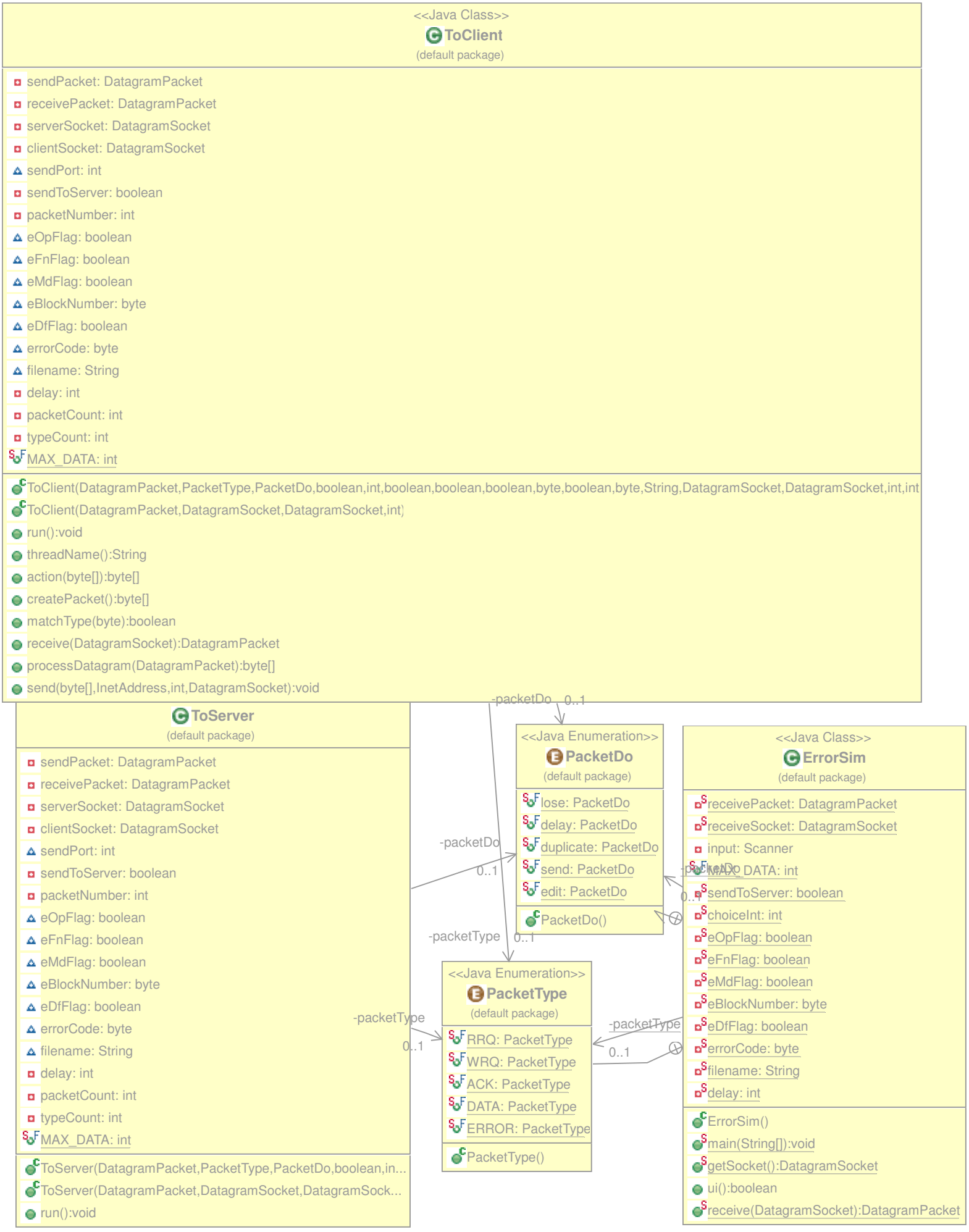
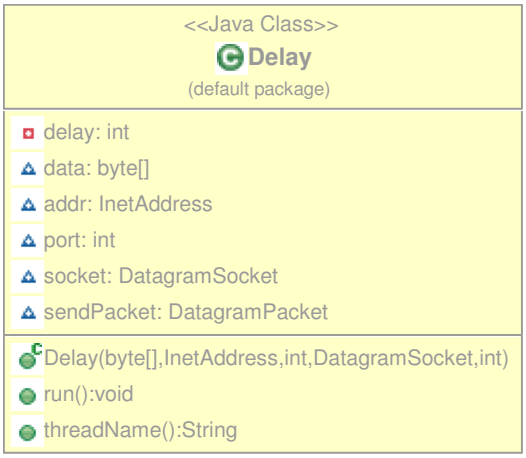
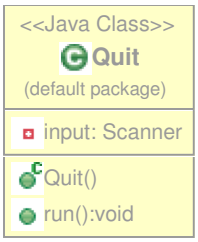

