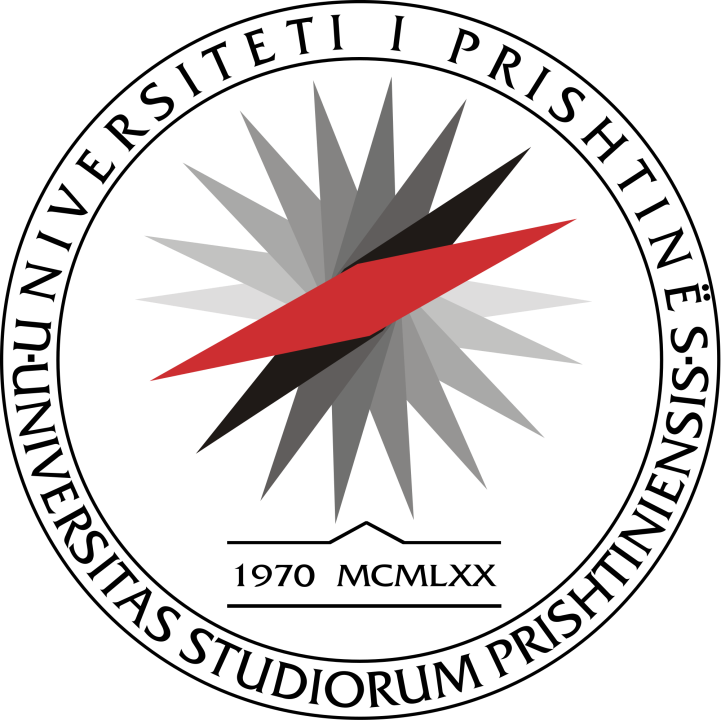
UNIVERSTETI I PRISHTINËS  
FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE



RAPORT : DIZAJNIMI KLIENT-SERVER

LËNDA: Rrjeta Kompjuterike

Profesori i lëndës: Prof. Blerim Rexha Punoi: Diare Daqi

Asistenti i lëndës: Msc. Haxhi Lajqi Nr.ID-së:190714100067

**Prishtinë,2021**

### Permbajtja:

[**Hyrje** 3](#_Toc70772222)

[Fiek-TCP-Protocol 4](#_Toc70772223)

[Fiek\_TCP\_Klienti 4](#_Toc70772224)

[Fiek\_TCP\_Serveri 6](#_Toc70772225)

[përshkrimi i metodave 8](#_Toc70772226)

[Testimi i metodave 9](#_Toc70772227)

[Metodat shtesë 13](#_Toc70772228)

[Fiek-UDP-Protocol 14](#_Toc70772229)

[Fiek\_uDP\_Klienti 14](#_Toc70772230)

[Fiek\_uDP\_serveri 14](#_Toc70772231)

[Testimi i disa metodave te Udp serveri 14](#_Toc70772232)

# 

# Hyrje

Projekti i parë nga lënda Rrjeta Kompjuterike ka për qëllim që të na ofrojë sa më shumë me përdorimin e soketave, mënyren sesi funksionojnë dhe për çfarë përdoren.

Në këtë raport u kërkua krijimi i një protokolli të quajtur FIEK, i realizuar në dy teknologjitë që i ofron shtresa Transport Layer, pra në TCP dhe në UDP.

Programimi i soketave është realizuar në gjuhën programuese Python duke përdorur libraritë si dhe funksionet e gatshme të kësaj gjuhe.Për realizimin e këtij projekti është përdorur vegla Microsoft Visual Studio 2019.

Disa koncepte themelore:

*Çfarë është serveri dhe çfarë detyra ka në këtë projekt?* -Serveri është një pajisje me një grup të caktuar të programeve ose protokolle që mundësojnë kryerjen e shërbimeve të ndryshme. -Në projektin tonë serveri ka për detyrë të pranojë kërkesa nga klienti dhe kerkesave valide të klientit t’i kthejë përgjigje duke i përdorur metodat e implementuara në të nga ana jonë.

*Çfarë është klienti dhe çfarë detyra ka në këtë projekt?* -Klienti është kërkues(shfrytëzues) i llojeve të ndryshme të shërbimeve që gjenden në server. -Në projektin tonë klienti ka për detyrë që vazhdimisht t’i bëjë kërkesa serverit.

*Çfarë është TCP?* -Transmision Control Protocol - -Një ndër protokollet më të rëndësishëm i orientuar drejt lidhjes që lehtëson shëmbimin e mesazheve ndërmjet pajisjeve kompjuterike në rrjet. Arsyeja që e bënë këtë protokoll kaq të rëndësishëm është besueshmëria e tij.TCP përdor një three-way handshake për të krijuar një lidhje të besueshme.

*Çfarë është UDP?* -User Datagram Protocol- -Me anë të UDP-së rrjetet kompjuterike mund t’i dërdojnë, mesazhe të shkurtra njëri tjetrit, të njohura si paketa.UDP nuk garanton reliabilitet sikurse bënë TCP.Paketat mund të vijnë të pa radhitura, të duken të dyfishta ose të humbasin pa paralajmëruar.

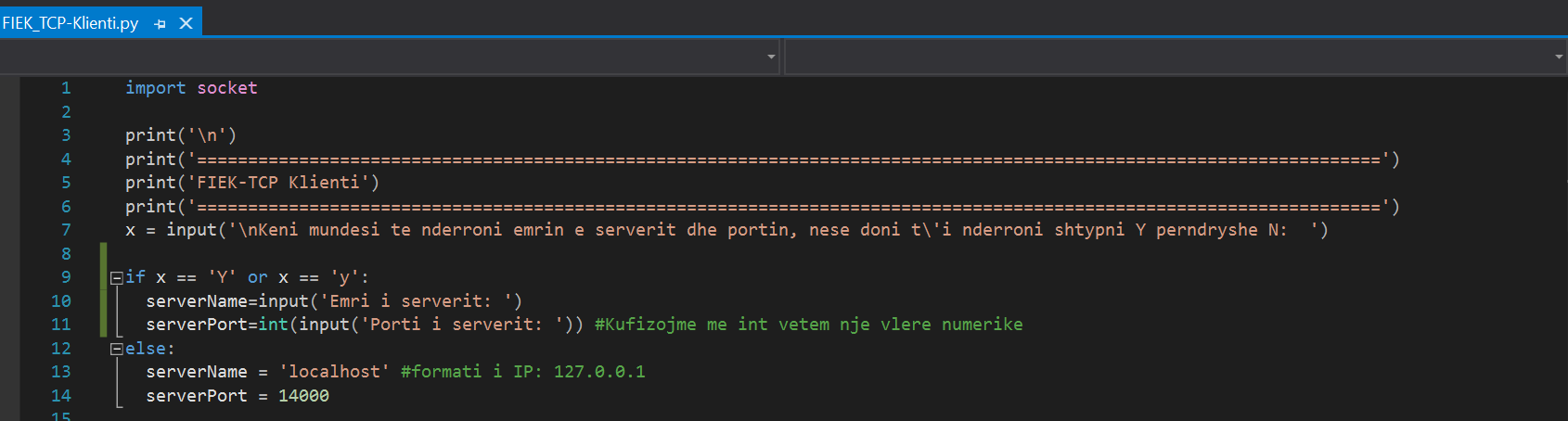
Çfarë janë soketat? -Soketat mundësojnë komunikimin midis dy proceseve të ndryshme qoftë në të njetën makinë punuese ose në makina të ndryshme.

Çfarë janë thread-at? -Thread-i është një sekuencë e udhëzimeve të tilla brenda një programi që mund të ekzekutohet në mënyrë të pavarur nga kodi tjetër. Një program me shumë thread-a mundëson që dy ose më shumë pjesë të ekekutohen njëkohësisht.Multithreaded mundësojnë që një soket i serverit të komunikojë më shumë se me një klient në të njejtën kohë në të njejtin rrjet.

