**Universiteti i Prishtinës “Hasan Prishtina”**

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

**Rrjeta Kompjuterike**



**Programimi me soketa**Valton Sherifi  
  
Prill 2019,

**Përmbajtja**

**1.Hyrje .....................................................................................................**

**2.Përshkrimi...............................................................................................**2.1 Veglat e përdorura   
2.2 Përshkrimi i Serverit dhe Klientit (TCP dhe UDP)  
2.3 Përshkrimi i kodit

**3.Metodat dhe rezultatet e testimit .............................................................** 3.1 Dritarja para/pasi qe klienti te kyqet  
3.2 Metoda IPAdresa  
3.3 Metoda nrPort  
3.4 Metoda Bashketingelloret  
3.5 Metoda Printimi  
3.6 Metoda EmriIKompjuterit  
3.7 Metoda Kohe  
3.8Metoda Loja  
3.9 Metoda Fibonacci  
3.10 Metoda Mbledhja 3.11Metoda Konvertimi 3.12Metoda HiqNumrat

**1. HYRJE** Ky projekt ka pasur qëllim të krijoj një rrjetë me anë të së cilës mundësohet komunikimi mes një serveri dhe një klienti. Ky komunikimi është bërë i mundshëm me përdorimin e Socket-ave. Protokollet të cilat janë përdorur gjatë komunikimit të këtyre dy sistemeve janë TCP dhe UDP.   
 Fillimisht mes serverit dhe klientit është krijuar një lidhje. Klienti lexon nga konsola disa të dhëna të cilat tregojnë se lidhja mes dy sistemeve është krijuar me sukses dhe se tashmë klienti mund të zgjedh njërën nga metodat. Pas zgjedhjes së metodës dhe shënimit të të dhënave, ato dërgohen në server i cili varësisht nga metoda që ka zgjedhur klienti i modifikon ato të dhëna. Në fund të dhënat e modifikuara i kthehen përsëri klientit.

Projekti përmban këto metoda:

1.IPAdresa  
2.nrPortit  
3.Bashketingelloret  
4.Printimi  
5.EmriIKompjuterit  
6.Koha  
7.Loja  
8.Fibonacci  
9.Mbledhja 10.Konvertimi 11.HiqNumrat

**2.PËRSHKRIMI  
  
2.1 Veglat e përdorura**Për realizimin e lidhjeve të përmendura më lart janë përdorur këto vegla: **Python 3.6 si gjuhë programuese, Visual Studio 2017 si (IDE).**  
  
**2.2 Përshkrimi i lidhjes mes Serverit dhe Klientit (TCP dhe UDP)**

Protokollet TCP dhe UDP janë protokolle të cilat i takojnë "Transport Layer". Lidhjet mes serverit dhe klientit në këtë projekt janë krijuar duke përdorur këto dy protokolle.  
 **UDP (User Datagram Protocol)** është protokoll i cili për dallim nga TCP, është më pak i besueshëm. Nuk garanton që të dhënat e dërguara do të mbërrijnë në destinacionin e caktuar dhe se gjatë bartjes së të dhënave ka mundësi të mëdha të humbjes së tyre (kjo është një nga arsyet pse përdoret më shumë për të ashtuquajturat "Streaming Multimedia"). Veçori tjetër e kësaj lidhje është se për çdo të dhënë të dërguar nga klienti, severi do t'i pranoj ose jo ato të dhëna.  
 **TCP (transmission control protocol)** është protokoll që fillimisht krijon një lidhje stabile mes serverit dhe klientit e pas kësaj fillon të bëjë transportimin e të dhënave. Është protokoll i besueshëm te i cili nuk humbin të dhënat dhe dërgimi i tyre shkon në mënyrë të renditur. Lidhja krijohet në fillim prandaj nuk ka nevojë që serveri çdo herë të pranoj ose jo të dhënat.

**3.METODAT DHE REZULTATET E TESTIMIT**

**3.1 Dritarja para/pasi klienti te kyqet**

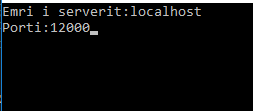
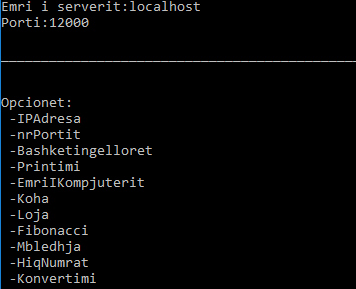
****

Figure 1: Pamja para/pasi që klienti te lidhet me server.

Klienti mund të shkruaj emrin e serverit dhe nr. e portit, pasi qe ti shenoj klienti lidhet me server ku i hapet lista me opcione(metodat) te cilat mund ti perdor duke i shkruar ato mire.

