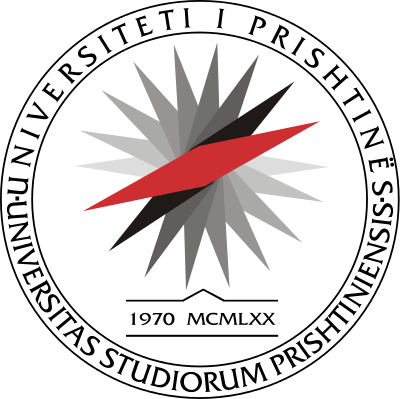
**Universiteti i Prishtinës**

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Programi: Inxhinieri Kompjuterike



**Projekti 1 : Dizajnimi Klient-Server**

**Raporti**

**Lënda: Rrjeta Kompjuterike**

Studenti: Blerta Jashari Nr. Indeksit: 160714100155

Profesori i lëndës: Blerim Rexha Assistenti i lëndes: Haxhi Lajqi

Data: 04.18.2019

Vegla e përdorur për zhvillimin e software-it: Visual Studio 2017 Community Edition

Gjuha programuse e përdorur: Python 3.6

Sistemi operativ i përdorur: Windows 10 Pro (64 bit)

# **Hyrje**

Ky raport përshkruan në detaje punën e bërë gjatë zhvillimit të Protokollit FIEK, detyrë e dhënë në kuadër të lëndës Rrjeta Kompjuterike si Projekti 1.

Projekti I dhënë është realizuar në atë mënyre që të egzistojë një lidhje mes Serverit dhe Klientit, ku këto dy programe mes vete ta kenë të mundur shkëmbimin e kërkesave,dhe kthimin e përgjigjeve duke komunikuar nërpërmjet socket-ave.

Server - përdoret për pranimin e kërkesës nga ana e klientit dhe kryen operacionin në bazë të kërkesës dhe jep rezultat si përgjigje në anën e klientit.

Klienti - përdoret për dërgimin e kërkesës në server pret për përgjigjen e saj në anën e klientit.

Projekti përmban 4 programe të ndryshme: FIEK-TCP klienti, FIEK-TCP serveri, FIEK-UDP klienti dhe FIEK-UDP server.

TCP protokolli besueshëm që garanton dorëzimin e të dhënave dhe gjithashtu garanton që të dhënat do të dorëzohen në të njëjtin mënyrë në të cilën janë dërguar pra tuhet rendi I i dërgesës.

UDP protokolli paraqet një protokoll jo të besueshëm, pra nuk ka garancion se të dhënat do të mbërrijnë në cak gjithmonë dhe që do të ruhet rendi i dërgesës ashtu siç e kemi përcaktuar.

Permbajtja

[**Hyrje** 2](#_Toc6486506)

[ **Përshkrimi i Metodave** 4](#_Toc6486507)

[ **Metoda IPADDR** 4](#_Toc6486508)

[ **Metoda PORTNR** 5](#_Toc6486509)

[ **Metoda BASHKETINGELLORE** 5](#_Toc6486510)

[ **Metoda PRINTO** 6](#_Toc6486511)

[ **Metoda HOST** 6](#_Toc6486512)

[ **Metoda KOHA** 7](#_Toc6486513)

[ **Metoda LOJA** 7](#_Toc6486514)

[ **Metoda FIBONACCI** 8](#_Toc6486515)

[ **Metoda KONVERTO** 8](#_Toc6486516)

[ **Metoda GUESSGAME** 9](#_Toc6486517)

[ **Metoda HOROSKOPIKINEZ** 9](#_Toc6486518)

[ **Dallimi nga programimi sipas TCP dhe UDP protokolli** 10](#_Toc6486519)

[ **Permbledhje rreth rezultateve** 11](#_Toc6486520)

# **Përshkrimi i Metodave**

Te gjitha metoda thirren duke shtypur emrin perkatese ne kete rast klienti dergon kerkesen ne server dhe nese nuk ndodh ndonje gabim serveri kthen pergjigje ne klient. Te gjitha metoda thirren pasi te jete realizura lidhja Client-Server.

