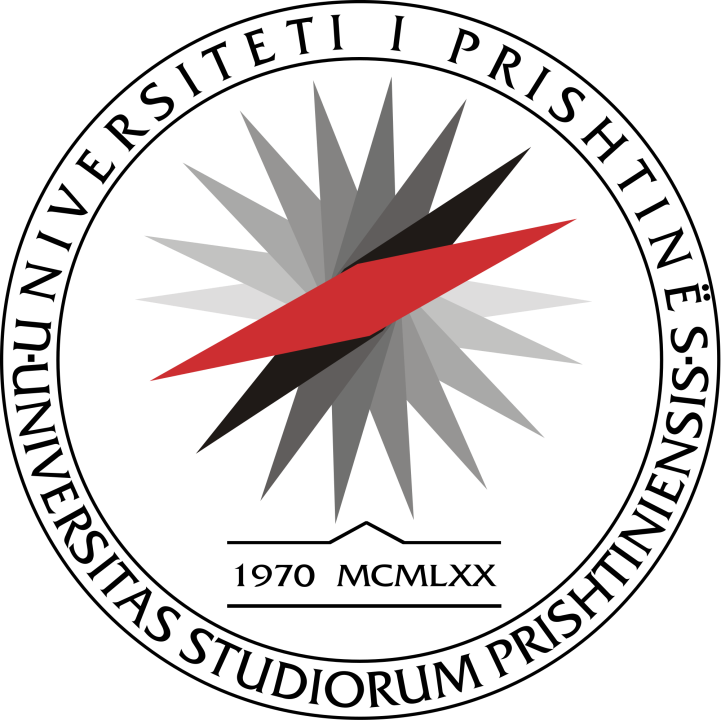
UNIVERSTETI I PRISHTINËS “HASAN PRISHTINA”  
FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE

DEPARTAMENTI I KOMPJUTERIKES



RAPORT: DIZAJNIMI KLIENT-SERVER

LËNDA: Rrjeta Kompjuterike

**Studenti**: Fortesa Hysenaj **Asistenti**:

**ID**: 170714100003 M.Sc. Haxhi Lajqi

Prishtinë, 2019

Përmbajtja

[**Përshkrimi 2**](#_Toc511249517)

[**Hyrje 3**](#_Toc511249518)

[**METODAT 4**](#_Toc511249520)

[IPADDRESA 4](#_Toc511249521)

[NUMRI I PORTIT 4](#_Toc511249522)

[BASHKETINGELLORE 5](#_Toc511249523)

[PRINTIMI 5](#_Toc511249524)

[EMRI I HOSTIT 6](#_Toc511249530)

[KOHA 6](#_Toc511249525)

[LOJA 7](#_Toc511249525)

[FIBONACCI 7](#_Toc511249526)

[KONVERTIMI 8](#_Toc511249527)

[PALINDROME 8](#_Toc511249528)

[DUPLIKIMI 9](#_Toc511249529)

[Përmbledhje rezultatesh 9](#_Toc511249529)

**Përshkrimi:**

Rrjetat kompjuterike janë të destinuara që të mundësojnë komunikimin me anë të pajisjeve.  
Për të arritur këtë komunikim duhet që të krijohet serveri se ku do të ruhen të dhënat që I ofrohen klientit. Projekti nga lënda “Rrjeta kompjuterike” ka pikërisht këtë për qëllim, që të implementohen metodat dhe njohuritë programuese që kemi fituar gjatë ligjëratave dhe ushtrimeve për të krijuar një server dhe një klient.   
Për arritjen e komunikimit server-klient na nevojiten socket-at, që llogariten si pika fundore që nevojiten për komunikim. Një socket në mënyrë unike ka një IP-adresë dhe një port.   
Varësisht nga kërkesat e klientit, ne duhej të programonim një sërë metodash të cilat kryejnë funksione të ndryshme. Duke përdorur dy protokolet TCP dhe UDP, të cilat i përkasin shtresës së transportit, kemi arrituar që të gjejmë zgjidhjen e problemit të komunikimit server-klient.  
Komunikimi në këtë rast kryhet me anë të console-s.

