­­­­ UNIVERSITETI I PRISHTINËS

FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE

**RRJETA KOMPJUTERIKE**

PROJEKTI 1:  
**Dizajnimi Klient-Server**

­­­­­­

Mentori: Punoi:

*MSc. Haxhi Lajqi Vigan Dika*

Prishtinë,

Prill, 2019

­­

Table of Contents

[1. Veglat e perdorura dhe metodat e implementuara 3](#_Toc6519468)

[2. Hyrje e shkurtwr pwr projektin 3](#_Toc6519469)

[2.1 FIEK – TCP (Transmission Control Protocol) 3](#_Toc6519470)

[2.2 FIEK – UDP (User Datagram Protocol) 3](#_Toc6519471)

[3. FIEK - TCP 4](#_Toc6519472)

[3.1 TCP Serveri - Pwrshkrimi i kodit 4](#_Toc6519473)

[3.2 TCP Klienit – Pwrshrkimi i kodit 5](#_Toc6519474)

[3.3 Trajtimi i gabimeve 5](#_Toc6519475)

# Veglat e perdorura dhe metodat e implementuara

Projekti wshtw implementuar ne gjuhen programuese Python, versioni 3.7 kurse si IDE wshtw pwrdorur JetBrains PyCharm versioni 2019.1.

Libraritw e pwrdorura pwr zhvillimin e projektit janw: *socket, sys, datetime, random, re, math, \_thread*.

Lista e metodave tw implementuara: *IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, ROCKPAPERSCISSORS, PRIM, KONVERTMI*.

# Hyrje e shkurtwr pwr projektin

Projekti ka si fokus komunikimin klient-server i cili wshtw i mundur me anw tw programimit me soketa (ang. sockets). Programimi me socketa wshtw njw mwnyrw e lidhjes sw dy pikave (soketave) qw komunikojnw me njwra tjetrwn. Njeri socket wshtw socketi dwgjues (listener socket), qw shwrben si pikw referuese pwr socketin tjetwr i cili tenton tw lidhet me tw. Serveri wshtw socketi dwgjues qw pret kwrkesa dhe e servon klientin me tw dhwna, kurse klienti wshtw socketi qw lidhet me server dhe merr tw dhwnat qw i kwrkon.

Serveri dwgjon nw njw adresw tw caktuar, njw kombinim unik (IP adresw, port), adresw kjo qw targetohet nga klienti qw kwrkon tw lidhet dhe ta shfrytwzojw atw.

Pasi vendoset lidhja e tyre, klienti do ta ketw tw mundur tw thwrrasw metodat e ofruara nga serveri dhe ta shoh pwrgjigjen tw cilwn e proceson serveri, ndwrsa serveri pwrpos qw ofron pwrgjigje pwr kwrkesat e klientit, wshtw nw gjendje tw evidentojw çdo komunikim tw ndwrsjelltw me klientin.

Ky lloj komunikimi i shfaqur nw projekt, nuk wshtw protokoll standard por wshtw njw njw protokoll i veçantw, tw cilin do ta quajmw protokolli FIEK. Ky protokol ka versionin e tij TCP qw quhet FIEK-TCP, dhe UDP versionin, FIEK-UDP.

## 2.1 FIEK – TCP (Transmission Control Protocol)

Wshtw protokoll i besueshwm. Komunikimi TCP nuk wshtw komunikim njw kahorw, andaj bazohet nw komunikimin e vazhdueshwm klient-server. Ky protokoll detekton paketat qw kanw dwshtuar tw transmetohen (dropped packages) dhe provon t’i ritransmetojw ato, njwkohwsisht duke dhwnw edhe arsyen e dwshtimit tw transmetitmit. Ky protokoll dwrgon tw dhwnat nw rendin e njejtw siç i ka dwrguar dwrguesi. Serveri ka mundwsi tw komunikojw me mw shumw se njw klient duke pwrdorur ‘multi-threading’, teknikw kjo qw lejon mw shumw klient tw kenw qasje nw kodin e njejtw.

