**Universiteti i Prishtinës**

**Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike**

**Lënda: Rrjetat Kompjuterike – 2019**



Tema:

**Dizajnimi Server-Klient**

**Studenti: Xheneta Hasani**

**Assistenti:M.Sc.Haxhi Lajqi**

**Data:18/04/2019**

**Vendi:Prishtine**

# Permbajtja

## 1. Veglat e perdorura, versionet e tyre dhe lista e metodave te implementuara

## 2. Hyrja

2.1 Shpjegim I shkurter per projektin

2.2 Skenari I klientit dhe serverit

## 3. Pershkrim I kodit te aplikacionit te serverit

3.1 Pershkrimi I serverit UDP

3.1.1 Logjika e aplikacionit

3.1.2 Realizimi permes kodit

3.2 Pershkrimi I serverit TCP

3.2.1 Logjika e aplikacionit

3.2.2 Realizimi permes kodit

## 4. Pershkrimi I kodit te aplikacionit te clientit (TCP dhe UDP)

4.1 Logjika e aplikacionit

4.2 Realizimi permes kodit

## 5. Pershkrimi I metodave

5.1 Pershkrimi I metodave pa hyrje

5.1.1 Metoda IP

5.1.2 Metoda PORT

5.1.3 Metoda HOST

5.1.4 Metoda TIME

5.1.5 Metoda LOJA

5.2 Pershkrimi I metodave me hyrje

5.2.1 Metoda ZANORE

5.2.2 Metoda PRINTO

5.2.3 Metoda FAKTORIEL

5.2.4 Metoda KONVERTO

5.2.5 Metoda VITI

## 6. Rezultatet e testimeve

Rezultatet e testimit te kerkesave te projektit me evidenca nga ekrani dhe permbledhje rreth funksionalitetit te metodave

## 6.1 Vertetimi I metodave

# Veglat e perdorura, versionet e tyre dhe lista e metodave te implementuara

Sistemi operativ:

* Windows 10

Veglat e perdorura dhe versionet e tyre:

* Microsoft Visual Studio 2017 Community,(Python 3.6)

Librarite e perdorura:

* From
* Sys
* Math
* Into

Klasat e perdorura:

* Socket
* Exception
* Thread
* Encoding
* DateTime
* Random
* String
* Sys
* Range
* Str
* List

Lista e metodave te implementuara:

* IP
* PORT
* ZANORE
* PRINTO
* HOST
* TIME
* LOJA
* KONVERTO
* FAKTORIEL
* VITI

# 2.Hyrja

## 2.1 Shpjegim I shkurter per projektin

Projekti perbehet nga programi i klientit dhe serverit te cilet komunikojne permes FIEK Protokollit.

Komunikimi ndermjet klientit dhe serverit behen permes socket-ave te cilat mundesojne komunikimin e tyre ne portin **11000** ose ne nje port te vecante te cilin mund ta caktoje klienti.

FIEK Protokolli eshte nje protokoll I thjeshte I komunikimit I cili per baze ka fjalen kyqe ne fillim te kerkeses e cila percakton metoden te cilen do ta kthej serveri. Ajo ka dy versione:

**1.FIEK-TCP**

**2.FIEK-UDP**

Ky protokoll ju pergjigjet vetem ketyre kerkesave: **IP, PORT, ZANORE, PRINTO, HOST, TIME, LOJA, KONVERTO** si dhe metodave individuale te cilat dergohen nga klienti ne server dhe per te cilat serveri kthen nga nje pergjigjje te vecante,.Klienti se bashku me kerkesen mund te dergoje edhe input varesisht nga kerkesa.

## 2.2 Skenari I klientit dhe serverit

**FIEK-UDP**:

Ne versionin UDP te prokollit FIEK komunikimi realizohet vetem permes UDP datagram-ve, ku me c’rast nuk krijohet nje lidhje(segment) mes klientit dhe serverit por komunikimi fillon drejtperdrejt(pa ndonje ndermjetesues). Klienti dergon kerkesen tek server permes nje UDP datagrami. Pasi qe kerkesa te validohet ne server ajo poashtu kthehet tek klienti permes UDP datagram-it. Protokolli FIEK limiton qe klienti te dergon vetem nje kerkese per datagram.