Metodat e implementuara në programet TCP dhe UDP janë:

**1.TIME**

**2.HOST**

**3.IP**

**4.PORT**

**5.ZANORE**

**6.FAKTORIEL**

**7.KONVERTO**

**8.PRINTO**

**9.KENO**

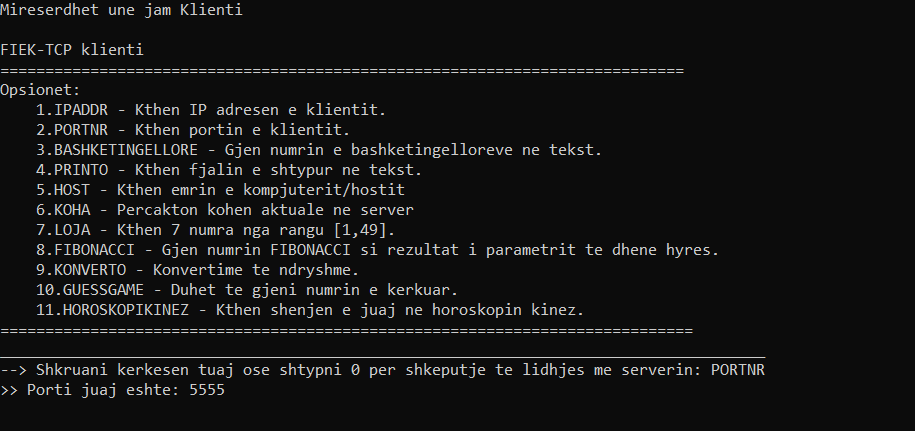
**10.GUESSGAME**

**11.HOROSKOPIKINEZ**

## **Metoda IPADDR**

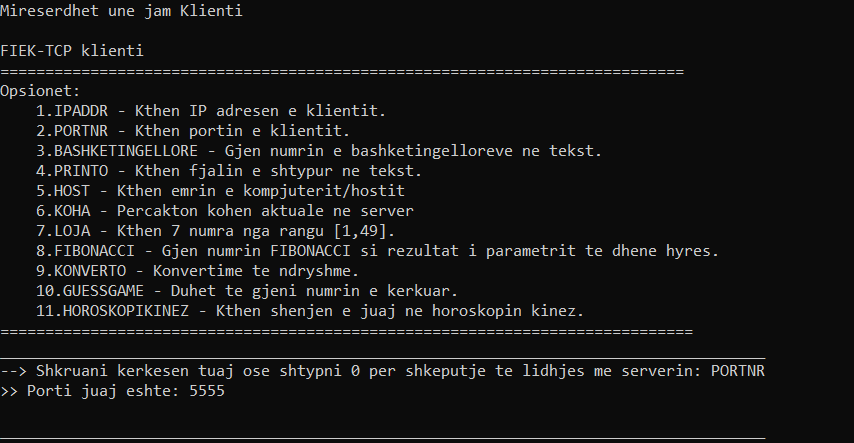
Metoda IPADDR kthen IP adresën e klientit në formë dhjetore me pikë në kuadër të një tekst mesazhi.

Metoda portnr



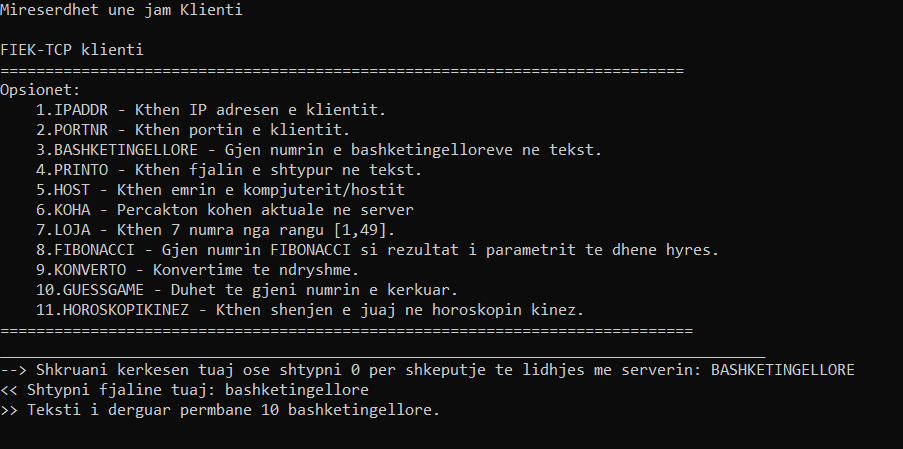
## **Metoda PORTNR**

Përcakton dhe kthen portin e klientit.