## 2.2 FIEK – UDP (User Datagram Protocol)

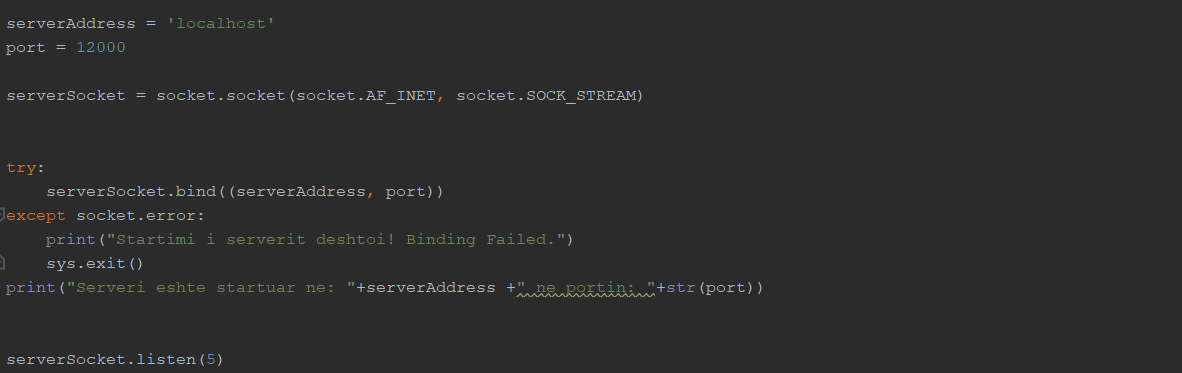
Vendosja e paketave nw njw sekuencw, vwrtetimi i dwrgesave dhe kwrkesa e ritransmetimit tw paketave qw kanw dwshtuar humbin shumw kohw dhe ngadalwson komunikimin, andaj protokolli UDP i anashkalon tw gjithw kwto procese. Gjatw komunikimit UDP, dwrguesi wshtw i interesuar qw tw dwrgojw sa mw shumw informata, kurse nuk sigurohet se i kanw mbwrritur tw gjitha paketat, pra nuk i detekton paketat e humbura dhe rrjedhimisht nuk wshtw i besueshwm.

# FIEK - TCP

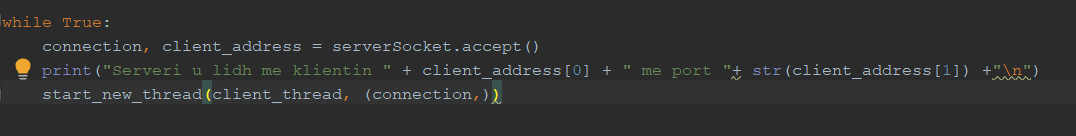
## 3.1 TCP Serveri - Pwrshkrimi i kodit

Synimi wshtw tw realizohet njw aplikacioni server i cili do tw jetw gjithmonw aktiv dhe nw pritje tw interaksionit me klientin.

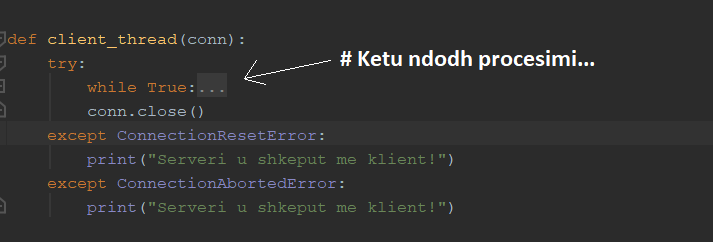
Sw pari serverit i caktohet njw adresw dhe nje port tek i cili ai do tw dwgjojw dhe presw komanda. Nw vijim krijohet socketi i serverit apo socketi dwgjues me metodwn .socket(), duke i dhwnw dy parametra: familjen e IP adresave, nw rastin tonw AF\_INET, ku i referohemi familjes sw IP adresave IPv4 dhe protokollin qw do ta ndjekw pwr komunikim, nw rastin tonw SOCK\_STREAM qw i referohet protokollit TCP. Socketin e krijuar e asociojmw me metodwn .bind() me IP adreswn dhe portin tani mw tw dhwnw. Me metodwn .listen(), e vendosim serverin nw gadishmwri tw pranojw komanda kurse parametri (nw rastin tone 5) tregon se sa klient mund tw lidhen me serverin nw tw njejtwn kohw.