**FIEK-TCP:**

Se pari vendoset nje lidhje(segment) ne mes te klientit dhe serverit permes socket-ave ne portin e caktuar(default 11000). Klienti permes command line e dergon kerkesen tek server, I cili nese kerkesa eshte valide I pergjigjet kerkeses specifike te klientit dhe me pas mbyll lidhjen me klientin. Serveri eshte ne gjendje te lidhet me me shume kliente ne te njejten kohe dhe ti kthej pergjigje klienteve ne menyre paralele.

# 3. Pershkrim I kodit te aplikacionit te serverit

## 3.1 Pershkrimi i serverit UDP

### 3.1.1 Logjika e aplikacionit

Serveri është console aplikacion që lidhet me klientin përmes socket-ave dhe komunikon me datagram. Serveri dallon llojet e kërkesave që vijnë nga klienti dhe për secilin lloj ka përgjigje të veqant. Serveri është aplikacion që gjendet në unazë të pafundme gjë që nuk e lejon të ndalet, pra gjate tërë kohës është aktiv. Për një datagram serveri mund të dërgoj vetëm një datagram.

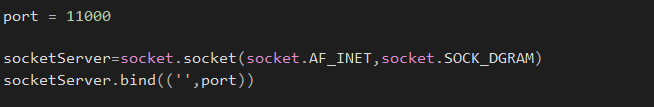
### 3.1.2 Realizimi permes kodit

Se pari importojme klasat te cilat na nevojiten per kodin ne vazhdim siq jane: **Socket-at, Math, Datetime,**

**String, Random, Sys.**



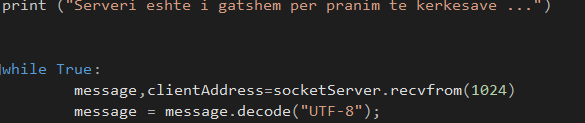
Vendosim nje port te dedikuar I cili na mundeson qe te komunikojm( ne kete rast porti 11000).Serveri do te tentoje te pret te dhena nga klienti permes ketij porti.

Thonjezat e njefishta nenkuptojne qe ky socket eshte i çasshem nga çfardo adrese që e ka serveri.

Ne kete menyre kur dergohet ndonje pakete ne portin 11000 në IP adresën e serverit, paketa do të drejtohet për tek ky socket. Pastaj serveri hyn në një loop të pafundme (while 1 ose while True) per te pranuar dhe procesuar paketat nga klientet.

Ne Console do na shfaqet “Serveri eshte I gatshem per pranim te kerkesave …”.

Mesazhin e dekodojme dhe me pas duhet ta enkodojme me po te njejten menyre.



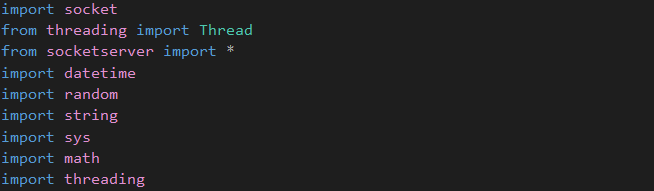
Rreshti i fundit eshte i ngjajshem me pjesen e kodit tek **UDP-Klienti**. Ketu nga çdo pakete që vjen në memorien e perkohshme, te dhenat e paketes ruhen ne variablen *message*, ndersa adresa e derguesit (IP dhe porti) ruhet ne në variablen ***clientAddress*.** Adresa e derguesit nevojitet ne rast se deshirojme t’I dergojmë ndonje përgjigjje klientit.