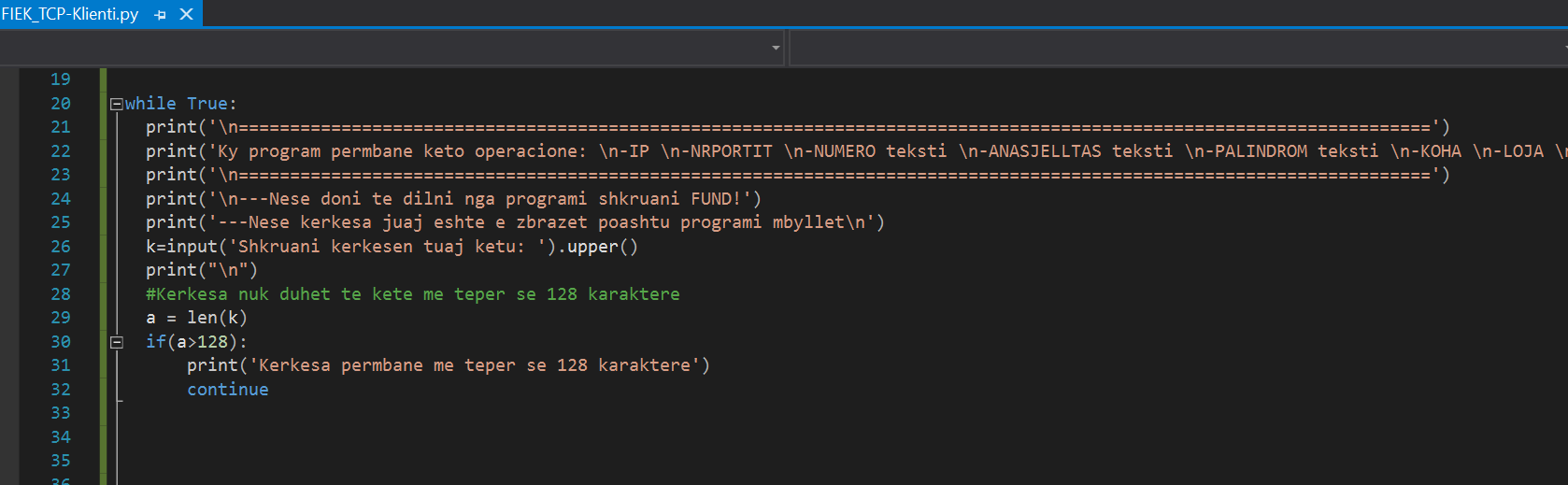
# Fiek-TCP-Protocol

# Fiek\_TCP\_Klienti

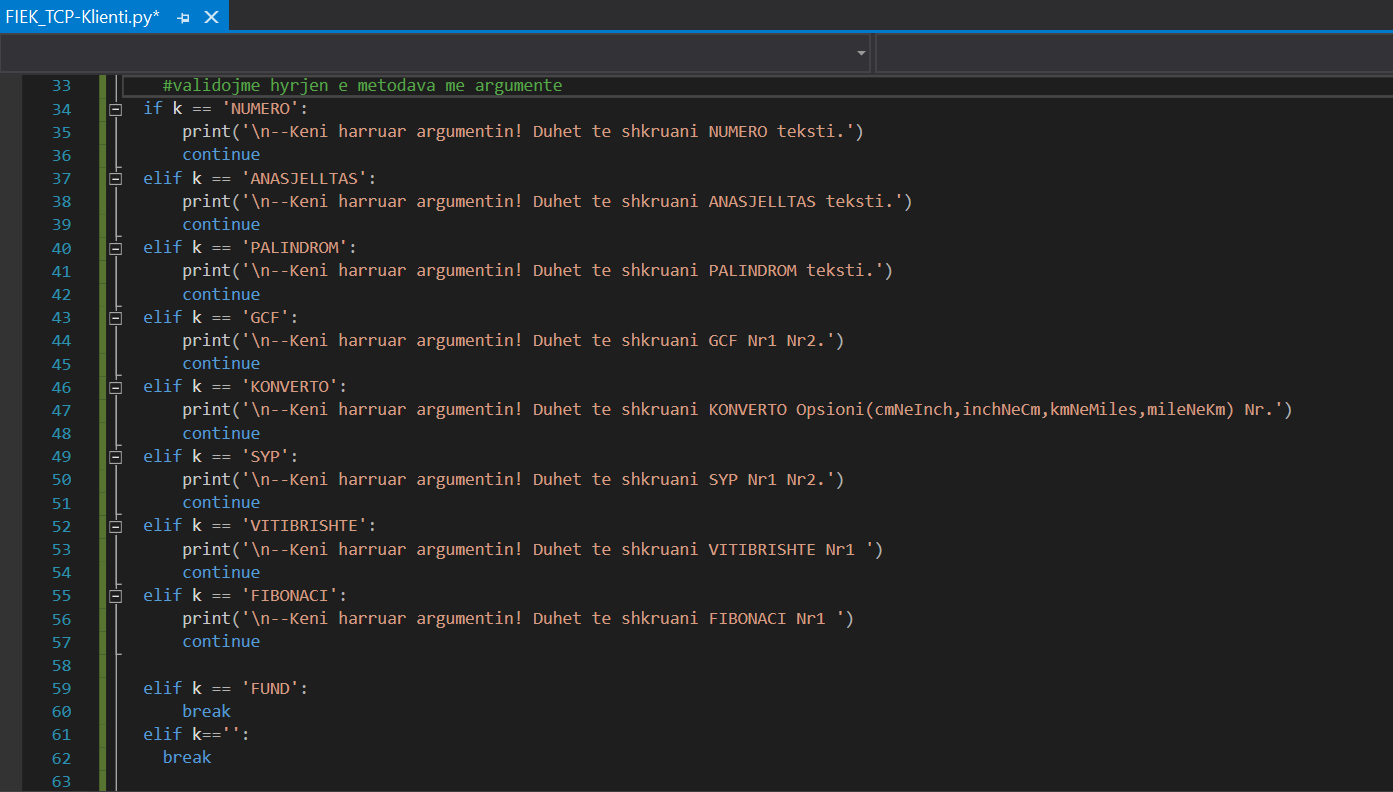
Klientin e fillojmë me kërkesen e detyrës e cila ka qenë për të mundësuar ndërrimin e portit dhe emrit të serverit sipas dëshirës së përdoruesit, ose edhe mund t’ i lihet default porti 14000.

****

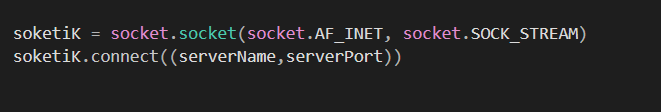
Te kjo pjesë krijojmë mundësinë që hyrja(inputi) të kthehet në shkronja të mëdha dhe që gjatësia të jetë më e vogël se 128 karaktere.

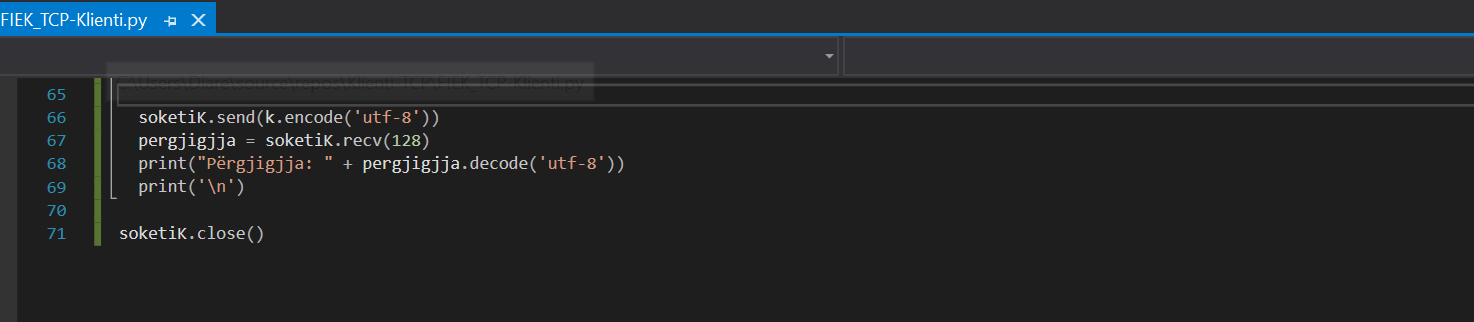


Te pjesa në vazhdim bëjmë njëfarë validimi që mos të lejojmë futjen e metodave që kanë parametra të cilat në input janë dhënë pa parametra nga ana e user-it.Poashtu japim mundësinë me përfundu programi me anë të komandës NDAL, edhe nëse nga ana e user-it lihet kërkesa e zbrazët dhe preket tasti enter poashtu përfundon programi.