Me anwn e njw unaze qw nuk pwrfundon asnjwhere e mbajmw serverin githnjw aktiv, pwrveç nw rastin kur ndodh ndonjw gabim qw shpie nw mbylljen e serverit. Me metodwn .accept() marrim dy argumente siç janw lidhja me klient dhe adresa e tij. Pasi ne duam qw serveri tw manovrojw me disa kwrkesa nga klient tw ndyrshwm njwkohwsisht ne pwrdorim teknikwn ‘multi-thread’, tw cilwn e vwmw nw veprim me thwrritjen e funksionit start\_new\_thread().



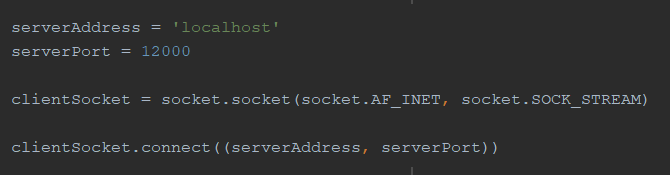
I gjithw procesimi tani kryhet nw metodwn e krijuar client\_thread(). Serveri dhe klienti komunikojnw mes vete me funksionet .recv() pwr tw marrw infromacion dhe .send() pwr ta dwrguar atw te klienti. Pas interaksionit server-klient pastrohet lidhja me anw tw funksionit .close(), qw e bwn serverin tw gatshwm pwr klientat e ardhshwm.



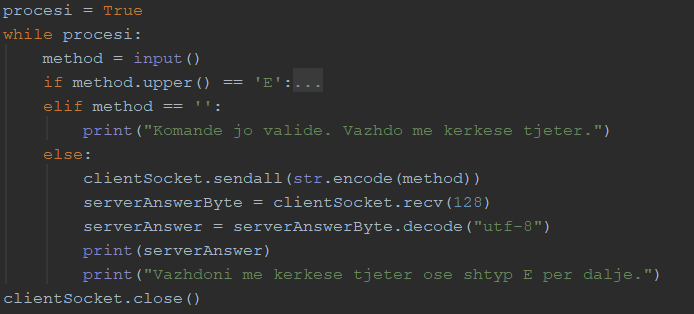
Metodat .socket(), .bind(), .listen(), .accept() merren nga libraria *socket*, kurse metoda start\_new\_thread merret nga libraria *\_thread*.

## 3.2 TCP Klienit – Pwrshrkimi i kodit

Ngjashwm sikurse tek serveri, do tw krijojmw edhe socketin e klientit me funksionin .socket() nga libraria socket. Kwtw socket e lidhim me socketin e serverit pwrmes metodws .connect() ku i japim dy parametra, IP adreswn dhe portin e serverit.



Unaza pwrdoret pwr t’i dhwnw lirinw klientit qw tw pwrfundojw lidhjen me server kur ai dwshiron vetw, e jo pas njw komande tw vetme. Me metodat .send() dhe .recv() bwhet shkwmbimi i informatave mes dy socketave, kurse pas pwrfundimit tw punws me metodwn .close() klienti liron portin pwr tu pwrdorur nga ndonjw klient tjetwr.



## 3.3 Trajtimi i gabimeve

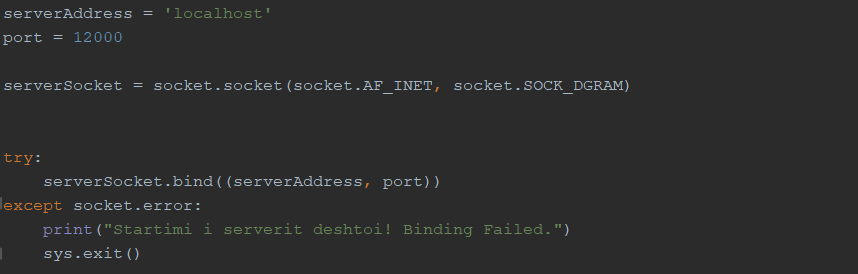
Siç edhe wshtw shfaqur nw fotografitw deri mw tash, janw pwrdorur komandat try: dhe except: pwr t’i trajtuar gabimet eventuale nw ekzekutimin e programeve. Qw serveri tw mos dwshtojw por tw qwndroj aktiv pwrgjatw kodit janw pwrdorur mjaft komanda try: dhe except: ku wshtw synuar tw parashikohen gabimet e mundshme siç janw TimeOutError, socket.error, MemoryError, ConnectionAbortError.. dhe tw shmangen ato.