## 3.2 Pershkrimi I serverit TCP

### 3.2.1 Logjika e aplikacionit

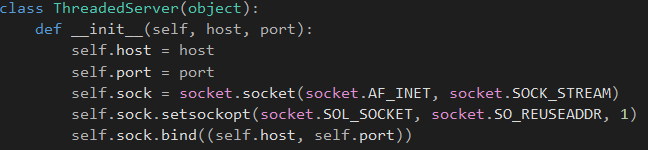
Serveri është console aplikacion që lidhet me klient përmes socket-ave dhe për secilen lidhje me klient krijon një thread të ri. Serveri dallon llojet e kërkesave që vijnë nga klienti dhe për secilin lloj ka përgjigje të veqantë. Serveri është aplikacion që gjendet në unazë të pafundme gjë që nuk e lejon të ndalet, pra gjate tërë kohës është aktiv.

### 3.2.2 Realizimi permes kodit

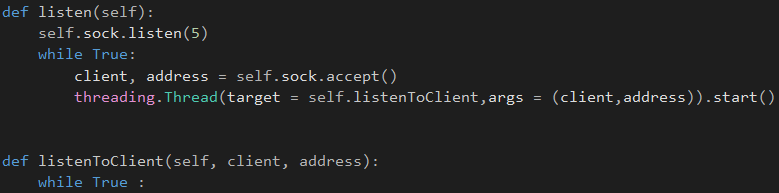


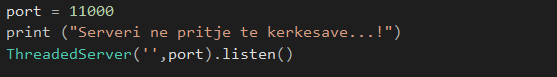
Importojme klasat qe nevojiten per metoda e perfunduara ne menyre qe ato te jene plotesisht funksionale.

Klasa ClientThread mundeson definimin e threadeve ne menyre qe te pranoje me shume kliente.  
Permes funksionit \_\_init\_\_ merren IP dhe porti I klientit te ri .Ndersa permes funksionit run realizohen kerkesat e klienteve.



Definojme portin e serverit dhe krijojme socket-in e serverit dhe permes nje vektori threads I marrim thread-at e klienteve. Poashtu permes nje teksti tregojme qe server eshte I gatshem.





Permes unazes e leme serverin ne dispozicion te perhershem.  
Lejojme serverSocket-in te pranoj lidhjet me kliente me c’rast krijohet connectionSocket-i.  
Krijojme thread te ri per secilin kliente te cilat I ruajme ne thread-s vektorin.

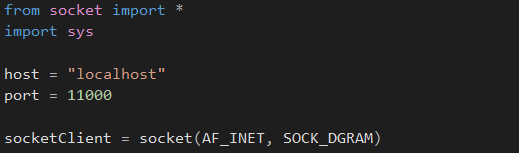
# 4. Pershkrimi I kodit te aplikacionit te clientit (TCP dhe UDP)

## 4.1 Logjika e aplikacionit

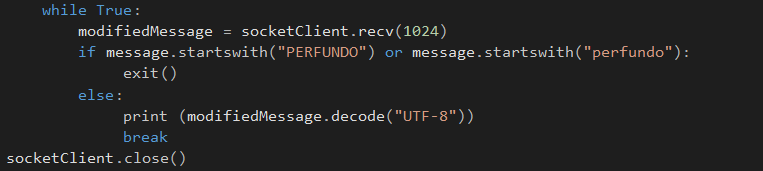
Klienti eshte nje windows command line e cila pas startimit pyet shfrytezuesin(end-user) se a deshiron te konektohet ne serverin default (127.0.0.1:11000) apo ne ndonje server tjeter.

Perndryshe klienti konektohet ne serverin default me c’rast klientit I mundesohet dergimi I komandave tek serveri.

## 4.2 Realizimi permes kodit



Importojme Socket-in, dhe definojme adresen te cilen nese mund ta vendosim edhe si localhost per te komunikuar Brenda nje kompjuteri.  
Definojme nje porte te caktuar me te cilin komunikon server me klient.  
Argument i parë (AF\_INET) nënkupton që përdorim IPv4 ndërsa i dyti (SOCK\_DGRAM) që ky është një UDP socket.