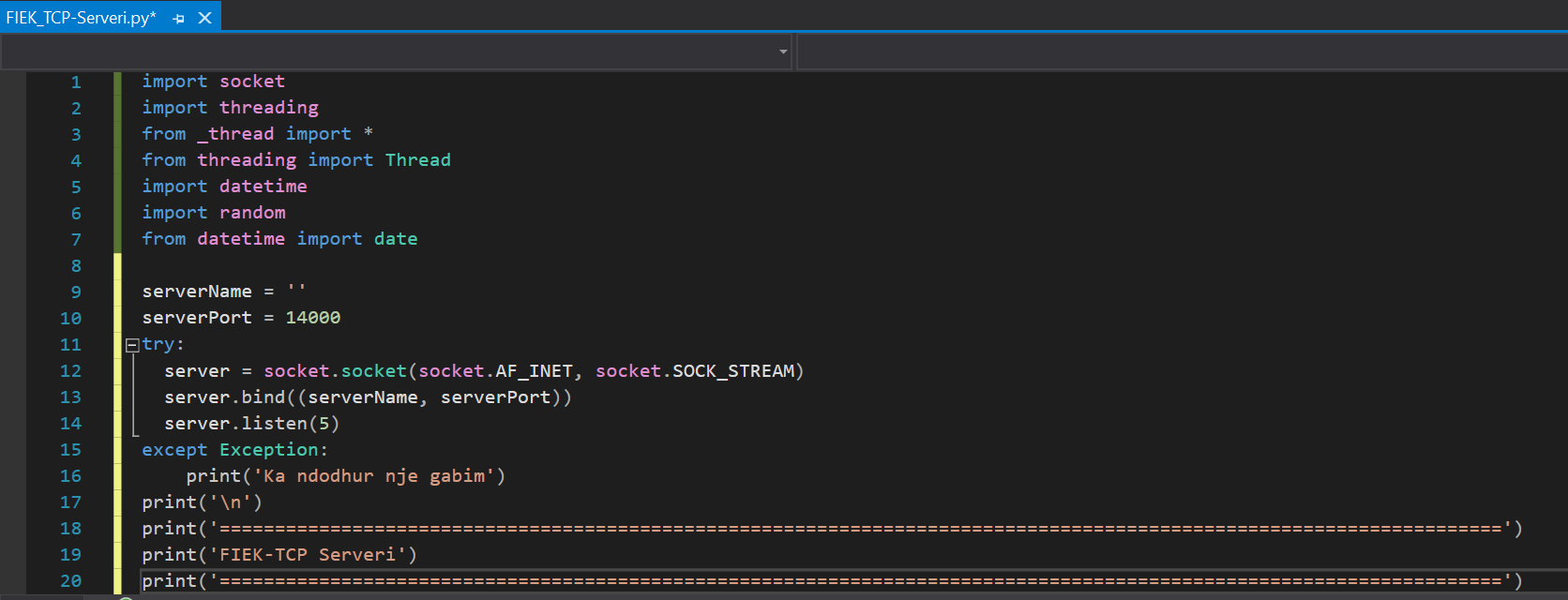
Në këtë pjesë e krijomë soketin të cilin e kemi emëruar soketiK.Që te krijojme nje objekt *socket* duhet perdorur socket.socket() dhe specifikojme llojin e soketeve si socket.SOCK\_STREAM e cila përdoret për të krijuar një socket për TCP, ndërsa AF\_INET paraqet - IPv4. Më pas këtë soket e kemi lidhur me serverin.



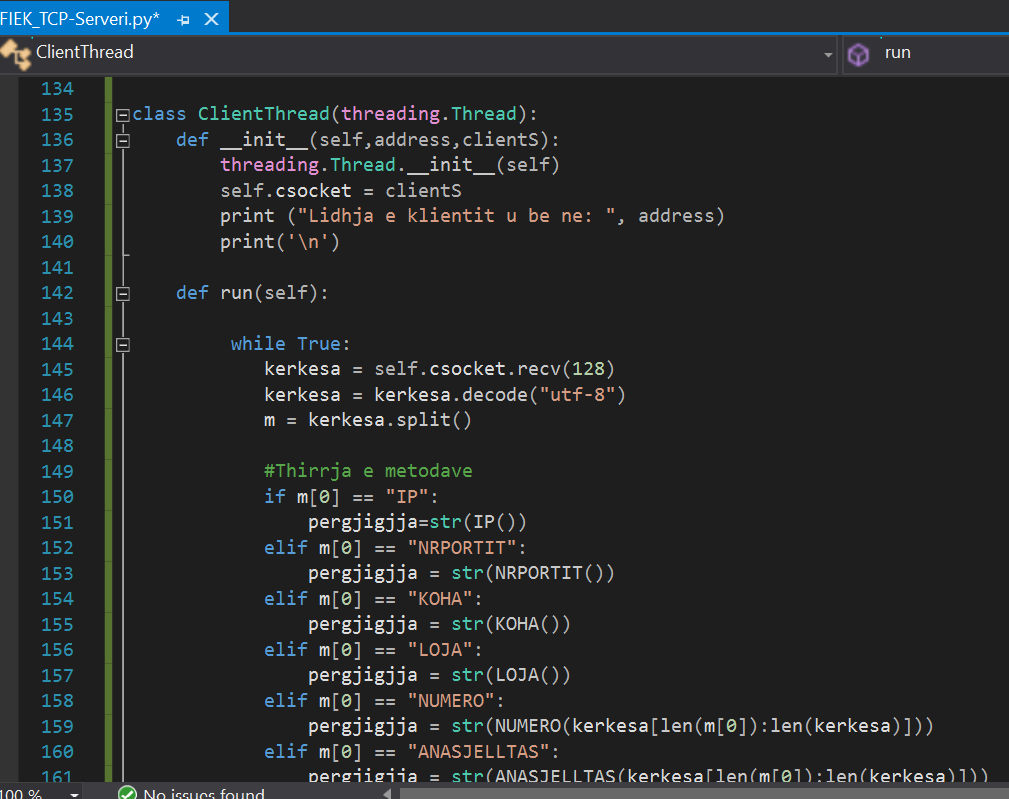
Në kodin më poshtë mundësohet që kërkesa që në rastin tonë është input t’i dërgohet serverit me anë të metodës send() të cilën e enkodon me (utf-8).Pas kësaj me anë të metodës recv(128) pranohet përgjigjja nga serveri ku edhe dekodohet me formatin e njejtë sikurse që enkodohet.Krejt në fund mbyllet soketi me anë të metodës close().

# Fiek\_TCP\_Serveri

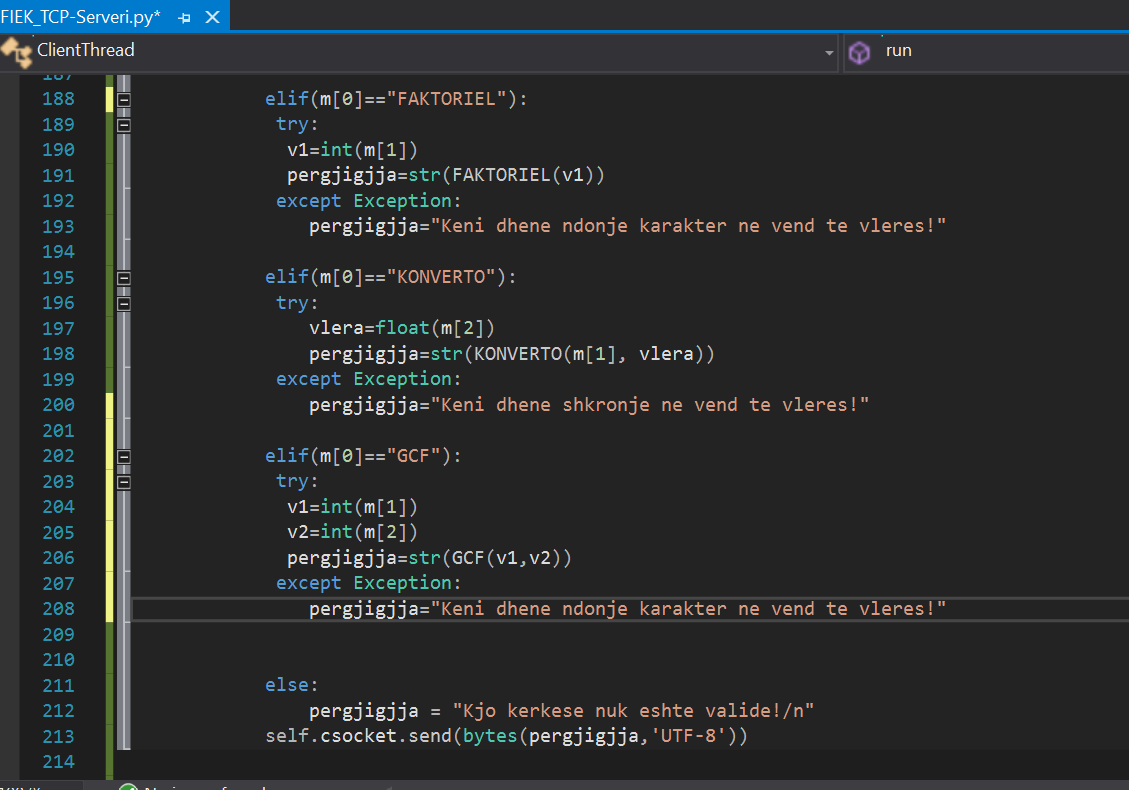
Fillimisht i kemi importuar të gjitha libraritë që na duhen për krijimin e serverit e po ashtu edhe për metodat që i përmbanë ky server.Më pas caktojmë emrin dhe portin e serverit, dhe krijojmë soketin e serverit. Bind mundëson që të lidhet në një port dhe IP adresë të caktuar, listen() i dëgjon kërkesat e klientit,që në rastin tonë maksimumi i kemi 5 klienta njëkohësisht.