# FIEK - UDP

## 4.1 UDP Serveri – Pwrshkrimi i kodit

Serveri qw pwrdorw protokollin UDP tani nuk i nevojitet qw tw konfirmojw lidhjen me klient, por menjwherw pasi tw vendoset lidhja ai do tw fillojw tw komunikojw me tw duke pwrdorur datagrama, dwrgimin e sw cilws gjithashtu nuk ka pse ta konfirmojw.

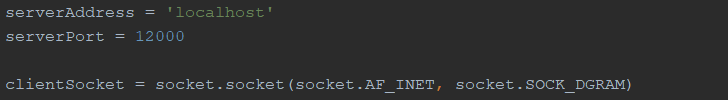
Pwr krijimin e socketit veprohet ngjashwm si nw rastin e protokollit TCP, por tani nw funksionin .socket() si parametwr i dytw i ipet SOCK\_DGRAM, gjw qw nwnkupton se ky program do tw zbatojw protokollin UDP.



Edhe komunikimi wshtw i menjwhershwm nga socketi nw socket pa ndonjw parametwr nw mes, qw do tw siguronte se nuk do tw kishte paketa qw dwshtojnw tw transmetohen.

## 4.2 UDP Klienti – Pwrshkrimi i kodit

Ngjashwm si twk aplikacioni Server, dallimi nw krijimin e socketit ndodhet nw parametrin e dytw qw i ipet funksionit qw krijon socketin, ku pwrcaktohet protokolli UDP.



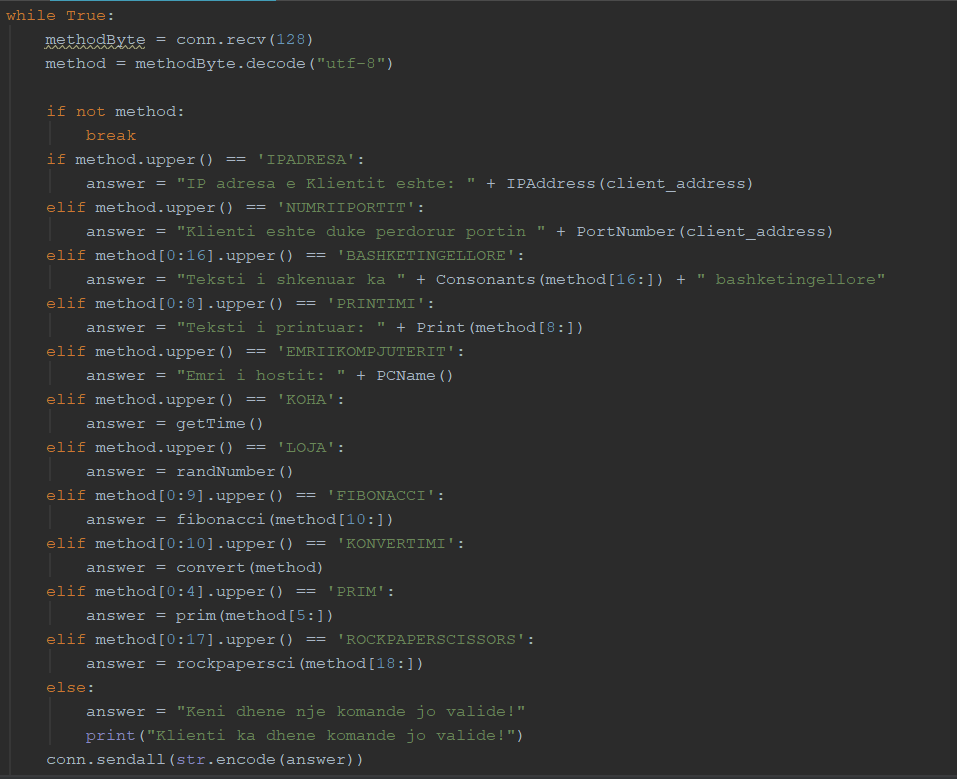
Forma e komunikimit gjithashtu wshtw direkte nga socketi nw socket, ku edhe caktohet adresa se ku duhet ta dwrgojw kwrkeswn, apo adresa e serverit.