Programi e krijon një paketë dhe ia vendos adresën e destinacionit në mënyrë eksplicite(rasti UDP) Kurse ne TCP progami i klientit thjeshte dërgon të dhënat në TCP connection dhe pastaj pret për përgjigjje nga server.Dhe ne fund me ane te socketClient.close() mbyllet lidhja mes klientit dhe serverit.

# 5. Pershkrimi I metodave

## 5.1 Pershkrimi I metodave pa hyrje

### 5.1.1 Metoda IP

Kerkesa:

* IP

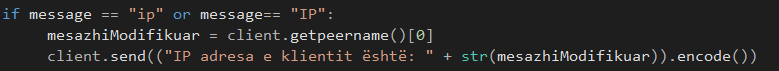
Aksioni:

* Përcakton dhe kthen IP adresën të klientit në formë dhjetore me pike në një tekst mesazh p.sh. (10.10.8.243)

Pergjigja:

* Një mesazh jo me i gjate se 128 karaktere qe përmban IP adresën e klientit.
* Mesazhi nuk guxon te përmbaj NULL karakterin

Realizimi permes kodit:

Shpjegimi I kodit:

* Ne console te klientit nese shenojme ip(IP) atehere na kthen Ip e klientit.

### 5.1.2 Metoda PORT

Kerkesa:

* PORT

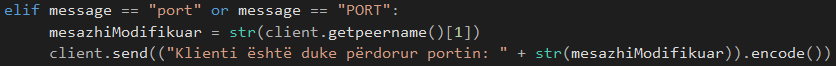
Aksioni:

* Përcakton dhe kthen portin e klientit. (Ky duhet te jete porti I klientit dhe jo porti I serverit.)Pergjigja:

Pergjigja:

* Një mesazh jo me i gjate se 128 karaktere qe përmban portin e klientit.
* Mesazhi nuk guxon te përmbaj NULL karakterin

Realizimi permes kodit:



Shpjegimi I kodit:

* Permes vetise message nga socket-i i klientit mirret numri i portit ne perdorim i cili ruhet ne variablen mesazhi.

### 5.1.3 Metoda HOST

Kerkesa:

* HOST

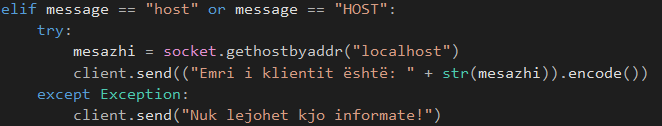
Aksioni:

* Kërkon emrin e hostit dhe e kthen atë. Nëse emri i hostit nuk mund te përcaktohet (gjendet) atëherë duhet te kthehet një mesazh dhe te tregohet se emri i hostit nuk mund te gjendet

Pergjigja:

* Një mesazh jo me i gjate se 128 karaktere qe përmban emrin e hostit ose te tregohet se emri i hostit nuk mund te përcaktohet.
* Mesazhi nuk guxon te përmbaj NULL karakterin.

Realizimi permes kodit:



Shpjegimi I kodit:

* Me ane te metodes message mirret emri i hostit i cili pastaj kthehet tek klienti.Ne qofte se marrja e emrit te hostit ka qene e pasukseshme klienti lajmerohet qe emri i hostit nuk eshte marre.

### 5.1.4 Metoda TIME

Kerkesa:

* TIME

Aksioni:

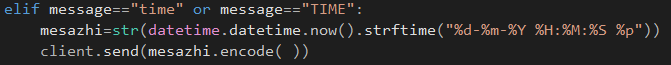
* Përcakton kohen aktuale ne server dhe e dërgon atë tek klienti si format te

lexueshme për njerëzit.

Pergjigja:

* Një mesazh jo me i gjate se 128 karaktere qe përmban kohen e serverit.
* Mesazhi nuk guxon te përmbaj NULL karakterin.

Realizimi permes kodit:



Shpjegimi I kodit:

* Permes klases DateTime mirret koha aktuale e sistemit dhe kthehet ne string ne formatin e caktuar.

