Öğrencinin;

ADI: FATİH

SOYADI: ÖZKAN

NO: 1621221014

BÖLÜM: BİLGİSAYAR MÜHENDİSLİĞİ

Projenin;

KONUSU: Kızma birader: 2 kişilik bir kızma birader oyunu server üzerinden oynanacaktır.

Dersin;

ADI: BİLGİSAYAR AĞLARI

EĞİTMEN: ALİ YILMAZ ÇAMURCU

İçindekiler

1- Özet

TCP/IP Nedir? Nasıl Çalışır?

TCP/IP **TCP** (Transmission Control Protocol) ve **IP** (Internet Protocol) protokollerinin birleştirilmesiyle oluşturulan internet üzerindeki bir iletişim metodudur.

Bu metot sayesinde internete bağlanan tüm cihazlar birbirleri ile haberleşebilir. Bir ağa bağlanan bilgisayarlar veri iletmek ve almak için birbirleri arasında **TCP/IP protokolü** ile haberleşmektedir. **Kısacası TCP/IP protokolü, bilgisayarlar arası veri iletişiminin kurallarını koyan bir iletişim protokolleri bütünüdür.** Bilgisayarlar arası iletişim farklı protokol aileleri veya tipleri üzerinden gerçekleştirilir. Kullanım amacına göre ise bu protokoller birbirlerinden ayrılmaktadır.

Örnek olarak **FTP** (File Transfer Protocol), dosya iletim protokolüdür. **SMTP** (Simple Mail Transfer Protocol) ise e-posta iletmek için kullanılırken, **SSH** protokolü Linux sistemler üzerinde güvenli bir tünel oluşturularak yönetim sağlama amacıyla kullanılmaktadır.

TCP Nedir?

TCP (Transmission Control Protocol) bilgisayarlar arasındaki iletişimin, küçük paketler hâlinde ve kayıpsız olarak gerçekleştirilmesini sağlayan bir protokoldür. Aslında TCP (Transmission Control Protocol) protokolünün en önemli özelliği kimlik doğrulaması yapması ve veriyi karşı tarafa gönderirken veya alırken verinin bütünlüğünü sağlamasıdır. Gelişmiş bilgisayar ağlarında ortaya çıkan kayıpları önlemek için TCP protokolü yazılmıştır. HTTP, HTTPS, POP3, SSH, SMTP, TELNET ve FTP gibi günlük hayatta sıkça kullandığımız protokollerin veri iletimi TCP vasıtasıyla yapılır.

UDP protokolüne göre yavaş ancak güvenli bir veri iletişimi sağlar. UDP (User Datagram Protocol) protokolünde ise verinin karşı bilgisayar tarafından alınıp alınmadığı kontrol edilmez ve veri iletişimi çok hızlı bir şekilde gönderilir.

TCP (Transmission Control Protocol) Nasıl Çalışır?

TCP protokolünün çalışma mantığı üç başlıkta incelenebilir. Birinci aşamada hedefe bir bağlantı isteği gönderilir. İkinci aşamada bağlantının gerçekleştiği onaylanır ve veri transferi başlar. Üçüncü aşamada ise veri transferinin tamamlandığı taraflara iletilerek bağlantı sonlandırılır. Bu üç aşamanın gerçekleşmesi ‘’State’’ işlemi olarak tanımlanır.

IP Nedir?

IP, Türkçe açılımı ve çevirisiyle internet protokolünün kısaltmasıdır. Bilgisayarların birbirleri ile iletişiminde en önemli nokta olan ağ adreslemede kullanılan düzendir. IP (internet protokolü), iki bilgisayar arasındaki paketlerin yönlendirilmesini sağlayan bir protokol olarak tanımlanır. Yönlendirme protokolü olarak tanımlanan IP, veri için gerekli olan yönlendirmenin kurallarını

belirlemektedir. IP'de verinin niteliği önemli değildir ve veri içeriği ile ilgilenmeden bu protokol sadece gideceği adresi belirler.

IP (Internet Protocol) Nasıl Çalışır?