****

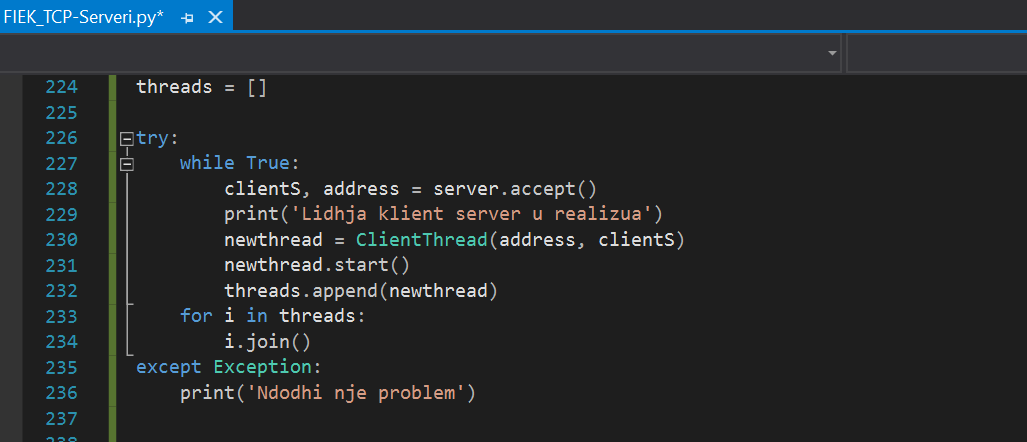
Në server pjesa e thredave është punuar me anë të një klase.Në server po ashtu janë implementuar të gjitha metodat e kerkuara e poashtu edhe ato shtesë.Poashtu shihet sesi i dërgohet kërkesa dhe si mundësohet kthimi i përgjigjes.Kemi përdorur metoden split()-që na munëson që t’i trajtojmë si një varg emrin dhe argumentet e metodës përkatëse.



Validimi i metodave me disa argumenta siq është GCF, KONVERTIMI, FAKTORIEL etj. Po ashtu shihet se nëse shkon ndonjë kërkesë qe nuk ekziston në server na tregon që nuk është kërkes valide.



Në këtë pjesë shihet krijimi i një threadi të ri. Tani duke qenë një server, socket duhet që të pranojë disa lidhje klientësh.Kjo bëhet duke thirrur accept() brenda një loop-e të vazhdueshme. Metoda accept() thirret një herë për secilën lidhje hyrëse dhe kthen një çift që përmban një socket që lidhet me klientin dhe IP adresën e klientit.



# përshkrimi i metodave

1.IP adresa- kjo metodë përdoret për të kthyer ip adresen e klientit.Marrjen e kësaj adrese e bëjmë me anë të getbyhostname(), ose mund të merret edhe me marrjen e anëtarit të parë të soketit.

2.porti i klientit –kjo metodë shërben për të marrë portin e klientit që është i lidhur në soket, dhe këtë ja kemi mundësuar me marrjen e anëtarit të dytë të soketit

3.numero-kjo metodë mundëson numërimin e shkronjave zanore dhe bashkëtingëllore.Fillimisht i inicializojmë dy variabla të barabarta me zero e më pas formojmë një unazë ku duhet të shkojë te secila shkronjë e tekstit të dhënë, me kushtëzime i rrisim njërën variabël ku kemi zanore ndërsa tjetrën ku kemi bashkëtingëllore.

4.Anasjelltas-kjo metodë e kthen vargun e anasjelltë(reversë).Pythoni na mundëson që kjo të zhvillohet me anë të kësaj shprehje [::-1].

5.palindrom-kjo metodë ka për qëllim vërtetimin e fjalëve që lexohen njejtë si nga e majta si nga e djathta.Fillimisht e marrim një variabël në të cilën ruajmë të anasjellten e tektit të dhënë e më pas i krahasojmë mes veti, nëse na del që është e njejtë i bie se është palindrom përndryshe jo.

6.koha-kjo metodë shërben për të kthyer kohën aktuale.Kjo metodë përdor librarinë datetime dhe metoda built-in të klasës datetime siq janë now() dhe strftime().Me anë të strftime përcaktojmë formatin e datës.

7.loja-është metodë e cila na kthen 5 numra të rastësishëm nga 1 deri në 35 dhe rezultati i këtyre numrave është i sortuar.Në këtë metodë fillimisht kemi caktuar një set bosh e më pas me një for loop kemi iteruar për 5 numra qe i marrim në mënyrë të rastësishme me anë të metodës random, më pas i shtojmë në varg dhe i sortojmë.

8.GCF-është metodë që kthen faktorin më të madh të përbashkët në mes dy numrave integer.Të dy numrat i merr si argumente dhe fillimisht i krahason dhe e gjenë se cili numër është më i vogël e më pas me një for loop shkon prej 1 deri te numri më i vogël plus një dhe vazhdimisht i shikon se a po plotëpjestohen numrat fillestar me ta derisa të përfundojë loop-a.