### 5.1.5 Metoda LOJA

Kerkesa:

* LOJA

Aksioni:

* Kthen 20 numra nga rangu [1,99].

Pergjigja:

* *Tekst psh. (1,4,9,14,23,28,30,35,39,43,48,55,61,96,70,81,74,76,77, 90 )*

Realizimi permes kodit:



Shpjegimi I kodit:

* Duke perdorur klasen Random I cili perdoret per gjenerimin e numrave te rastesishem,gjenerojme 20 numra te qfardoshem .Kemi perdorur metoden sample e cila permban dy parametra, ne parametrin e pare te se ciles jepet kufiri I numrave dhe ne te dytin numrin e deshiruar te antareve ne list.

## 5.2 Pershkrimi I metodave me hyrje

### 5.2.1 Metoda ZANORE

Kerkesa:

* ZANORE{HAPESIRE}tekst

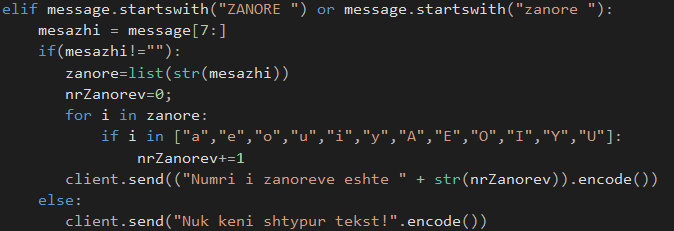
Aksioni:

* Gjen numrin e zanorëve ne tekst dhe kthen përgjigjen.

Pergjigja:

* Një mesazh jo me i gjate se 128 karaktere qe përmban numrin e karaktereve ne tekst.
* Mesazhi nuk guxon te përmbaj NULL karakterin

Realizimi permes kodit:



Shpjegimi I kodit:

* Se pari serveri shikon se a fillon teksti me “zanore” ose “ZANORE”.Mesazhin e pranuar e kthejm ne nje string ku permes substring-ut e ndajme ate tekst te derguar nga klienti ne menyre qe fjala kyce e metodes “zanore” te mos perfshihet ne tekst.Permes nje unaze dhe nje kushti e percaktojne numrin e zanoreve ne ate tekst.

### 5.2.2 Metoda PRINTO

Kerkesa:

* PRINTO{HAPESIRE}tekst

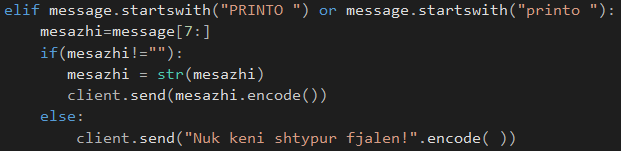
Aksioni:

* Kthen fjalinë e shtypur ne tekst.
* Hapësirat ne fillim dhe ne fund te fjalisë nuk duhet te kthehen

Pergjigja:

* tekst

Realizimi permes kodit:



Shpjegimi I kodit:

* Se pari serveri shikon se a fillon teksti me “printo” ose “PRINTO” , dhe pastaj permes metodes substring e ndane tekstin e derguar nga klienti ne menyre qe fjala kyqe e metodes “printo” te mos perfshihet ne tekst.

### 5.2.3 Metoda FAKTORIEL

Kerkesa:

* FAKTORIEL{HAPESIRE}numer

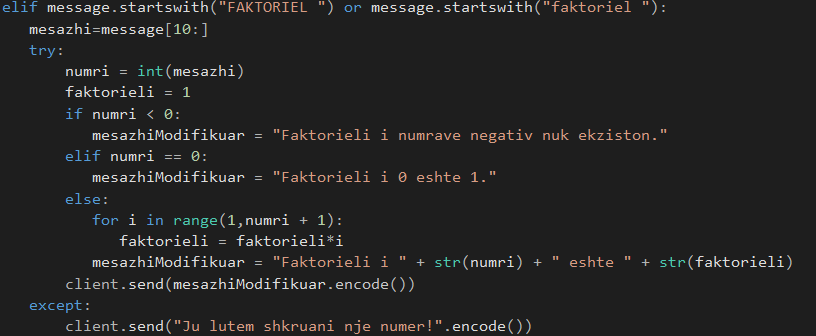
Aksioni:

* Gjenë faktorielin si rezultat i parametrit të dhënë hyrës.

