,byte[]):byte[]
- ● createError(byte,String):byte[]
- ● parseAck(byte[]):void
- ● parseData(byte[]):byte[]
- ● parseError(byte[]):void
- ● send(byte[],InetAddress,int):void
- ● receive():DatagramPacket
- ● processDatagram(DatagramPacket):byte[]
- ● getOpcode(DatagramPacket):Opcode
- ● isValidPacket(DatagramPacket):boolean

ClientConnection

<<Java Class>>
ClientConnection
(default package)

- S F MAX_DATA: int
- △ requestPacket: DatagramPacket
- △ sendPacket: DatagramPacket
- △ receivePacket: DatagramPacket
- △ sendReceiveSocket: DatagramSocket
- S F TIMEOUT: int
- S F fileDirectory: String
- △ filename: String
- △ addr: InetAddress
- △ port: int
- △ error: byte[]
- □ in: BufferedInputStream

- ● C ClientConnection(DatagramPacket)
- ● C ClientConnection(DatagramPacket,byte[])
- ● run():void
- ● threadName():String
- ● readReq():void
- ● writeReq():void
- ● closeConnection():void
- ● getOpcode(DatagramPacket):Opcode
- ● getFilename(DatagramPacket):String
- ● send(byte[]):void
- ● receive():DatagramPacket
- ● processDatagram(DatagramPacket):byte[]
- ● parseAck(byte[]):void
- ● parseData(byte[]):byte[]
- ● parseError(byte[]):void
- ● createAck(byte):byte[]
- ● createData(byte,byte[]):byte[]
- ● createError(byte,String):byte[]
- ● isValidPacket(DatagramPacket):boolean

UserInput

<<Java Class>>
UserInput
(default package)

- □ input: Scanner

- ● C UserInput()
- ● run():void

Opcode

<<Java Enumeration>>
Opcode
(default package)

- S F RRQ: Opcode
- S F WRQ: Opcode
- S F ACK: Opcode
- S F DATA: Opcode
- S F ERROR: Opcode

- ● C Opcode()

Opcode (2)

<<Java Enumeration>>
Opcode
(default package)

- S F RRQ: Opcode
- S F WRQ: Opcode
- S F DATA: Opcode
- S F ACK: Opcode
- S F ERROR: Opcode
- □ F op: byte

- ▲ C Opcode(byte)
- ■ op():byte

Opcode (3)

op 0..1

<<Java Enumeration>>
Opcode
(default package)

- S F RRQ: Opcode
- S F WRQ: Opcode
- S F ACK: Opcode
- S F DATA: Opcode
- S F ERROR: Opcode

- ● C Opcode()

Server

<<Java Class>>
Server
(default package)

- △ receivePacket: DatagramPacket
- △ S receiveSocket: DatagramSocket
- S F MAX_DATA: int

- ● C Server()
- ● S main(String[]):void
- ● S getSocket():DatagramSocket
- ● listener():void
- ● makeConnection(DatagramPacket):void
- ● makeConnection(DatagramPacket,byte[]):void
- ● receive():DatagramPacket
- ● processDatagram(DatagramPacket):byte[]
- ● getOpcode(DatagramPacket):Opcode
- ● isValidPacket(DatagramPacket):boolean
- ● parseError(byte[]):void
- ● createError(byte,String):byte[]