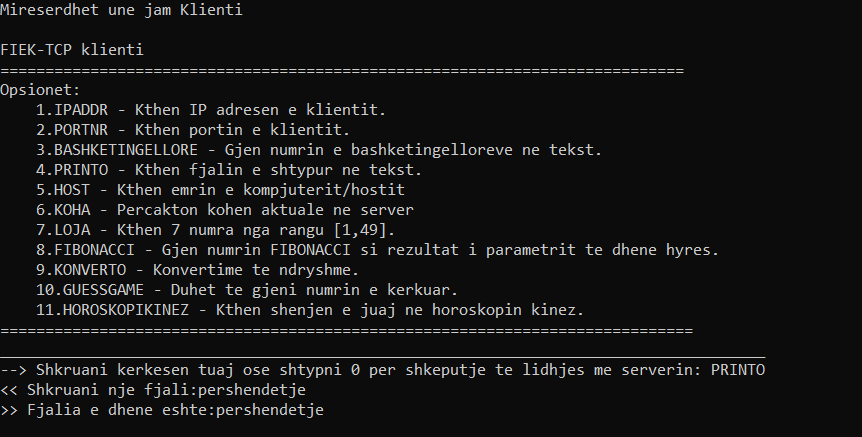
## **Metoda BASHKETINGELLORE**

Metoda Zanore e gjen numrin e zanoreve në tekstin e dhënë e cila dergohet nga klienti ne server.Serveri me atë rast nga ai tekst në bazë të një numratori të cilin e ka ne funskionin e tij, shikon me radhë cdo shkronjë të atij teksti dhe nëse has në bashketingellore numratori rritet per 1 dhë kështu në kthen si rezultat numrin e bashketingelloreve ne atë tekst.



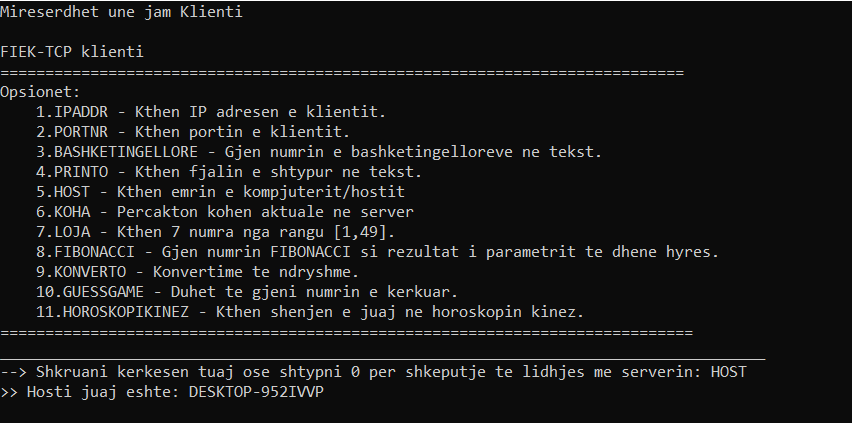
## **Metoda PRINTO**

Metoda Printo është metode e cila kthen cfarëdo mesazhi që klienti e shkruan,Serveri po atë mesazh e printon, dhe ia dergon prap Clientit për të parë mesazhin që ka shtypur.



