

UNIVERSITETI I PRISHTINËS
Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike



RAPORT

Lënda: Rrjetat Kompjuterike

Projekti 1: Dizajnimi Klient-Server

Studenti: Arbena Musa
Nr. i ID: 170714100093

Prishtinë, Prill 2019

Emri i projektit: Projekti 1-Dizajnimi Klient-Server

Emri i studentit: Arbena Musa

Vegla e përdorur: Visual Studio - Python

```
3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
```

Sistemi operativ: Windows 10 Pro, Version: 1803

Lista e metodave të implementuara: IPADRESA,

NUMRIIPORTIT,

BASHKETINGELLORE,

PRINTIMI,

EMRIIKOMPJUTERIT,

KOHA,

LOJA,

FIBONACCI,

KONVERTIMI.

Përmbajtja

Hyrje

Përshkrimi i projektit

Programimi Klient-Server

TCP protocol

UDP protocol

Përshkrimi i metodave

Metoda IPADRESA

Metoda NUMRIIPORTIT

Metoda BASHKETINGELLORE

Metoda PRINTIMI

Metoda EMRIIKLIENTIT

Metoda KOHA

Metoda BASHKETINGELLORE

Metoda FIBONACCI

Metoda KONVERTIMI

Metoda VERSIONIIPYTHON

Metoda VERSIONIIOS

Testimi

Testimi TCP protocol

Testimi *UDP protocol*

Përmbledhje e rezultateve të testimit

Hyrje

Përshkrimi i projektit

Ky projekt përshkruan dizajnimin, implementimin dhe testimin e komunikimit klient-server përmes socket programming në gjuhën programuese Python. Protokolli i përdorur do të jetë FIEK protokolli i cili u mundëson serverëve dhe klientëve komunikimin përmes dy versioneve të tij, si TCP-protocol dhe si UDP-protocol. Serveri përmban 11 kërkesa (metoda) të cilave mund t'iu referohet klienti. Për secilën kërkesë të klientit server kthen përgjigjen përkatëse mbrapsht te klienti.

Protokolli FIEK-TCP

Protokolli TCP (Transmission Control Protocol) është një nga protokollet kryesore. Gjatë komunikimit mes dy pikave bëhet kontrollimi i transportit të paketave. Çdo humbje e paketave në rrjet detektohet dhe ridërgohet nga dërguesi te marrësi. Të dhënat pranohen nga marrësi në të njejtin rend siç janë dërguar nga dërguesi.

Protokolli FIEK-UDP

Protokolli UDP (User Datagram Protocol) transmeton të dhënat si datagrams, mes dy pikave të ndryshme të cilat nuk kanë nevojë të konektohen paraprakisht. Nuk kontrollohet transporti i paketave, pra në rast të humbjes marrësit nuk i ridërgohen përsëri.

Programimi Klient-Server

Socket është një endpoint (pikë fundore) që mundëson komunikimin e dy proceseve brenda të njejtë kompjuter apo mes kompjuterëve të ndryshëm. Socket paraqet një mënyrë virtuale të komunikimit të dy kompjuterëve në rrjet. Socket hap lidhjet në rrjet që të ju mundësojë aplikacioneve të shkruajnë dhe të lexojnë nga njëri tjetri përmes rrjetës. Një Socket identifikohet në mënyrë unike nga dy komponente, IP adresa dhe numri i portit.

Klienti e ka mundësinë ta zgjedhë vetë IP adresën dhe numri e portit, apo të përdorë ato që caktohen në kod.

```
print("A doni te caktoni vete serverin dhe portin qe deshironi te perdorni?")
pergjigja=input().upper()
if(pergjigja=="PO"):
    print("Jep emrin e serverit:")
    servername=input().lower()
    print("Jep numerin e portit")
    p=input()
    port=int(p)
else:
    servername='localhost'
    port=12000
```

TCP protocol

```
serversocket=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
serversocket.bind((servername, serverport))
```

Dy funksionet që përdoren në komunikim janë .sendall() që merr si parameter vlerën që duhet të dërgohet tek pika tjetër e komunikimit dhe .recv() që ka si parameter gjatësinë në byte të vlerës që duhet të mirret nga pika tjetër.

UDP protocol

```
serversocket=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
serversocket.bind((servername, serverport))
```

Dy funksionet që përdoren në komunikim janë .sendto() që merr si parametër të parë vlerën që duhet dërguar tek pika tjetër e komunikimit , ndësi si parameter të dytë e ka adresën e pikes tjetër.Dhe .recvfrom() që ka si parameter gjatësinë në byte të vlerës që duhet të mirret nga pika tjetër.