IP çalışma mantığında TCP veri paketinin **izleyeceği yol** belirlenir. Bu işlem yapılırken de TCP katmanından gelen veri paketine kendi **SEGMENT** başlığını ekler. Bu durumda ortaya datagram çıkmış olur. IP (Internet Protocol), veriyi karşı tarafa yönlendirirken, alıcının bu veriyi kabul edeceği veya etmeyeceği konusunda bir doğrulama yapmaz. Hata kontrolü bir üst katmanın işidir ve bu sayede IP kendi başına çalışabilen bir protokol olarak bilinir.

TCP/IP Referans Modeli

4. Katman – Uygulama

3. Katman – Taşıma

2. Katman – İnternet

1. Katman – Ağ

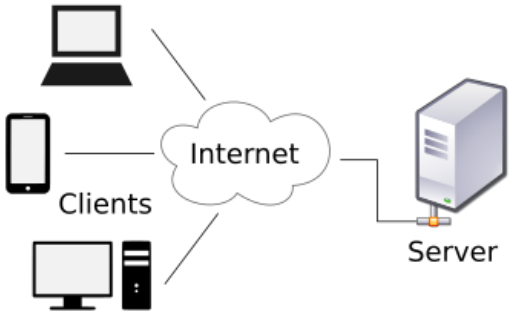
Uygulama Katmanı: Uygulama katmanında farklı sunucular üzerindeki süreç ve uygulamalar arasındaki iletişim sağlanır. Bu iletişim sırasında ise HTTP, HTTPS, SSH, SMTP, TELNET ve FTP gibi protokoller kullanılmaktadır.

Taşıma Katmanı: Bir noktadan diğer noktaya veri akışını sağlayan katmandır. Bu katman üzerinde TCP ve UDP protokolleri kullanılmaktadır.

İnternet Katmanı: Router cihazları ile birbirlerine bağlanmış olan ağlar arasında verinin bir kaynaktan hedef bilgisayara kadar gerekli olan iletiminin sağlanması için kullanılır. Bu veri aktarımı sırasında kullanılan bilgiler internet katmanı ile transfer edilir.

Ağ Erişim Katmanı: Bu katmanda uç nokta ile ağ arasında yer alan bağlantı arabirimi kullanılır. Bilgisayar dili 0 ve 1'lerden oluşmaktadır ve bu katmandaki iletişim için veri paketleri 0 ve 1'lere dönüştürülerek taşınır.

Server/Client ?



İstemci (Client) : Bir ağ üzerinde, sunucu bilgisayarlardan hizmet alan kullanıcı bilgisayarlarıdır. Bilgiye erişim yetkileri sunucu tarafından belirlenir. Eğer bir bilgisayardan Internete bağlanılarak web siteleri ziyaret ediliyorsa o bilgisayar İstemci bilgisayardır.

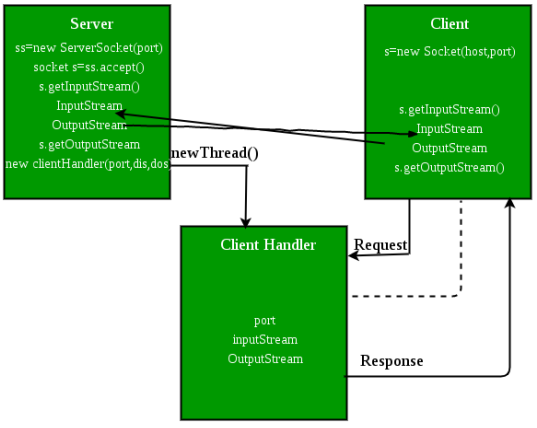
Sunucu (server) : Sunucular bir ağ üzerinde bilgileri veya uygulamaları kullanıcılara paylaştıran, donanım ve yazılım bileşenlerinden oluşan özel bilgisayarlardır.

Sunucular sundukları hizmete göre isimlendirilirler. Örneğin sunucu bilgisayar bir veritabanı hizmeti sağlıyorsa "veritabanı sunucusu" bir "web server" olarak hizmet veriyorsa "Web sunucusu" şeklinde isimlendirilebilirler.

Ağ programlamada neden "Thread" kullanılmalı?

Nedeni basit, belirli bir zamanda sunucuya yalnızca **tek bir istemcinin bağlanmasını değil, aynı anda birçok istemcinin bağlanmasını istiyoruz.** Mimarimizin aynı anda birden fazla istemciyi desteklemesini istiyoruz . Bu nedenle, bir istemci isteği geldiğinde, her isteğin işlenmesi için ayrı bir iş parçacığı atanabilmesi için sunucu tarafında iş parçacıkları kullanmalıyız.

