# Hacettepe University
# Computer Science and Engineering Department

| | |
|---|---|
| **Name and Surname** | **:** Ozan Özenoğlu |
| **Identity Number** | **:** 20724945 |
| **Course** | **:** Bil 342 |
| **Experiment** | **:** Experiment 2 |
| **Subject** | **:** Socket Programming in Java |
| **Data Due** | **:** 23.04.2011 |
| **Advisors** | **:** R.A. Ali Seydi Keçeli, Dr. Ahmet Burak Can |
| **e-mail** | **:** b20724945@cs.hacettepe.edu.tr |
| **Main Program** | **:** Server.class – Client.class |

## 2. Software Using Documentation
### 2.1. Software Usage

You must the start server firstly to provide information of peers to peers. To start server you should type *java server <server-port-number>* command into command line. You have to type a valid port number at local machine which is not bound any program. After this command you will see a GUI that belong server which shows peers action like connect or disconnect. After server startup , you can run client *with java client <server-ip-adress> <server-port-number> <user-name>* command. If you don’t want to type command at command line **just type** *java client* **to open interface that you can type ip , username and server port .** After connection some client you should choose a client from the list and click the start game button. After this step an agreement message sent other peer and if return of this message is possitive a new game will begin immediately . Game is a Checker game you can find the rules of game at http://www.jimloy.com/checkers/rules2.htm

## 2.2. Provided Possibilities

Provided Possibilities are **chatting at server channel , chatting at peer to peer and play checkers game** one to one .

## 2.3 Error Messages

There are some error message on client side. If connection between peers may corrupted a Corrupted connection message will show on command line. There are no GUI to show error message so if you want to handle error messages you should run peer on command line.

Let’s have a look these error message:

İf you corrupted you will get this **: Corrupted Connection** please try connect again.
İf any connection can’t opened between peers after click the “Play Button” you’ll get this Exception with default error message: **IOException**

# 3.Software Design Notes

## 3.1. Desctiption of the program

### 3.1.1. Problem

We need to implement peers which connect each other and a server to provide a service that supplier information of peers to each other.  Also we need to design a GUI for checker game with checker algorithm (this was  the hardest part of problem.)

### 3.1.2.Solution

SERVICE SERVER:

**Server Implemention:** Server need to handle a lot of clients and need to service these client at same time. This problem point directly MultiThreading library at java. So we need to implement Runnable interface or extends Thread class to provide multitasking. Firstly server create a ServerSocket and starts to listen this socket. After a connection come we get a socket from ServerSocket to give as a parameter to  new thread which handle operation between client and server. This Thread is a another class and has method to provide service clients.

In communication between client and server there are different type of message. But there are not too many as peers’s communication . So I don’t need to use a complicated protocol between client's and server. I don't use ObjectStream class . I just use PrintStream class to write char array and InputDataStream to read . There are three type of operation between client and server. One of them is AddPlayer. When server want to send information of a client to another client this operation is used. Server create a simple message which start with "AddPlayer" and send client with the client information that will send. And client get a message from server check the starting word and decide what to do. I use this method for simplicity but peer communication is differ from this due to type of communacition between them. When server want to say delete a client from list to client it sends a message start with "DeletePlayer" with user name. When client connected to server it send it's port and userName to registration player list . There is no message except this client to server except this , when a client decide to close connection between server and itself send a message to server "/exit". Thats all of protocol between server and client.

**PEER SIDE:**

**Client Implemention:**

If you read server ımpelemention part you should know how client connects to server and register itself. And how list other clients. In this part we try to explain how clients conenct each other. Firstly you should select a player from player List. After selection you can click start button to start game. After a acceptance from other peer a new game should begin. Well , what happens in background for this operation. When a client connects to server it sends information of itself to registration . And starts to recieve messages which probably belong AddPlayer operation if there are another clients which are online. When recive AddPlayer operation message with information client parse first message and get information of client. Then create a Player Object from Player Class and then add this instance into PlayerList which type of is Player. Player Class statictly provide us some getSocket,getPlayerWithName and etc. Named method . When client needs an information of other client use getPlayerWithName to get player instance and then getSocket method to get live Socket between player and itself. After got a socket with other peer client create a **Game Object** which instance of Game Class with this socket . But in the other side peer has a server side to provide communication with other clients. If there is no a separete server side peer can't handle connection request.

**Server Implemention:**

When we start client actually we start two thread . One of them is client and the other one is server. Server creates a ServerSocket with port that will given as parameter to Peer Program at the start of running the Peer. Then starts to listen this socket . When a connection request comes accept this and gets a socket from this connection. After got a socket server create a **Game** object from **Game** Class with this socket .