Pergjigja:

* Numër i plotë psh. FAKTORIEL 5 kthen si rezultat 120

Realizimi permes kodit:



Shpjegimi I kodit:

* Se pari serveri shikon se a fillon teksti me “faktoriel” ose “FAKTORIEL” , dhe pastaj permes metodes substring e ndane tekstin e derguar nga klienti ne menyre qe fjala kyqe e metodes “printo” te mos perfshihet ne tekst.Pjesen e mesazhit te ndare e cast-ojme ne integer dhe e ruajme ne variablen numri. N.q.s numri eshte me I vogel se zero atehere shfaqe mesazhin “Faktorieli I numrave negative nuk ekziston”.Nese numri eshte I barabarte me zero shfaq mesazhin”Faktorieli I 0 eshte 1”, perndryshe e llogarite faktorielin e atij numri.

### 5.2.4 Metoda KONVERTO

Kerkesa:

* KONVERTO{HAPESIRE}opcioni{HAPESIRE}numer

Aksioni:

* Kthen si rezultat konvertimin e opcioneve varësisht opcionit të zgjedhur.
* Lista e parametrave opcioni janë:
* CelsiusToKelvin
* CelsiusToFahrenheit
* KelvinToFahrenheit
* KelvinToCelsius
* FahrenheitToCelsius
* FahrenheitToKelvin
* PoundToKilogram
* KilogramToPound

Pergjigja:

* Numër i plotë psh. KONVERTO CelsiusToKelvin 10 kthen rezultatin 283.15

Realizimi permes kodit:



Shpjegimi I kodit:

Se pari serveri shikon se a fillon teksti me “konverto” ose “KONVERTO” , dhe pastaj permes metodes substring e ndane tekstin e derguar nga klienti ne menyre qe fjala kyqe e metodes “printo” te mos perfshihet ne tekst.Permes metodes split mesazhi ndahet permes hapsires ne vektore me tre variabla.  
Permes kushteve te caktuara dhe varesishte nga kerkesa e klientit llogaritet kerkesa e caktuar.

### 5.2.5 Metoda VITI

Kerkesa:

* VITI{HAPESIRE} viti I lindjes

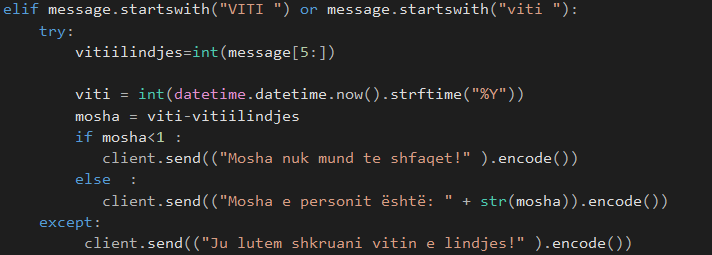
Aksioni:

* Gjenë moshen e klientit si rezultat i vitit të dhënë hyrës.

Pergjigja:

* Numër i plotë psh. Mosha e personit është: 1995 kthen si rezultat 23

Realizimi permes kodit:



Shpjegimi I kodit:

* Se pari serveri shikon se a fillon teksti me “VITI” ose “viti” , dhe pastaj permes metodes substring e ndane tekstin e derguar nga klienti ne menyre qe fjala kyqe e metodes “viti” te mos perfshihet.Pastaj përmes metodës datetime.now().strftime(“%Y”) e marrim vitin actual në cilin gjindemi dhe përmes kodit mosha = viti – vitiilindjes llogaritet mosha e personit.

# 6. Rezultatet e testimeve

## 6.1 Vertetimi i metodave