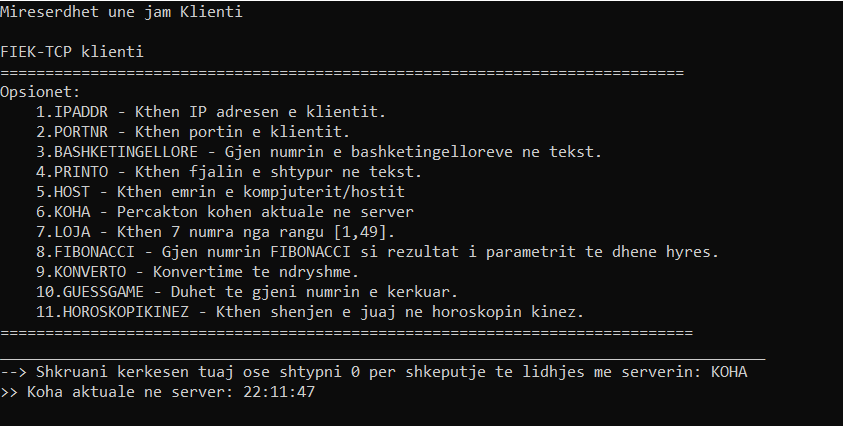
## **Metoda HOST**

Metoda HOST mundeson kthimin e emrin e Hostit,dhe gjithashtu me një metode try: nese emri I hostit nuk gjindet Serveri të na lajmeroj për këtë gjë.



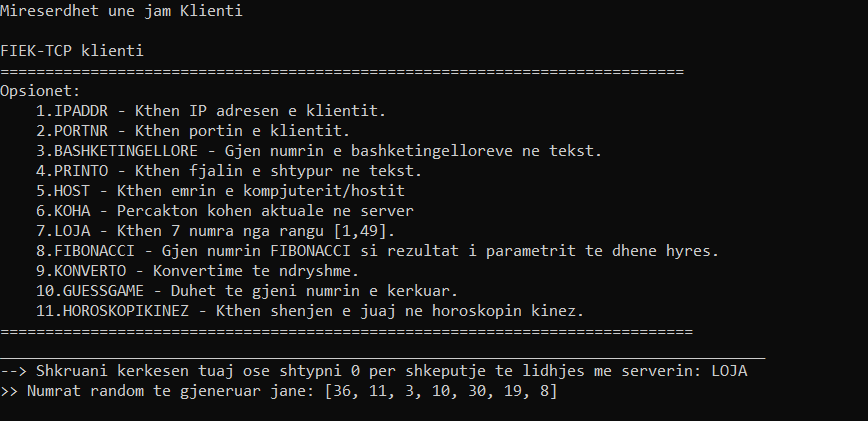
## **Metoda KOHA**

Metoda KOHA kthen kohën aktuale në server, si format të lexueshëm për përdoruesit. Kjo është përfituar nga libraria e importuar nga python datetime, ku është marrë koha aktuale në system përmes metodës *datetime.now( )*.



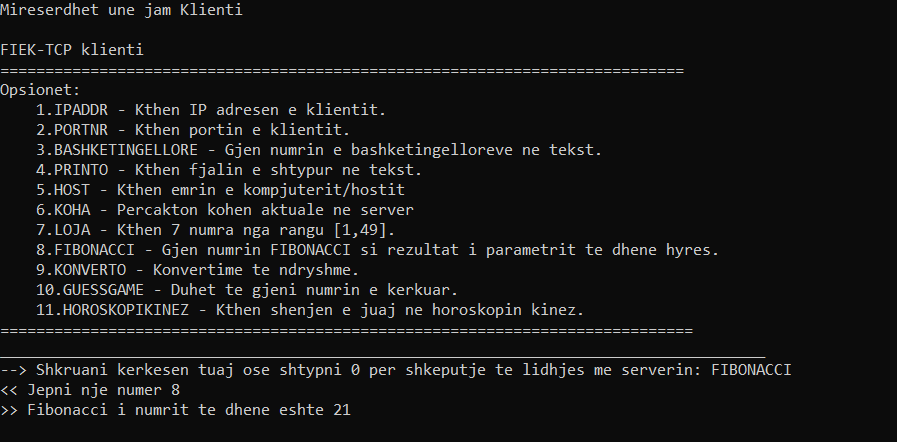
## **Metoda LOJA**

Metoda LOJA kthen një varg të 7 numrave random të rangut 1 deri 49.