**Game Implemention:**

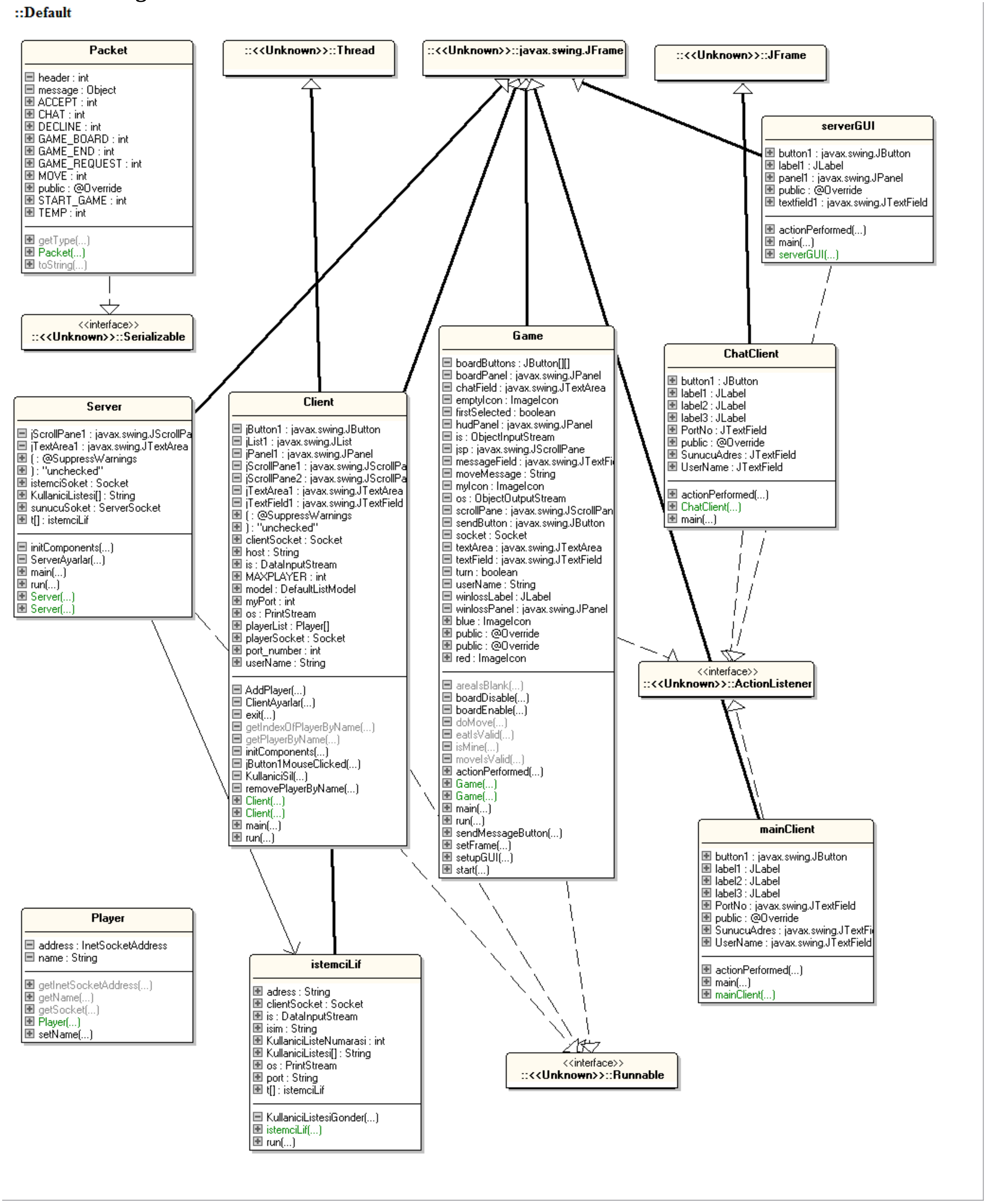
This Class has all method which provide make playable Checker game. This Class has algorithm of Checker game and has method provide communication with peer to peer. Near to all message send via this thread. This class use a Protocol Class which implements serilizeable and has some types of message to create serilizeable packet which will be send. We need to serilizeable instance of object to send on sockets with WriteObject method. Let's have a look this Class and its protocol design.

**Protocol :**

We have a few type message . Most used of this type is **MoveSent** . MoveSent using by Game Class when it needs to send a move operation to other player. We have static final int variable in Packet class to set type of packets. When we need send a message other client we just simply create a instance of this class with message and a type and send this object using ObjectWriter to other peer. The Second most using type of message is Chat. As I told before players can **Chat** between each other. All chat message packet with this type of packets.

Other type of packer is Start_Game, Game_End, Game_Request,Accept,Decline and Temp. This types explain what they use for with their name .

## 3.3. Class Diagram



### 3.5. Special Design Properties

We use object oriented methodology like encapsulation , interface implemention and etc.

### *3.6.* Execution Flow Between Subprograms

## 4. SOFTWARE TESTING NOTES :

We had test software many times and found some bugs in software , lets have a look what we found.

### 4.1. Bugs and Software Reliability

Like mentioned above there are a few bugs in this game .

**Most Important BUG**<u>: Some time clients can’t recieve other clients information and can’t show user list. When you sure other players connected to server but your new client doesn’t show them to you , re –run and re – connect your clinet please.!!</u>

**#bug1:** When you click a button on game board and then decide to move another gem and click it you have to click twice on the button .

**#bug2 :** When a user exit the game the other user client can’t handle it and try to countine so it going to crash.

**#bug3:** When a user close the game the main client also exit.

**Other Bugs:** There are some other bugs which handled by program so users can’t understand ☺ But if user want to show error mesage can look console what happens in background.

### 4.2. Software Extendibility and Upgradability

This software server is useable for other operations like chat or p2p client communacition. But client is only for chat and play checker game.

### 4.3. Performance Considerations

*Most cpu usage belong to I/O and GUI creating. But generally this program light fit and nippy.*

### 4.4. Comments

*Very good problem . While we make solution to this problem a lot of things ı have learned and I had enjoyable time.*

## REFERENCES

http://java.sun.com/docs/books/tutorial/networking/index.html
http://java.sun.com/j2se/1.4.2/docs/guide/imageio/spec/title.fm.html

**Main Reference : Alfa yayınları Herbert Schildt Java SE 6**