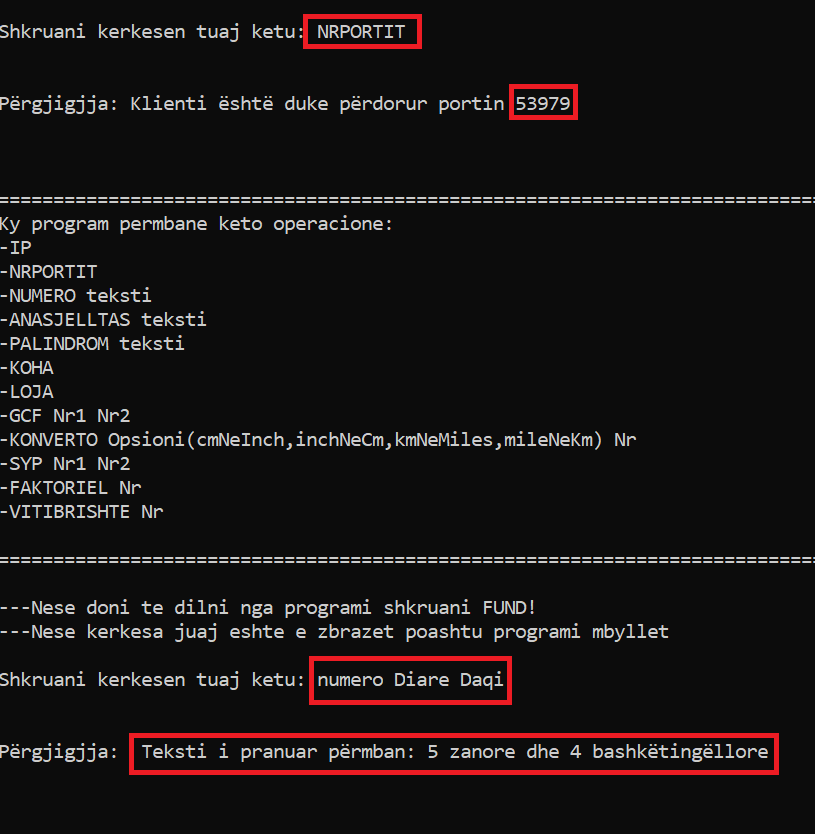
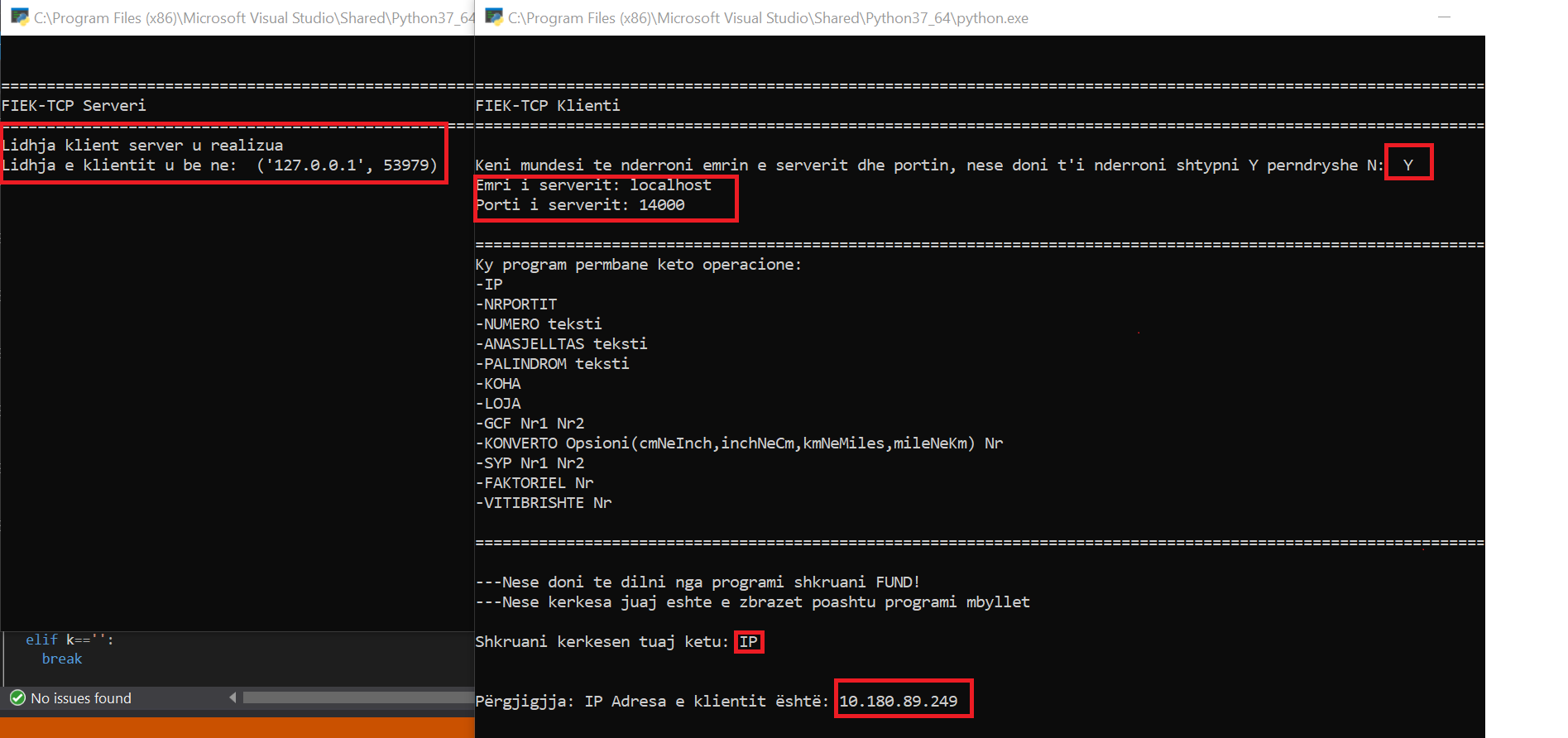
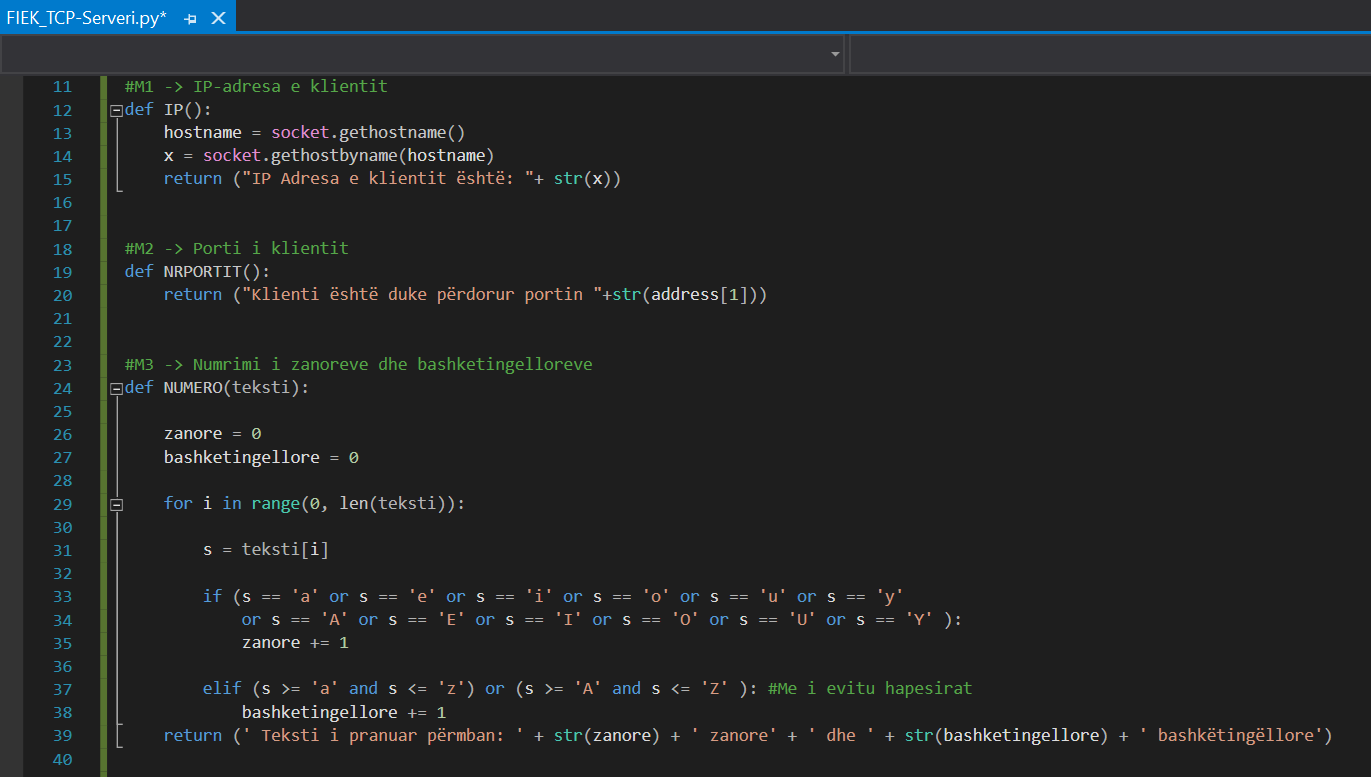
9.konverto-është metodë që shërben për të kaluar nga një njësi në tjetrën, i merr dy parametra ku njëri tregon opsionin tjetri vlerën.Kjo metodë u zhvillua me anë të kushtëzimeve të opsionit, pra varësisht nga opsioni shumëzohet me vlerën përkatëse.

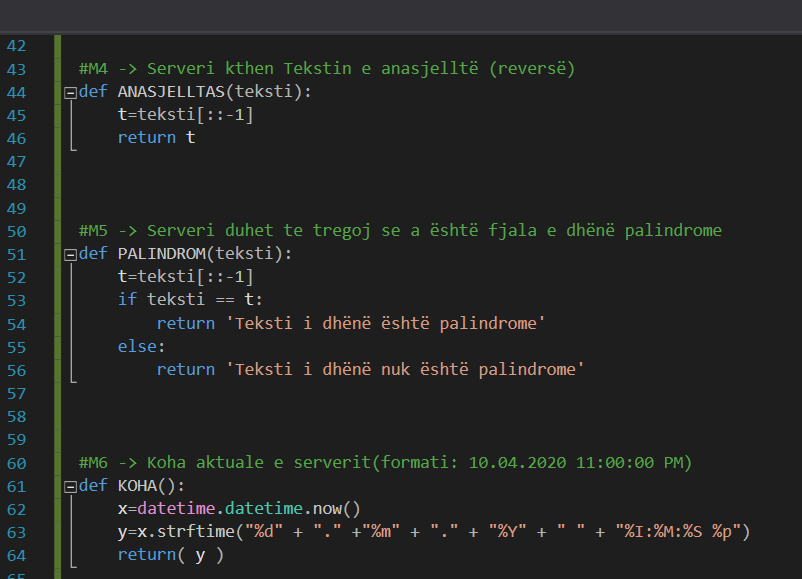
Metodat shtese

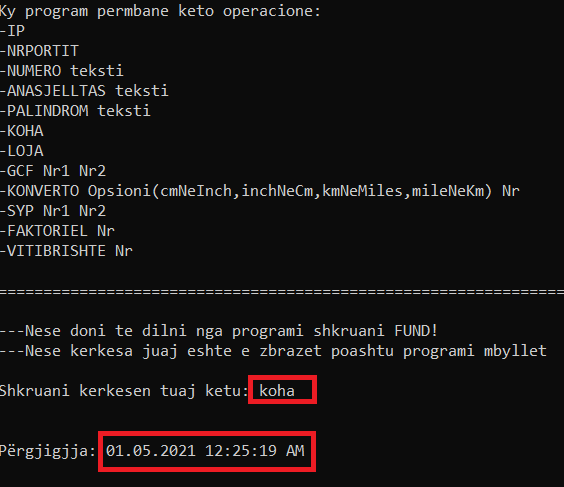
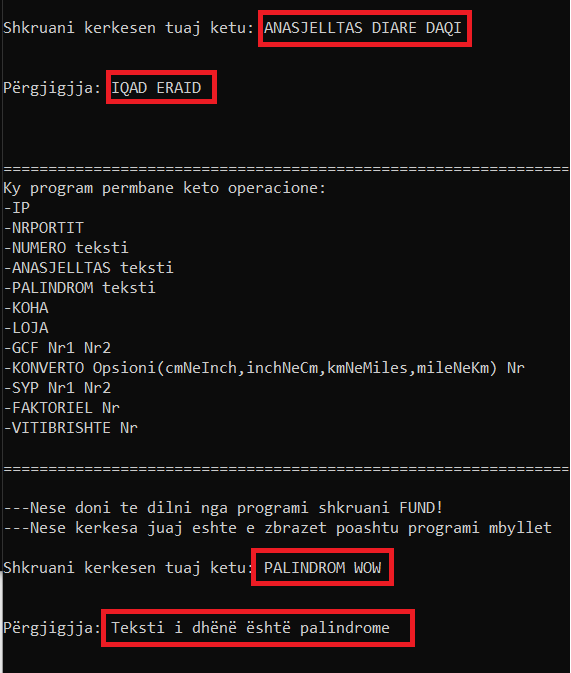
1.syp-është metodë që llogarit syprinën e sipërfaqes së drejtëkëndëshit duke i pranuar brinjët e tij si parametra.