Normal olarak, Server.java ve Client.java olmak üzere iki java dosyası oluşturacağız . Sunucu dosyası, Sunucu (sunucu oluşturmak için genel sınıf) ve ClientHandler (çoklu iş parçacığı kullanarak herhangi bir istemciyi işlemek için) olmak üzere iki sınıf içerir. İstemci dosyası yalnızca bir ortak sınıf İstemci içerir (bir istemci oluşturmak için). Aşağıda, bu üç sınıfın birbirleriyle nasıl etkileşime girdiğine dair akış diyagramı yer almaktadır.



Bağlantının Kurulması: Sunucu soket nesnesi başlatılır ve bir while döngüsü içinde bir soket nesnesi sürekli olarak gelen bağlantıyı kabul eder.

Akışların Alınması: Giriş akışı nesnesi ve çıkış akışı nesnesi, mevcut isteklerin soket nesnesinden çıkarılır.

İşleyici nesnesi oluşturma: Akışları ve bağlantı noktası numarasını aldıktan sonra, bu parametrelerle yeni bir clientHandler nesnesi (yukarıdaki sınıf) oluşturulur.

start() yönteminin çağrılması : **start()** yöntemi, bu yeni oluşturulan iş parçacığı nesnesinde çağrılır.

ClientHandler sınıfı : Her istek için ayrı threadler kullanacağımızdan, ClientHandler sınıfını genişleten Threads'in çalışmasını ve uygulamasını anlayalım. Her istek geldiğinde bu sınıfın bir nesnesi başlatılacaktır.

Her şeyden önce, bu sınıf **Thread**'i genişletir , böylece nesneleri, Threads'in tüm özelliklerini üstlenir.

İkinci olarak, bu sınıfın yapıcısı, gelen herhangi bir isteği benzersiz bir şekilde tanımlayabilen üç parametre alır, yani bir **Socket** , okunacak bir DataInputStream ve yazılacak bir DataOutputStream . İstemciden herhangi bir istek aldığımızda, sunucu bağlantı noktası numarasını, DataInputStream nesnesini ve DataOutputStream nesnesini çıkarır ve bu sınıftan yeni bir iş parçacığı nesnesi oluşturur ve bunun üzerinde **start()** yöntemini çağırır.

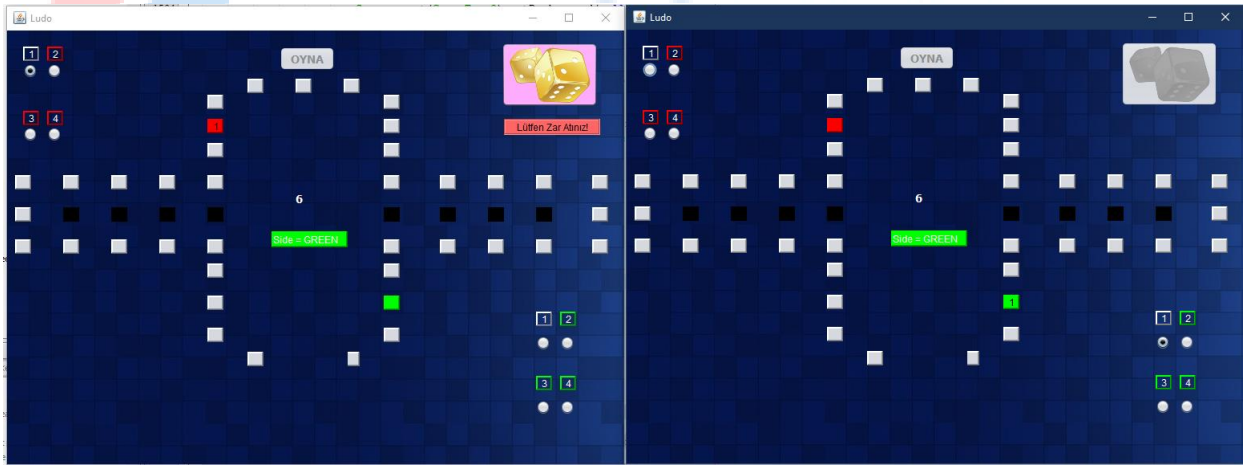
2- Proje Konusu

Program: Kızma birader: 2 kişilik bir kızma birader oyunu server üzerinden oynanacaktır.