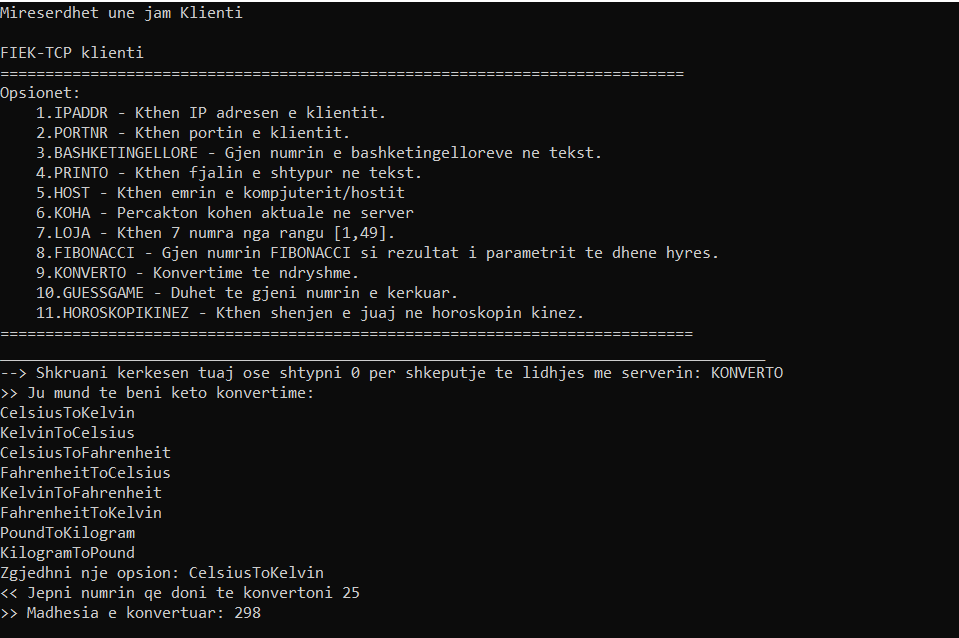


## **Metoda FIBONACCI**

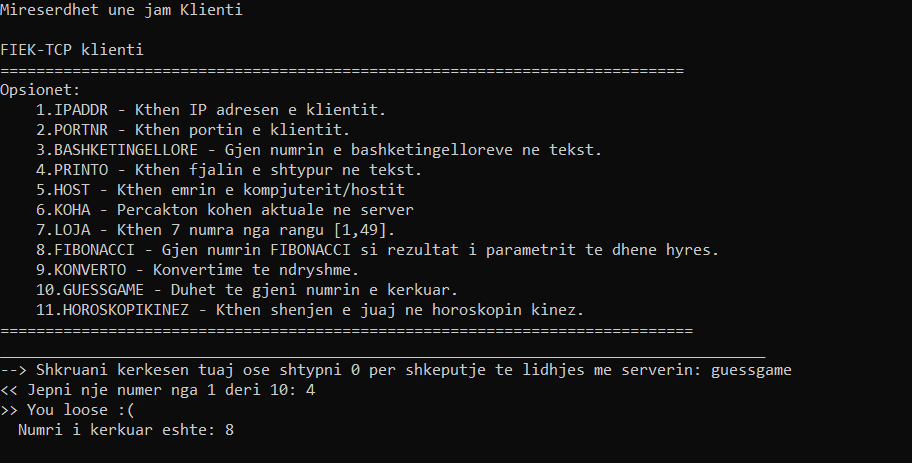
Metoda FIBONACCI gjen numrin Fibonacci që paraqitet si rezultat i parametrit të dhënë hyrës.