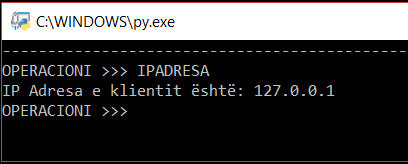
### Hyrje

Komunikimi është prej çështjeve kyçe në rrjeta kompjuterike. Dizajnimi i një serveri dhe klienti është mënyra se si të arrihet ai komunikim.  
Në kuadër të lëndës së Rrjetave Kompjuterike kemi zhvilluar një projekt, i cili krijon dhe dizajnon socket-at në anën e serverit dhe në anën e klientit.  
Duhet pasur parasysh se në implementimin e serverëve që kryejnë punë më të komplikuara do të qasen më shumë se një klient. Për këtë arsye në programin tonë duhet të arrihet ajo mundësi që më shumë se një klient të paraqesin kërkesa kundrejt serverit tonë.  
Poashtu, është kushtuar kujdes edhe asaj që nga kërkesat e klientit mos të dëmtohen të dhënat në server. Në rastin kur ndodh kjo, p.sh klienti bën kërkesë që nuk mund të i kthehet përgjegje nga serveri jonë, atëherë serveri njofton klientin që nuk mundet të i pergjigjet një kërkeseje të tillë dhe pastaj klienti do të lidhet sërish me serverin pas shkëputjes prej tij që të adresojë një kërkesë të re.  
Dallimi qenësor në mes te TCP-së dhe UDP-së është se TCP fillimisht krijon lidhje server-klient, ndërsa UDP nuk e kërkon këtë, që njëkohësisht rrit shpejtësinë dhe rrit rreziqet që të dhënat të shkëmbehen pa garancion të plotë.  
Për krijimin e këtij projekti është përdorur gjuha programuese Python dhe vegla PyCharm 2018. Testimi është bërë në sistemin operativ Windows 10 Pro.

# METODAT

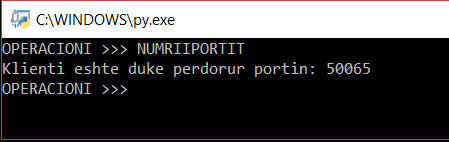
### IPADDRESA

Kjo metodë ka për qëllim që të kthejë si rezultatë IP Adresën e klientit. Nëse klienti zgjedh metodën IPADRESA ai do të marrë përgjigje nga serveri për IP Adresën e tij.



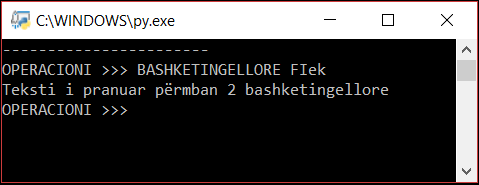
### NUMRIIPORTIT

Kjo metodë na jep portin e klientit. Pranon adresën e klientit me të cilën është lidhur në socket dhe nga ajo merr anëtarin e dytë.



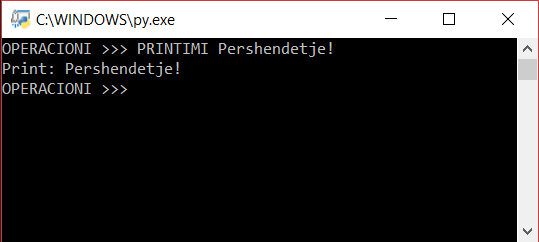
### BASHKETINGELLORE

Me anë të kësaj metode klienti shkruan një fjalë/fjali dhe e dërgon kërkesën që kjo metodë të gjejë numrin e bashketingelloreve të cilat gjenden. Metoda nuk është Case-Sensitive, pra klienti mundet të shkruajë fjalët edhe me shkronja të vogla edhe me shkronja të mëdha.



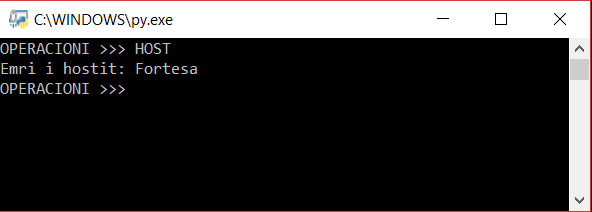
### PRINTIMI

Me anë të kësaj metode klienti shkruan tekstin dhe e dërgon kërkesën që kjo metodë të printojë atë që klienti e shkruan. Për shembull, nese klienti shkruan fjalën “Pershendetje!” e njejta do të printohet nga metoda.



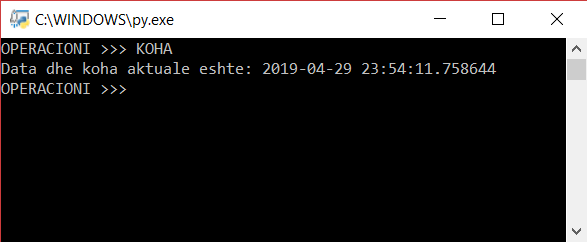
### EMRI I HOSTIT

Klienti përmes kësaj metode kërkon emrin e kompjuterit/hostit dhe serveri e kthen atë si përgjigje. Në qofte se emri i kompjuterit nuk mund të gjendet atëherë kthehet mesazhi “Emri i hostit nuk u gjet!”.