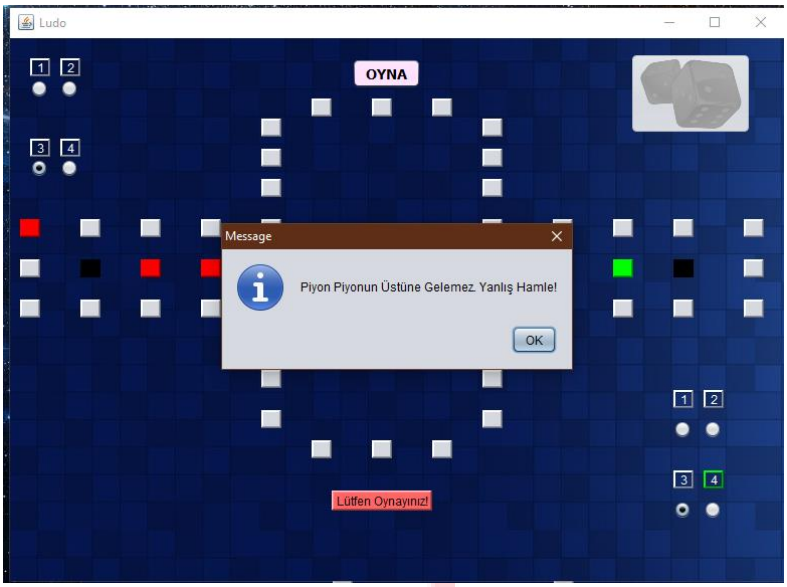
3- Proje Çıktıları ve Başarı Ölçütleri

Proje başarı ile tamamlanmıştır. Server aktif olduğu halde 2 adet Client uygulaması Server'a bağlanıp karşılıklı sıra ile Kızma Birader oyununu oynayabilmektedirler.

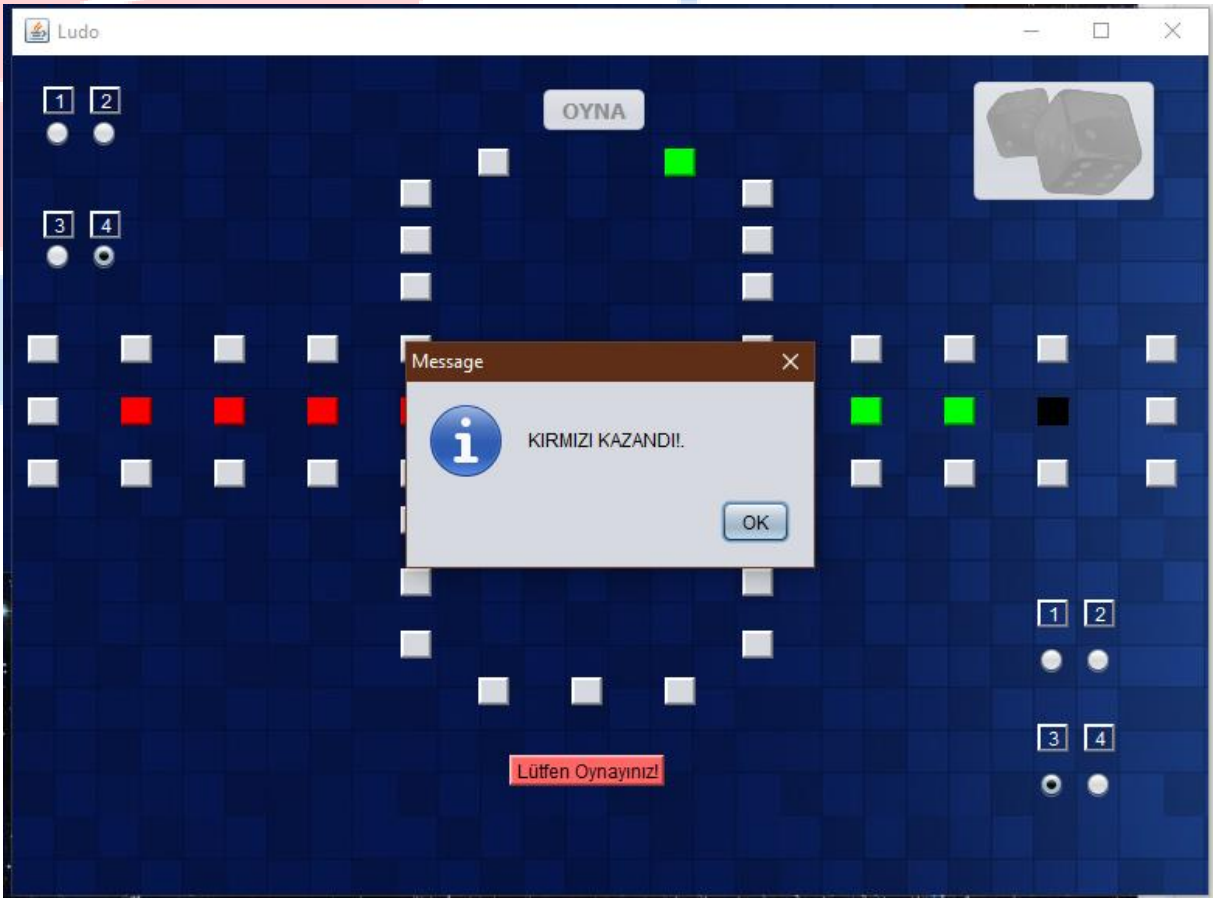
Yukarıdaki ekran görüntüsü Java Swing ile tasarlanmış Kızma Birader oyununa aittir. İki Client da aynı ekran tasarımına sahiptir. Sıraya göre “OYNA”,”ZAR” ve uyarı labelleri aktif ve inaktif olmaktadır.