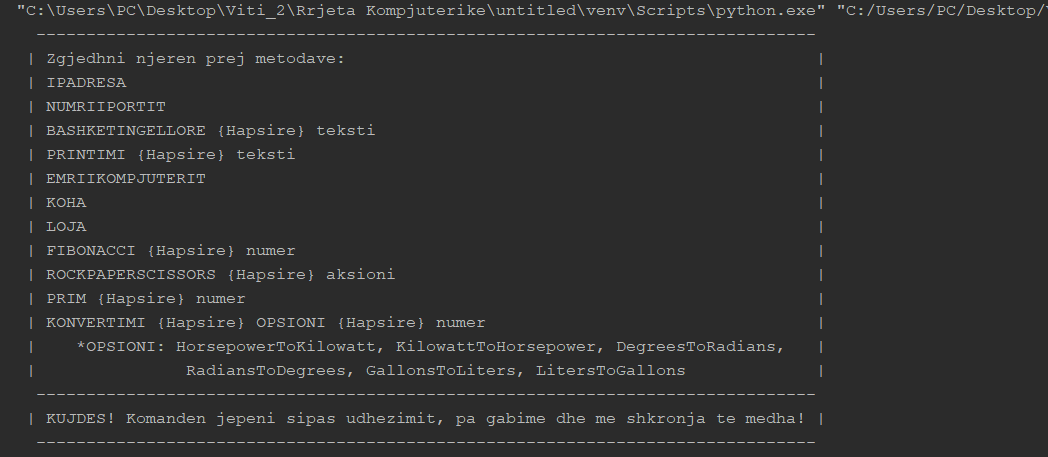
# Metodat

Nw kod wshtw paraqitur unaza kryesore sipas sw cilws operon serveri. Serveri merr kwrkeswn nga klienti me funksionin .recv(), e dekodon dhe varwsisht nga opsioni i zgjedhur i klientit e ri-drejton kwrkeswn nw funksionin pwrkatws. Ky filtrim bwhet pwrmes *if statements*. Varwsisht nga kwrkesa e klientit serveri merr pergjigjen dhe ia kthen atw klientit i cili e pranon dhe e shfaq nw ekran (pjesa e serverit). Serveri me anw tw *else statement,* kontrollon komandwm jo valide dhe kthen pergjigje te klienti qw tw provojw pwrswri me komandw tjetwr, pwr arsye qw tw mos dwshtojw aplikacioni Server.

\*Kjo wshtw foto e funksionimit tw serverit qw do t’i referohemi si foto\_1 pwr shpjegime tw mwtejme.

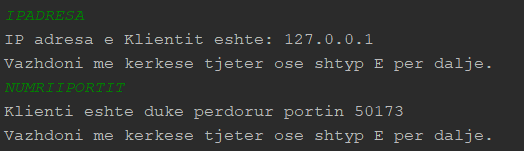


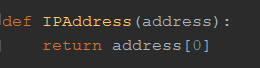
Paraqitja e opsioneve nw klient:



## 5.1 IPADRESA & NUMRIIPORTIT

Metodw e thjeshtw e cila merr si parametwr njw *tuple*, qw nw python paraqet njw sekuencw tw objekteve tw pandashwm. Argumenti i dytw qw merret me metodwn .accept() qw wshtw adresa e klientit *client\_address*, paraqet njw tuple, anwtari i parw i sw cilws wshtw IP adresa e i dyti porti. Kwshtu, nw foto\_1, thwrritet funksioni IPAddress e i ipet si parametwr client\_address (tuple i marrw nga funksioni accept). Ngjashwm me IP adreswn gjehet edhe porti tw cilin wshtw duke e pwrdorur klienti, i cili pwrdor anwtarin e dytw tw tuple-it client\_address. Nw anwn e djathtw shihet pwrgjigja nw klient.





## 5.2 BASHKETINGELLORE

Metoda qw i gjen bashkwtingwlloret nw njw tekst wshtw krijuar me njw unazw. Unaza lwviz nwpwr tw gjithw shkronjat e tekstit tw dhwnw. Teksti i dhwn me input i shkon funksionit si parametwr. Mirwpo nwse shohim *foto\_1*, shohim qw si parametwr funksionit i wshtw dhwn vetwm njw pjesw e ndarw e stringut tw marrw (me kllapa tw mesme ‘[ ]’), gjw qw na lejon tw marrim vetwm pjeswn e stringut pas komandws BASHKETINGELLORE, pra tekstin tw cilin duhet kontrolluar.