<<Java Class>>
ToClient
(default package)
sendPacket: DatagramPacket
receivePacket: DatagramPacket
serverSocket: DatagramSocket
clientSocket: DatagramSocket
sendPort: int
sendToServer: boolean
packetNumber: int
eOpFlag: boolean
eFnFlag: boolean
eMdFlag: boolean
eBlockNumber: byte
eDfFlag: boolean
errorCode: byte
filename: String
delay: int
packetCount: int
typeCount: int
MAX_DATA: int
ToClient(DatagramPacket,PacketType,PacketDo,boolean,int,boolean,boolean,boolean,byte,boolean,byte,String,DatagramSocket,DatagramSocket,int,int
ToClient(DatagramPacket,DatagramSocket,DatagramSocket,int)
run():void
threadName():String
action(byte[]):byte[]
createPacket():byte[]
matchType(byte):boolean
receive(DatagramSocket):DatagramPacket
processDatagram(DatagramPacket):byte[]
send(byte[],InetAddress,int,DatagramSocket):void
-packetDo
0..1
ToServer
(default package)
sendPacket: DatagramPacket
receivePacket: DatagramPacket
serverSocket: DatagramSocket
clientSocket: DatagramSocket
sendPort: int
sendToServer: boolean
packetNumber: int
eOpFlag: boolean
eFnFlag: boolean
eMdFlag: boolean
eBlockNumber: byte
eDfFlag: boolean
errorCode: byte
filename: String
delay: int
packetCount: int
typeCount: int
MAX_DATA: int
ToServer(DatagramPacket,PacketType,PacketDo,boolean,in...
ToServer(DatagramPacket,DatagramSocket,DatagramSock...
run():void
<<Java Enumeration>>
PacketDo
(default package)
lose: PacketDo
delay: PacketDo
duplicate: PacketDo
send: PacketDo
edit: PacketDo
PacketDo()
-packetDo
0..1
-packetType
0..1
<<Java Enumeration>>
PacketType
(default package)
RRQ: PacketType
WRQ: PacketType
ACK: PacketType
DATA: PacketType
ERROR: PacketType
PacketType()
-packetType
0..1
-packetType
0..1
<<Java Class>>
ErrorSim
(default package)
receivePacket: DatagramPacket
receiveSocket: DatagramSocket
input: Scanner
MAX_DATA: int
sendToServer: boolean
choiceInt: int
eOpFlag: boolean
eFnFlag: boolean
eMdFlag: boolean
eBlockNumber: byte
eDfFlag: boolean
errorCode: byte
filename: String
delay: int
ErrorSim()
main(String[]):void
getSocket():DatagramSocket
ui():boolean
receive(DatagramSocket):DatagramPacket


<<Java Class>>
Delay
(default package)
delay: int
data: byte[]
addr: InetAddress
port: int
socket: DatagramSocket
sendPacket: DatagramPacket
Delay(byte[],InetAddress,int,DatagramSocket,int)
run():void
threadName():String


<<Java Class>>
Quit
(default package)
input: Scanner
Quit()
run():void