- Yukarıda ki ekran görüntüsünde Server’a başarılı bir şekilde bağlanmış 2 adet Client vardır.
- Bu ekranlardan soldaki ‘YEŞİL’ tarafa sağdaki ise ‘KIRMIZI’ tarafa aittir.
- Sıra ile zar atılır ve gelen zara göre oyun oynanır.
- Piyon çıkmak için ‘6’ zarı atılması şarttır.
- ‘6’ atan taraf piyonunu çıkarak oyuna başlar ve gelen zarlar miktarınca ilerleme kaydeder.



- ‘6’ üst üste piyon gelemez. Eğer üst üste piyon gelecek şekilde oynanırsa oyun uyarı verir ve sıra karşı tarafa geçer.
- Her iki tarafa ait 4’er adet piyon vardır.
- Her iki takım için de ortada 4’er adet piyon yuvası ayrılmıştır.



- Bu 4 ilk yuvayı dolduran taraf oyunu kazanır ve oyun biter.

4- Proje Süresince Yapılanlar

Derste işlenilen Server, message, sınıfları yazıldı.

Client, Sclient sınıfları yazıldı.

Oyun kuralları araştırıldı ve internet üzerindeki çeşitli versiyonları incelendi.

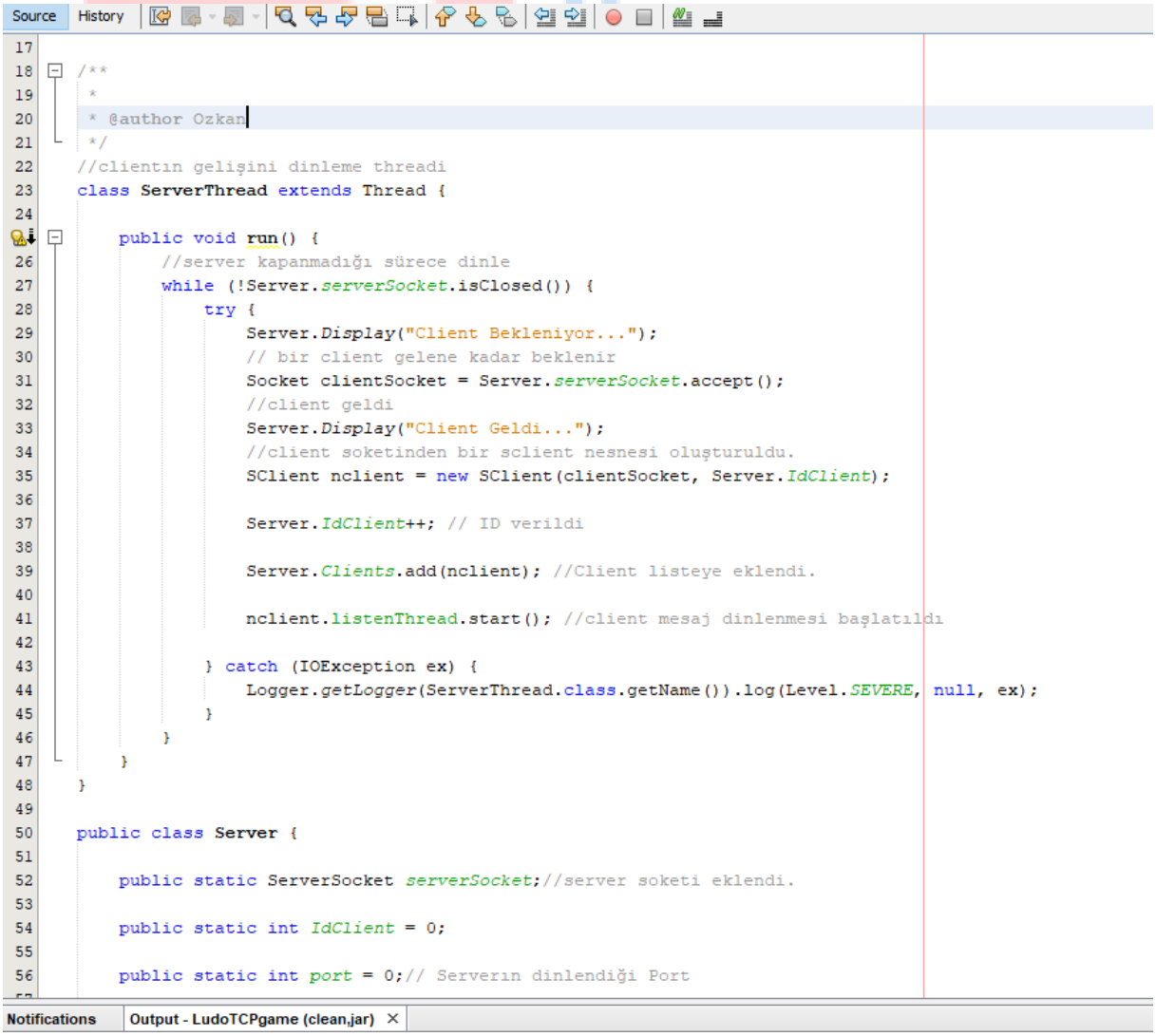
Gerekli parametreler oluşturuldu.

Kurallar oluşturuldu.

Oyun döngüsü kuruldu.

Client - Server uygulaması gerçekleştirildi.

Projeye ait kodların ilgili commentleri kod satırları içerisindedir. Örnek:



```
Source  History

17
18  /**
19   *
20   * @author Ozkan
21   */
22  //clientın gelişini dinleme threadi
23  class ServerThread extends Thread {
24
26      public void run() {
27          //server kapanmadığı sürece dinle
28          while (!Server.serverSocket.isClosed()) {