**3.2 Metoda IPADDR**Metoda IPAdresa i kthen klientit IP ADRESEN e tij.

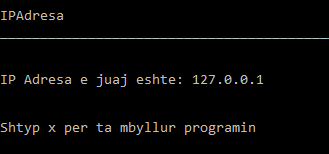


Figure 2: Rezultati pas përdorimit te metodës IPAdresa.

Pasi klienti te shtyp IPAdresa ai mund ta sheh ipadresenë e tij.

Kjo bëhet e munder permes metodes clientAddress ne server.

**3.3 Metoda nrPortit**Metoda nrPortit i kthen klientit portin në të cilin është lidhur me serverin.

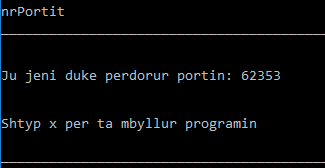


Figure 3: Rezultati pas perdorimit te metodes nrPortit.

Ngjashëm si me ipaddress, pasi klienti të shtyp nrPortit i shfaqet numri i portit te cilin po e perdor.

**3.4 Metoda Bashketingelloret**Metoda Bashketingelloret i kthen klientit numrin e bashkëtingëlloret të fjalës të cilën e ka shënuar si input.

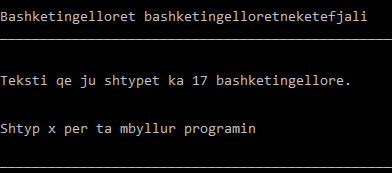


Figure4: Rezultati pas perdorimit te metodes Bashketingelloret.

Kjo metod e mer tekstin e klientit te cilën e ruan në një string, dhe përmes unazës for shiqohet secila shkrojn se a bënë pjesë në variable e deklaruar “bashketingelloret”, ëer secilën shkronjë qe bënë pjesë, e rrisim “nr” per 1 dhe në fund kthejm rezultatin

"Teksti qe ju shtypet ka "+ str(nr) +" bashketingellore."

**3.5 Metoda Printimi**Metoda Printimi i kthen klientit fjalinë që e ka shkruar si input.

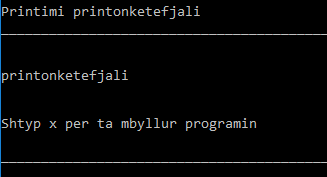


Figure 5 : Rezultati pas perdorimit te metodes Printimi.

Metoda Printimi merë tekstin pas fjalës Printimi, te cilen e kthen prap te klienti.

Në rast që klienti nuk shtyp asgjë pas Printimi shfaqet mesazhi:

Pas 'PRINTIMI' shtypeni tekstin!

**3.6 Metoda EmriIKompjuterit**Metoda EmriIKompjuterit i kthen klientit emrin e hostit.

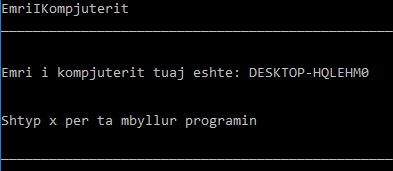


Figure 6 :Rezultati pas perdorimit te metodes EmriIKompjuterit.

Përmes metodes get.hostname(), klienti me shtypjen e opcionit EmriIKompjuterit i kthehet pergjigjia me emrin e kompjuterit të tij. Me rast se nuk gjindet emir I kompjuterit të klientit kthehet pergjigjia : Emri i kompjuterit tuaj nuk dihet.

**3.7 Metoda Koha**Metoda Koha i kthen klientit kohën në momentin e thirrjes së metodes.

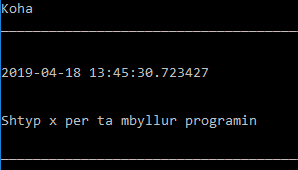


Figure 7:Rezultati pas perdorimit te metodes Koha.

Opcioni apo metoda Koha kthen kohe aktuale te thirrjes se metodes, kjo behet e mundshme permes serverit me thirrjen e metodes datetime.datetime.now().

**3.8 Metoda Loja**Metoda Loja i kthen klientit 7 numra random prej 1 deri në 49.

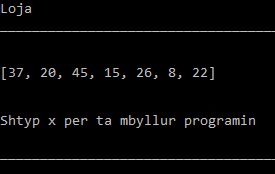


Figure 8:Rezultati pas perdorimit te metodes Loja.

Kjo metodë është e mundëshme permes metodes random.sample(range(1, 49), 7) ku (1,49) definojn shtrirjen e numrave prej 1 deri n 49 ndërsa numri 7 sasijen e numrave te cilat shfaqen ne dritare.

**3.9 Metoda Fibonacci**Metoda Fibonacci i kthen klientit numrin e n-të me rradhë që ndodhet në serinë fibonacci.

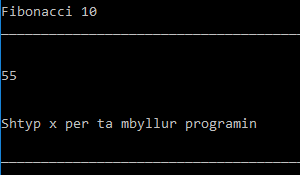


Figure 9:Rezultati pas perdorimit te metodes Fibonacci.