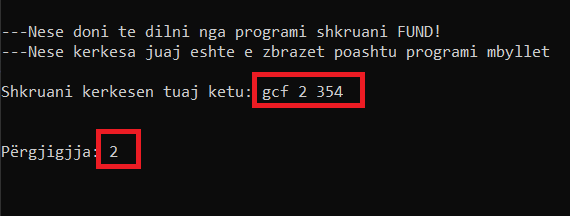
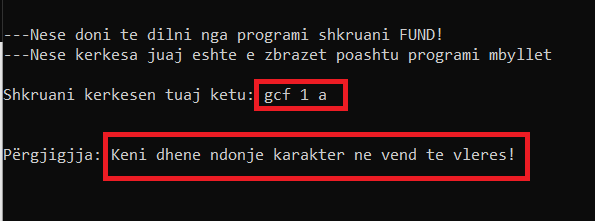
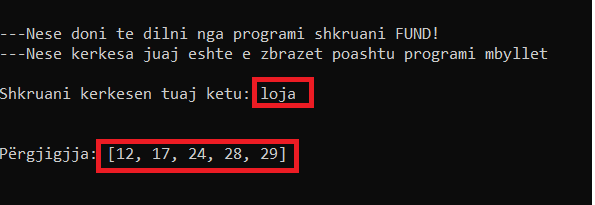
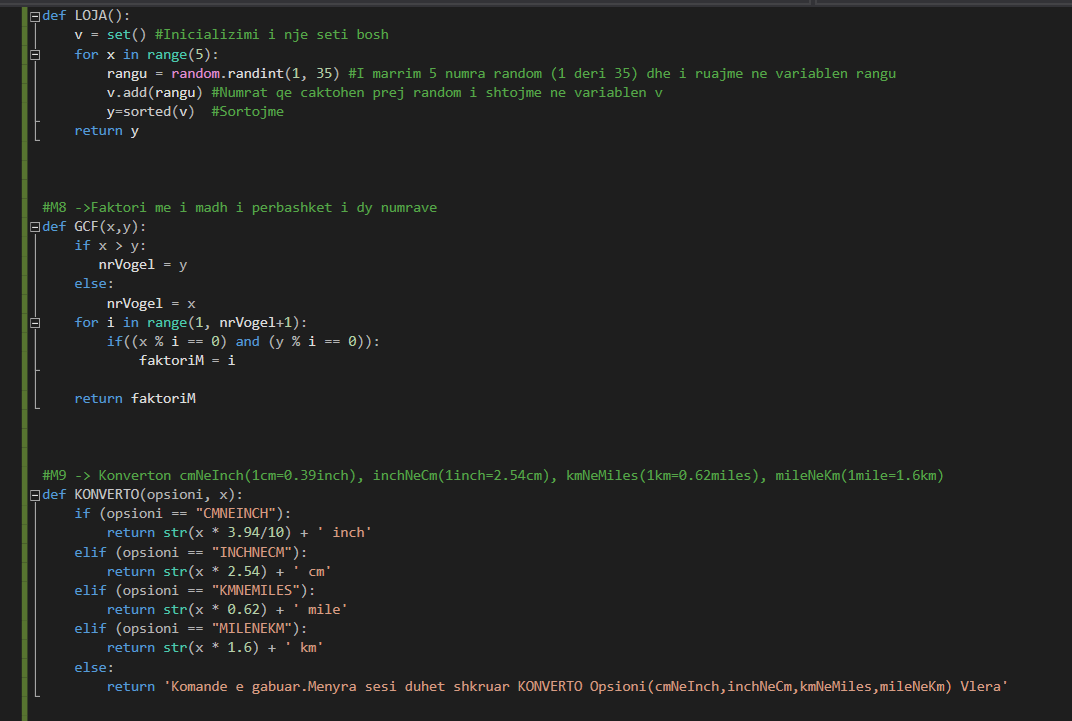
2.FAktorieli-është metodë që merr një argument dhe nëse ai argument është më i madh se një shkon duke gjetur prodhimin e atij numri duke e zbritur për një derisa të bëhet një.Pra kjo metodë bëhet me anë të kushtëzimeve, nëse numri është 1 atëherë e kthen vetë atë numër, nëse është më i vogël se një na kthen mesazhin numër negativ dhe nëse nuk vlejnë rastet e lartëpërmendura shkon dhe bëhet funksion rekurziv dhe vazhdon procedura e njejtë derisa numri të bëhet 1.

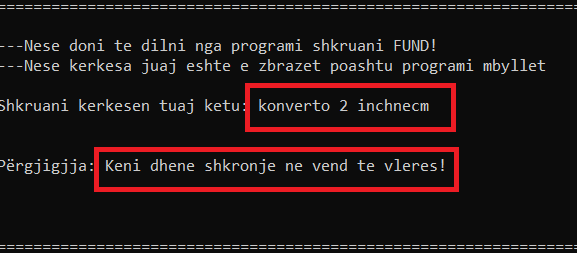
3.vitibrishte-është metodë që çfarëdo numri që ne i dërgojmë si parametër do të na kthejë nëse ajo vlerë do ishte vit i brishtë ose jo.Kjo metodë realizohet përmes kushtëzimeve duke shikuar plotëpjestueshmërinë.