29              try {
30                  Server.Display("Client Bekleniyor...");
31                  // bir client gelene kadar beklenir
32                  Socket clientSocket = Server.serverSocket.accept();
33                  //client geldi
34                  Server.Display("Client Geldi...");
35                  //client soketinden bir sclient nesnesi oluşturuldu.
36                  SClient nclient = new SClient(clientSocket, Server.IdClient);
37
38                  Server.IdClient++; // ID verildi
39
40                  Server.Clients.add(nclient); //Client listeye eklendi.
41
42                  nclient.listenThread.start(); //client mesaj dinlenmesi başlatıldı
43
44              } catch (IOException ex) {
45                  Logger.getLogger(ServerThread.class.getName()).log(Level.SEVERE, null, ex);
46              }
47          }
48      }
49
50  public class Server {
51
52      public static ServerSocket serverSocket;//server soketi eklendi.
53
54      public static int IdClient = 0;
55
56      public static int port = 0;// Serverın dinlendiği Port

Notifications  Output - LudoTCPgame (clean,jar)
```

5- Ek Açıklamalar

6- Kaynakça

https://github.com/sametkaya

https://www.geeksforgeeks.org/introducing-threads-socket-programming-java/

https://sefik.net/?id=143&mid=83#.Ym1pPdpByUk

https://berqnet.com/blog/tcp-ip