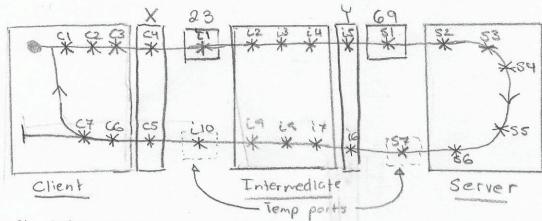
# UCH Diagram:



C1 = mates message cz = printe mossage

e3 = creates message datagram

CY = sends message data gram

CS = receives response

c6 = creates strong & bytes

CT = prints response

51 = receives datagram

52 = creates string of byte EI

53 = prints message

54 = creates response

S5 = prints response

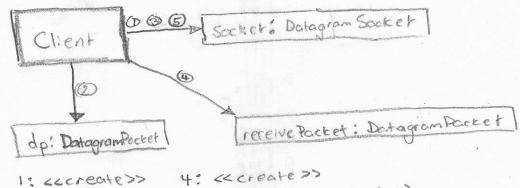
56 = creates response datagran

St = sends response data gram

il = receives message datagran iz = creates string i byte [] 13 = prints message i 4 = recreates datagram is = sends data gram 16 = receives response datagram CI = creates string ; byte [] is = prints responde iq = re-creates datagram i 10 = sends datagrand

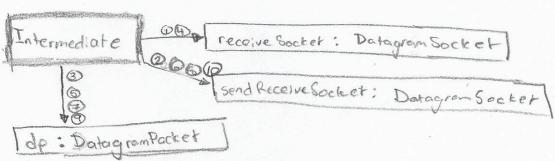
## UML Collaboration Diagrams ;





- 1: Eccreate>>
- s: receivel receive Pocket) 2: << create>>
- 3: send (dp)

### Intermediate:



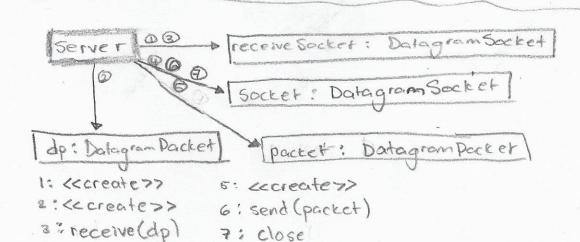
- 1: KCCCCate)
- 5 : «ccreate>> 6: send (dp)
  - 9: Kcreate>> 10: send (dp)

- 2: << create>> 3: «ccreate»>
- 7: «ccreate»
- 4: receive(dp)

4: «create >>

8: receive(dp)

### Server:



# JHL Class Diagrams:

# Client Hoocket; Datagram Socket + Client + main (String I Jargs) + test (Client c) + read (String s) + write (String s) + con (byte [] msg, String Str) + receive () + make Packet (int i, String s, String m) + make String (int i, String s, String m) + invalid Request (String s)

```
Intermediate

+ intermediate Port; int

+ serverPort; int

+ receive Socket; Datagram Socket

+ send Receive Socket; Datagram Socket

+ server Address; I net Address

+ Intermediate()

+ main (String [] args)

+ start()

+ run()

+ make String (byte [] data, int length)
```

```
Server

treceiveSocket; DatagramSocket

t port; int

t Server()

t main (String[] args)

t Start()

t receive()

t receive()

t make String(byte [] data, int length)

t readOrWrite(byte[] data, int length)
```