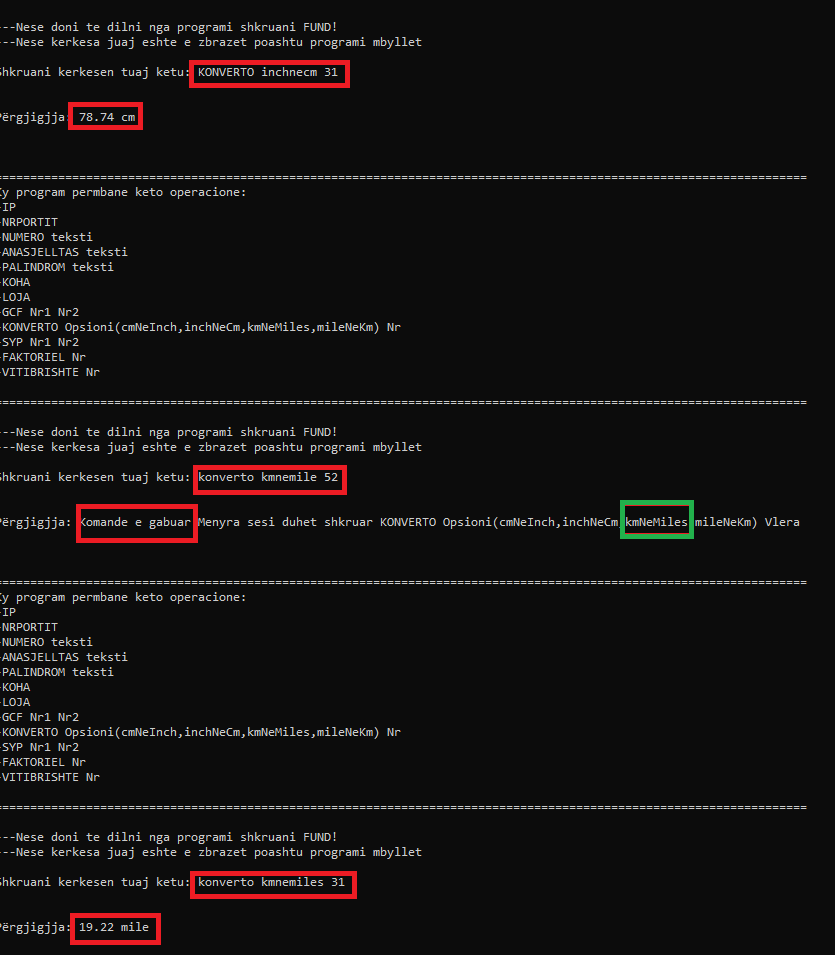
Testimi i metodave 



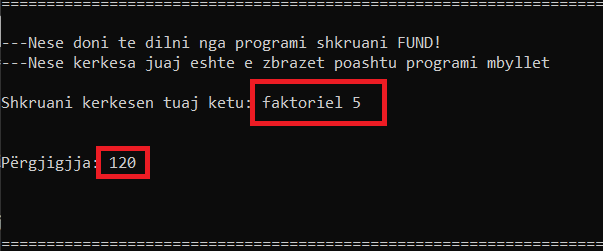
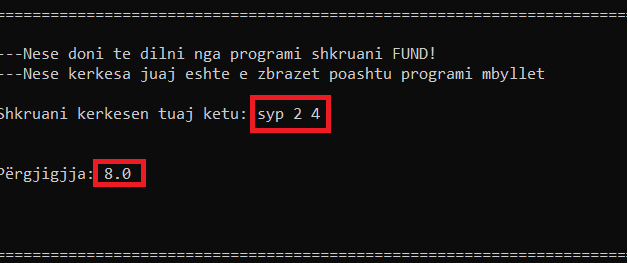
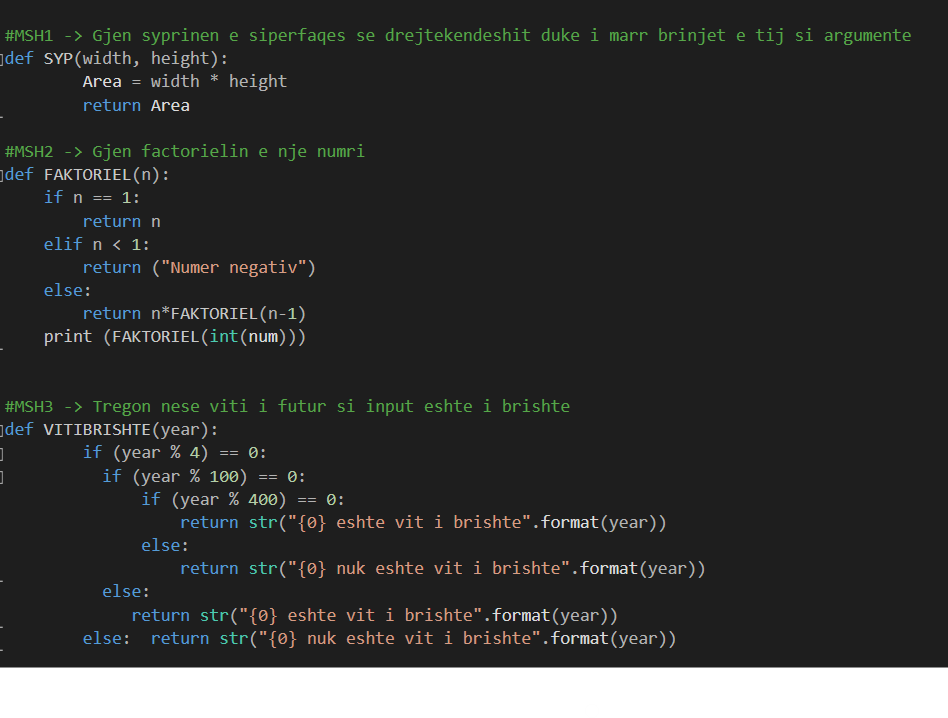


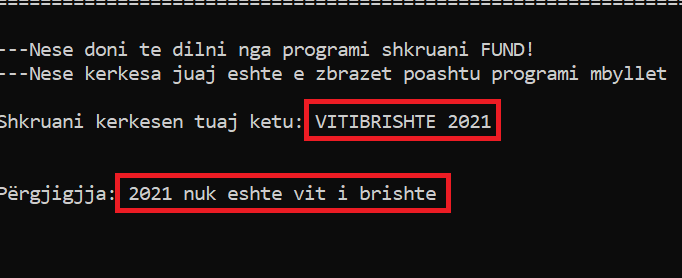






# Metodat shtesë

****

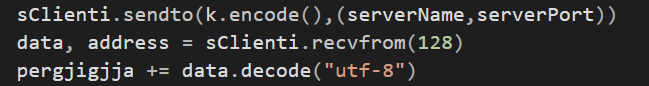
****

# Fiek-UDP-Protocol

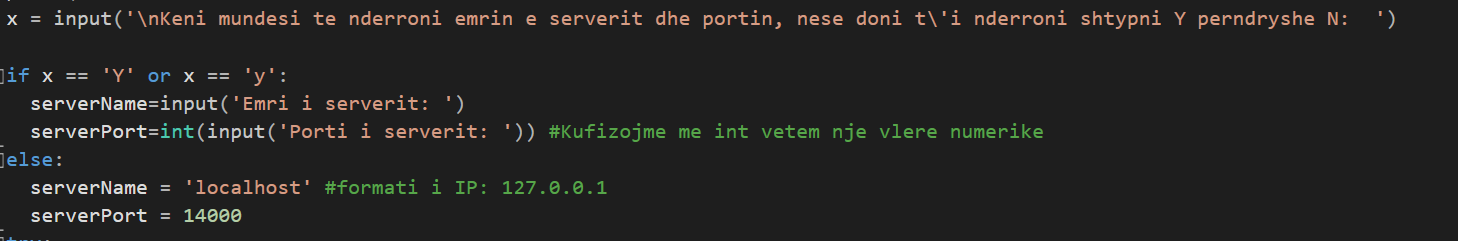
Ne mund të krijojmë klient/server të UDP të ngjashëm me klientin/serverin në TCP.Dallimi i vetëm është në specifikimin e SOCK\_DGRAM në vend të SOCK\_STREAM kur krijojmë objektin socket.Komunikimi ndërmjet këtyre soketave mund të realizohet përmes Command Line, në rastin kur përdorim protokollin UDP nuk kemi lidhje permanente me serverin.

# Fiek\_uDP\_Klienti

Fiek\_UDP\_Klienti nuk formon lidhje me serverin,por ai thjeshtë i dërgon atij një datagram.

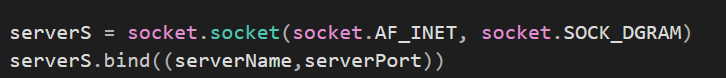


Po ashtu edhe këtu mundësojmë ndërrimin e portit ose IP adresës sipas preferencës së klientit.