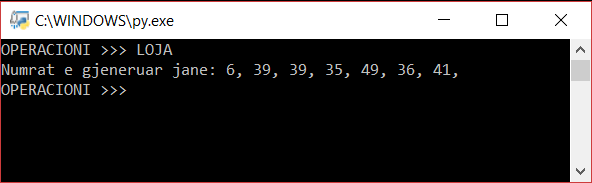
### KOHA

Kjo metodë përcakton kohën aktuale në server që e dërgon te klienti si përgjigje kur klienti e kërkon atë.



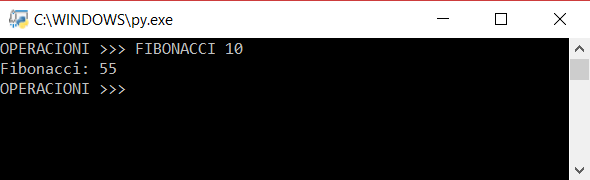
### LOJA

Kjo metode eshte e atille qe kur klienti shkruan metoden LOJA atëherë në mënyrë automatike do të gjenerohen 7 numra të rastesishem (random) ne intervalin 1-49. Për këtë është përdorur metoda random.



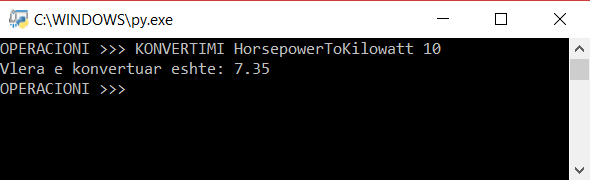
### FIBONACCI

Numrat Fibonacci janë seria e numrave, ku secili numër (numër Fibonacci) është shuma e dy numrave paraprake. Sa per ilustrim: 0, 1, 1, 2, 3, 5, 8, etj. Është shkruar formula në metodë. Klienti do të shkruajë metodën dhe një numër, p.sh. numrin 10, i cili rezulton me 55 dhe që do të thotë se me 10 numrat e parë arrihet shuma 55.



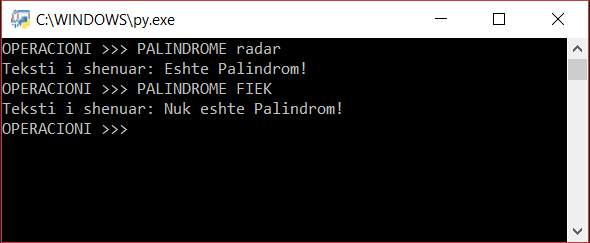
### KONVERTIMI

Për të bërë konvertimin nga një njësi matëse në një tjetër, klienti do të përdorë metoden KONVERTIMI e cila bën konvertimet: Kilowatt në Horsepower, Horsepower në Kilowatt, Degrees në Radians, Radians në Degrees, Gallons në Liters, Liters në Gallons.  
Klienti shkruan emrin e metodes, zgjedh njërin nga opsionet (p.sh. HorsepowerToKilowatt) dhe shkruan një vlerë e cila pasi të dërgohet në server konvertohet në njësinë e kërkuar.



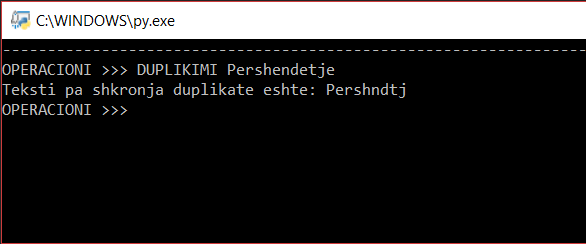
### PALINDROME

### Me anë të kësaj metode klienti shenon një fjalë, frazë ose sekuencë që lexohet njëjtë nga e djathta në të majtë dhe anasjelltas, p.sh., RADAR, ARRA, ANA dhe si rezultat do tregohet nëse teksti është palindrom apo nuk është!



### DUPLIKIMI

Kur klienti e zgjedh këtë metodë dhe e shënon një tekst, metoda si rezultat do ti largojë shkronjat që përsëriten dy apo më shumë herë dhe do të printojë tekstin vetëm me shkronjat unike.



**Përmbledhje rezultatesh**

Nga printscreen-at e mësipërm shihet se të gjitha metodat funksionojnë dhe kthejnë rezultat të

saktë.

Puna e klient-server vazhdon kështu deri sa klienti të shtyp tastin ENTER.