Duke e shfrytëzuar formulen e Fibonaccit ne nje funksion te serverit

def fib(n):

if (n <= 1):

return n;

return fib(n-1) + fib(n-2);

i mundesohet klientit qe pas Fibonacci ta shenojn nje numer dhe ta din se numri i N-të

qfar vlere ka në vargun e Fibonaccit.

**3.9 Metoda Mbledhja**Metoda Mbledhja mbledh dy numra.

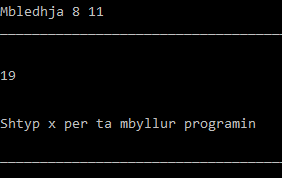


Figure 10:Rezultati pas perdorimit te metodes Mbledhja.

Klienti me thirrjen e metodës Mbledhja duhet ti shenoj dy numra me hapsirë mes numravë për ta pare ne rezultatin qe kthen server mbledhjen e atyre dy numrave. Dy numrat ruhen ne variabla te ndryshme ku pastaj ne variablën sum ruhet bledhja e tyre dhe klientit ne dritare I shfaqet rezultati.

elif(inputklienti[0]=='Mbledhja'):

try:

first\_number =(int(inputklienti[1]))

second\_number = (int(inputklienti[2]))

sum = first\_number + second\_number

connection.send(str.encode(str(sum)))

**3.10 Konvertimi**

Metoda Konvertimi bene konvertimin e njesive.

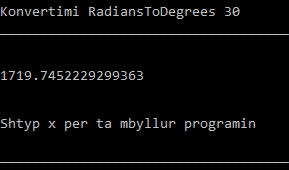


Figure 11:Rezultati pas perdorimit te metodes Konvertimi.

**3.11 HiqNumrat**

Metoda HiqNumrat heq numrat nga një input I klientit.

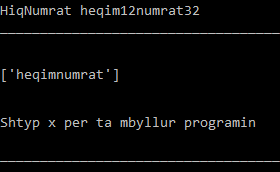


Figure 12:Rezultati pas perdorimit te metodes HiqNumrat.

Klienti ka mundësin se përmes ksaj metode ti heq numrat në një fjali.  
Kjo bëhet e mundeshme permes

rezultati = ''.join(i for i in stringu if not i.isdigit())

**4.1 Pamja kur startohet server**

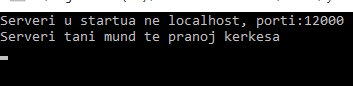
****

Figure Pamja pasi ti aktivizohet server

Në këtë foto shihet se serveri është startuar dhe po pret per kerkesa, deri më tani shihet nuk është shtuar as nje lidhje.

**4.2 Pamja pasi te shtohet nje lidhje**

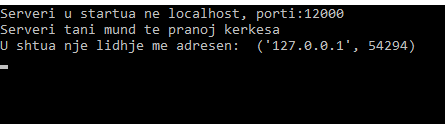
****

Figure Lidhja e shtuar dhe ip adresa e klientit

Në këtë foto shihet që serveri tregon që është lidhur një klient dhe tregon ip e klientit gjithashtu dhe portin. Me këtë server është e mundëshme edhe lidhja e më shumë klienteve.. deri në 10 klient permes metodes server.listen(5).

**4.3 Me shum se 128 karaktere**

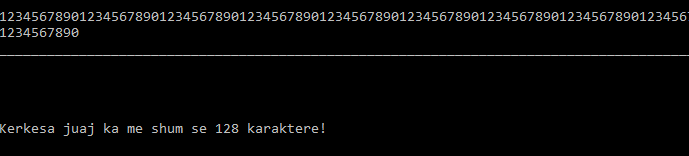


Figure Klienti shtyp me shume se 128 karaktere

Me rast se klienti shtyp me shumë se 128 karaktere, at’herë kthehet mesazhi “Kerkesa juaj ka me shum se 128 karaktere!”, ne kodin e klientit, inputi merren ne nje variable ‘kerkesa’ ku pastaj ajo permes IF shikohet se nqs ka me shum 129 karaktere nuk pranohet kerkesa dhe del mesazhi I lartshenuar, me rast se ka me pak se 129 karaktera ‘kerkesa vazhdon…

kerkesa=kerkesa.strip()

if len(kerkesa) > 129:

print("\n\nKerkesa juaj ka me shum se 128 karaktere!")

continue

**4.4 Mbyllja e dritares se klientit**

Pasi klienti të përdor njërën apo më shumë metoda(opcione), ka mundësin qe ta mbyll dritarën duke shtypur tastin “x”. Kjo bëhet e mundshme ne klient permes If, ku inputi klientit ruhet ne ‘kerkesa’ dhe nqs kerkesa është ‘x’ at’herë mbyllet dritarja.

if kerkesa == "x":

client.close()

break