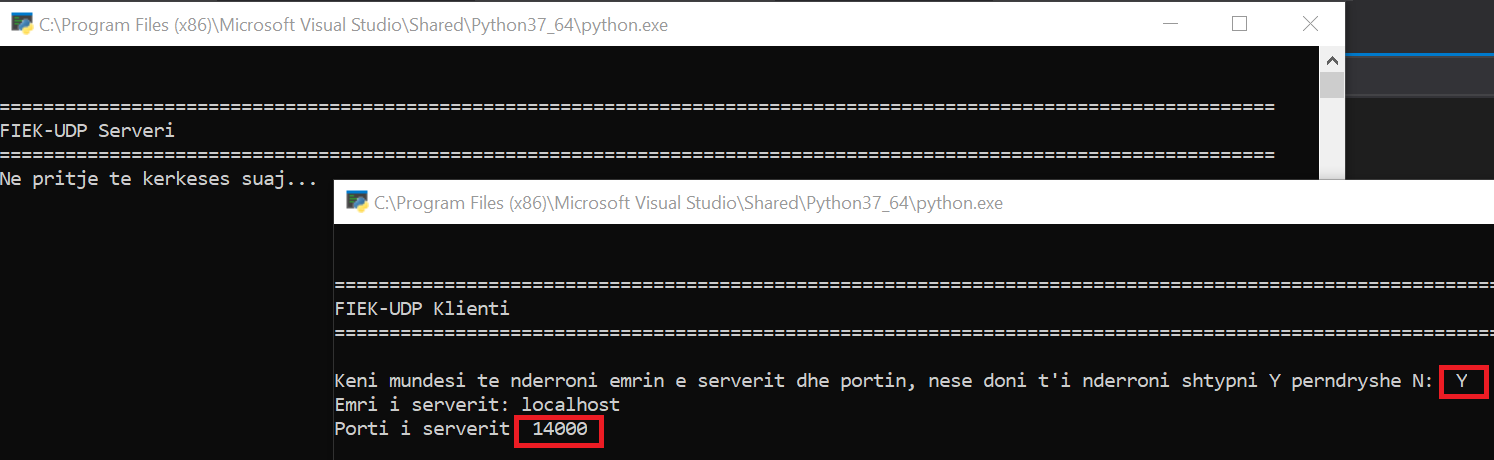
# Fiek\_uDP\_serveri

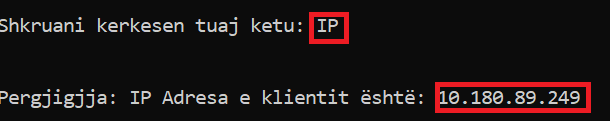
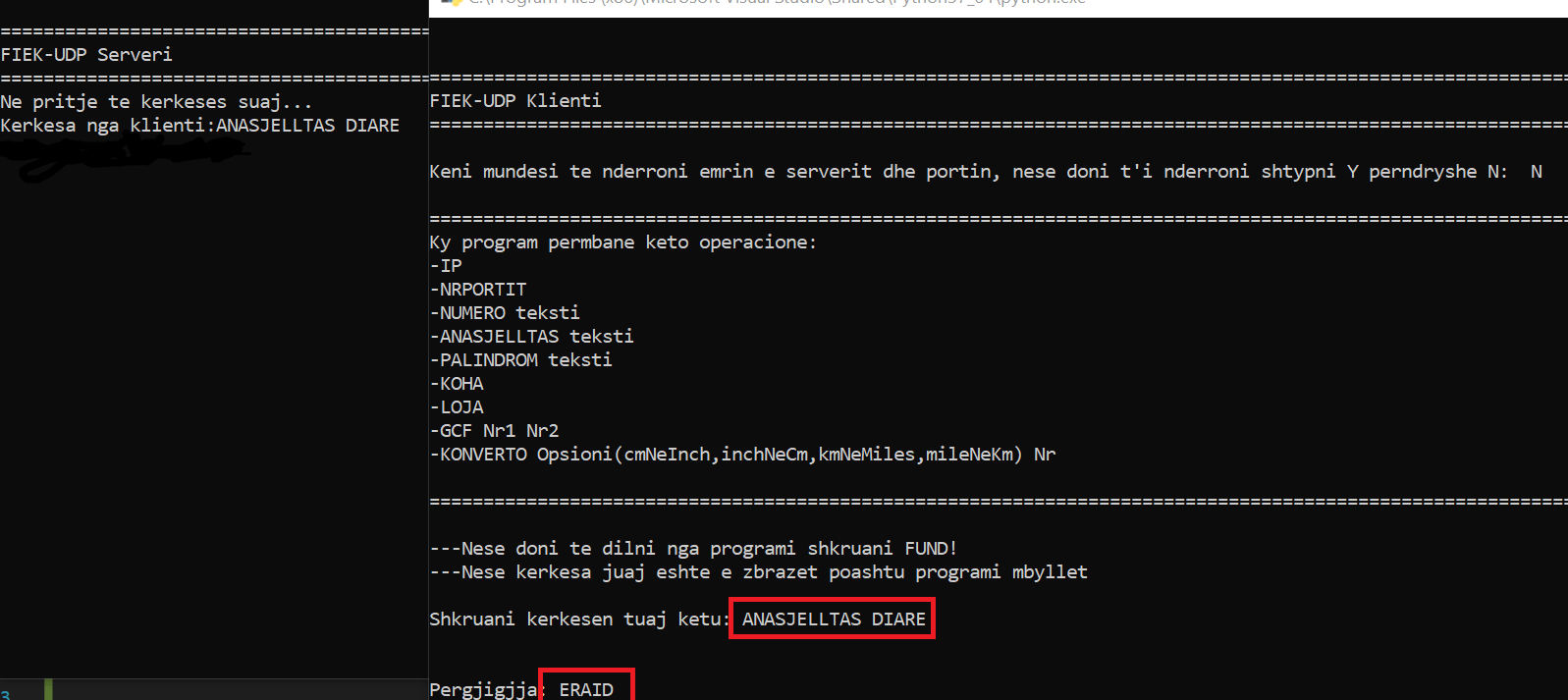
Serveri UDP nuk pret sikurse serveri TCP të krijohet lidhja por vetëm pret që t’i vijë datagrami në të cilën gjendet edhe adresa e dërguesit dhe serveri në bazë të kësaj di ta kthejë përgjigjen në dërguesin e kërkesës.



Të gjitha metodat që ndodhen tek TCP serveri janë poashtu të implementuara edhe në UDP server, ato funksionojnë ngjashëm edhe në këtë server.

Testimi i disa metodave te Udp serveri

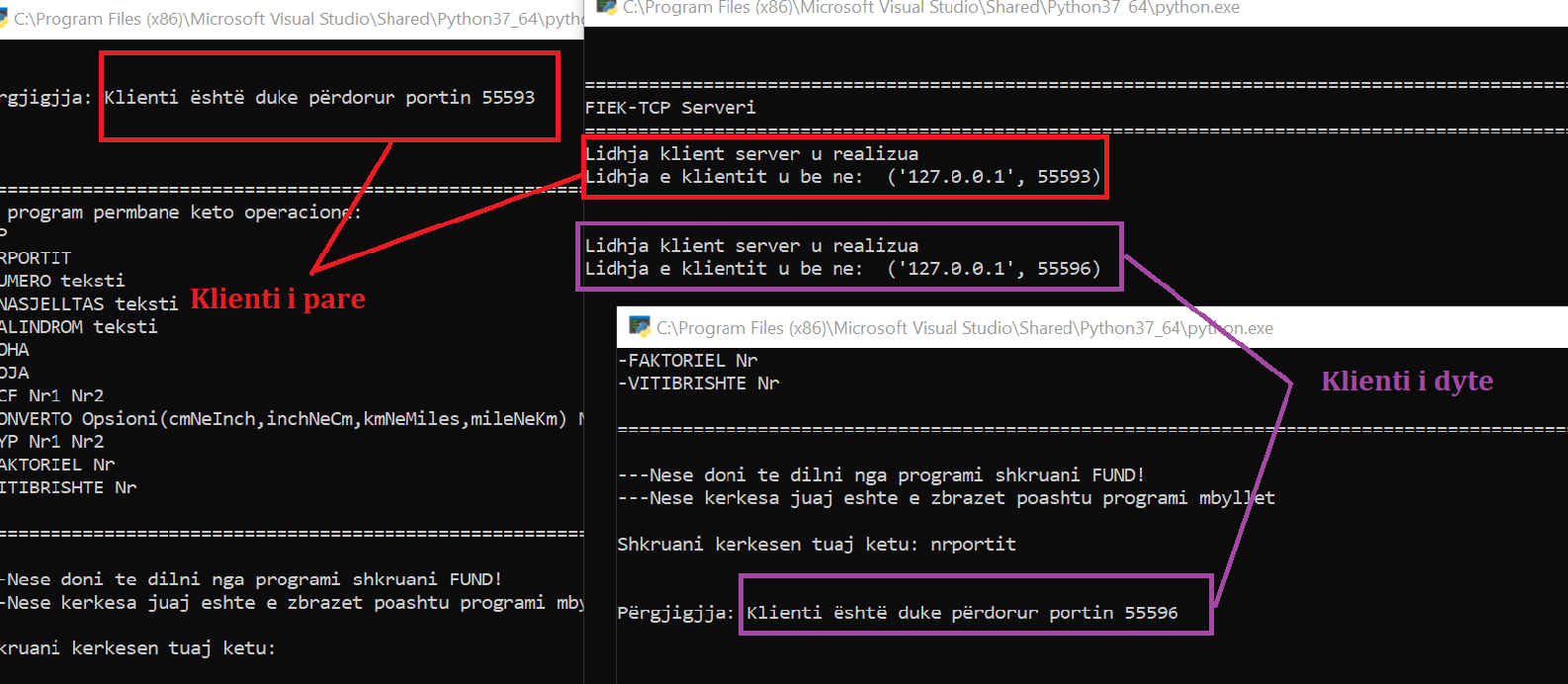




KONKLUZIONI

Siq edhe është kërkuar në projekt është bërë implementimi i të gjitha kërkesave dhe të gjitha kthejnë rezultat sikurse që kërkohet.Kur serveri pranon kërkesa jo valide jep mesazh që na bënë me dije se ajo kërkesë është jovalide.Gjithsej janë shtuar tri metoda shtesë, ku janë diskutuar tek pjesa e përshkrimit të metodave.

U mundësua që serveri TCP të jetë multithreading që të pranojë kërkesa nga disa klient njëkohësisht.



Problemi i vetëm më është paraqitur kur e mbylli klientin në TCP që më kthen një error në server.Përndryshe tjerat janë në rregull.

referencat

<https://www.w3schools.com/python/default.asp>

<https://www.positronx.io/create-socket-server-with-multiple-clients-in-python/>

<https://stackoverflow.com/questions/5520821/python-multithreaded-server>

https://www.geeksforgeeks.org/socket-programming-multi-threading-python/

Libri : Computer Networking A Top-Down Approach 7th edition.pdf