## **Metoda KONVERTO**

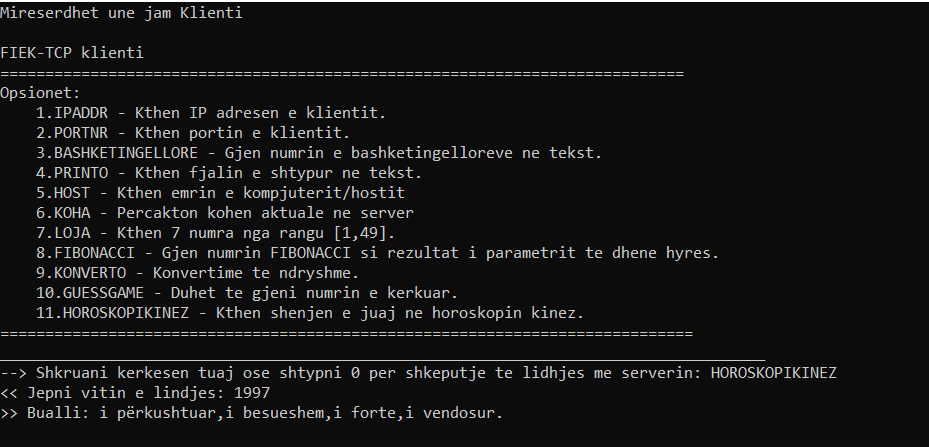
Metoda KONVERTO kthen si rezultat konvertimin e parametrit të dhënë varësisht prej opcionit të zgjedhur.Pas zgjedhjes se kesaj metode duhet te shtypet lloji I konvertimit te cilin deshirojm te bejem dhe vleren.Pra metoda merr dy parametra nga klienti, së pari tipin e konvertimit dhe pastaj vlerën që deshiron të konvertoj

## **Metoda GUESSGAME**

Metoda GUESSGAME është një lojë ku duhet të gjejm numrin e kerkuar.Keto klienti I dergon serverit një numer nga 1 deri 10 serverin e krahason numrin me numrin e tij secret (secret\_number) e cila gjenerohet random (1-10) dhe I krahason te dy vlerat dhe varësisht a janë te barabarta a jo kthen përgjigjen përkatëse. 

## **Metoda HOROSKOPIKINEZ**

Metoda HOROSKOPIKINEZ kthen horoskopin kinez varesisht nga viti i lindjes qe dergohet nga klienti ne server.Keto vitet jane te ruajtur ne array(vargje) ku viti I dhene nga klienti permes if elif kalon neper vargje deri sa te gjendet vlera dhen dhe kthen pergjigjen perkatese.



# **Dallimi nga programimi sipas TCP dhe UDP protokolli**

Dallimi nga programimi sipas TCP protokollit qëndron se UDP protokolli ëhtë *connection-less,* pra nuk vendoset një lidhje ndërmjet klientit dhe serverit që ata të komunikojnë mes vete.Parametri i dytë i socketit të TCP protokolli do të jetë [SOCK\_STREAM](https://stackoverflow.com/questions/35725732/what-is-the-function-of-sock-stream) dhe kur parametri i dytë i socketit do të jetë SOCK\_DGRAM, që do të thotë se kemi të bëjmë me UDP socket.

TCP: s=socket(AF\_INET,SOCK\_STREAM) UDP: s=socket(AF\_INET,SOCK\_DGRAM)

Te TCP kemi nevojë të vendosim një lidhje në mes klientit dhe serverit duke perdorur metoda connect( ) ndersa te UDP nuk kemi nevojë.

Dallimi tjetër qëndron në metodën e dërgimit të të dhënave. Të TCP përdoret metoda send() ndersa UDP përdoret metoda sendto( ), ku si parametra iu bashkangjitet emri i hostit dhe numri i portit të serverit.

TCP:s.sendall(str.encode(mesazhi)) UDP:s.sendto(str(mesazhi).encode(),(hosti,porti))

Edhe metoda e pranimit te te dhenave dallon TCP përdoret metoda recv() të UDP recvfrom(). Gjthashtu kemi edhe një dallim tjeter te TCP duhet te mbyllet lidhja e serverit me klientin pas nje kërkese që bëhet me anë të metodes s.close(),ndersa te UDP nuk kem nevoj te mbyllin lidhjen.

# **Permbledhje rreth rezultateve**

Klienti ka pasur mundësi të dërgoj kërkesa të cilat përmbajnë emrin e metodave të përcaktuara dhe serveri në bazë të asaj kërkese nëpërmjet disa kushtëzimeve ka kthyer përgjigjen te klienti, ku përgjigja ka përfshirë zgjidhjen e asaj metode. Nga testimi me larte mund të vërehet se secila metodë e dhënë funskionon, dhe secila prej tyre kthen rezultatin e kerkuar.