Përshkrimi i metodave

Pas vendosjes së komunikimit mes serverit dhe klientit, klientit i mundësohet thirrja e metodave:

```
while True:
    print('Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, KONVERTIMI, VERSIONIIPYTHON, VERSIONIIOS)?')
    var=input().upper().encode()
    if var=='':
        break
```

```
=====
Ky eshte programi FIEK-TCP Client.
Klienti eshte gati per te komunikuar me serverin.
=====
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, KONVERTIMI)?
```

Metoda qe thirret nga klienti kalon ne server.

```
print('=====')
print('Ky eshte programi FIEK-TCP Server.')
print('Serveri eshte duke punuar ne portin '+str(serverport)+' Ky port mund te ndryshohet nga klienti sipas nevojës.')
serversocket.listen()
print('Serveri eshte gati per te pranuar kërkesa.')
print('=====')

while True:
    try:
        socketKlienti, addr = serversocket.accept()
        print("Klienti i lidhur ne portin "+str(addr[1]))
        start_new_thread(clientthread, (socketKlienti, ))
    except:
        print("Error 2")

serversocket.close()
```

```
def clientthread(socketKlienti):
    try:
        while True:
            opsioni = socketKlienti.recv(1024).decode()
            kërkesat(opsioni)
            socketKlienti.close()
    except:
        print("Error")
```

Metoda që e zgjedh klienti merret si input, enkodohet dhe dërgohet te server. Nga ana tjetër server është i gatshëm për të pranuar kërkesa. Kërkesa që e pranon serveri dekodohet dhe kalon si parametë i metodës kërkesa().

Metoda kërkesa() merr si parameter metodën e kërkuar nga klienti.

```

def kerkesat(op):
    opi=op
    op = op.split()
    if(op[0]=="IPADRESA"):
        IPADRESA()
    elif(op[0]=="NUMRIIPORTIT"):
        NUMRIIPORTIT()
    elif(op[0]=="BASHKETINGELLORE"):
        BASHKETINGELLORE(opi[17:])
    elif(op[0]=="PRINTIMI"):
        PRINTIMI(opi[9:])
    elif(op[0]=="EMRIIKOMPJUTERIT"):
        EMRIIKOMPJUTERIT()
    elif(op[0]=="KOHA"):
        KOHA()
    elif(op[0]=="LOJA"):
        LOJA()
    elif(op[0]=="FIBONACCI"):
        FIBONACCI(op[1])
    elif(op[0]=="KONVERTIMI"):
        KONVERTIMI(op[1],op[2])
    elif(op[0]=="VERSIONIIPYTHON"):
        VERSIONIIPYTHON()
    elif(op[0]=="VERSIONIIOS"):
        VERSIONIIOS()
    else:
        NOFUN()

```

Varësisht nga zgjedhja e klientit thirret metoda përkatëse. Metoda kërkesa e merr tërë zgjedhjen e klientit dhe e ndan përmes split(). Argumenti i parë i fituar është gjithmonë emri i metodës. Kur metoda jepet së bashku me parametër atëherë argumenti i dytë (ose i tretë) është parametri i metodës që thirret.

Metoda IPADRESA

```

def IPADRESA():
    z="IP Adresa e klientit eshte: "+ addr[0]
    socketKlienti.sendall(str.encode(str(z)))

```

Kjo metodë merr anëtarin e parë të addr, në të cilin është ruajtur IP adresa e klientit, e enkodon dhe e dërgon tek klienti.

Metoda NUMRIIPORTIT

```

def NUMRIIPORTIT():
    z="Klienti eshte duke perdorur portin "+ str(addr[1])
    socketKlienti.sendall(str.encode(str(z)))

```

Kjo metodë merr anëtarin e dytë të addr, në të cilin është ruajtur numri i portit në të cilin është konëktuar klienti,

e enkodon dhe e dërgon te klienti.

Metoda BASHKETINGELLORE

```

def BASHKETINGELLORE(a):
    nrbashketingellore = 0
    bashk=["b","c","d","f","g","h","j","k","l","m","n","p","q","r","s","t","v","x","z"]
    z=a.lower()
    for x in z:
        if x in bashk:
            nrbashketingellore += 1
    a=('Teksti i pranuar permban %d bashketingellore'%nrbashketingellore)
    socketKlienti.sendall(str.encode(a))

```

Serveri merr nga klienti tekstin si parametër nga i cili duhet llogaritur numri i bashkëtingëlloreve. Teksti i marrë dekodohet, kthehet në shkronja të vogla dhe secili karakter kontrollohet se a është shkronjë bashkëtingëllore. Numri i shkronjave bashkëtingëllore dërgohet te klienti si string i enkoduar.

Metoda PRINTIMI

```
def PRINTIMI(a):  
    socketKlienti.sendall(str.encode(a))
```

dërgon klientit.

Ky funksion merr nga klienti tekstin si parameter i cili duhet të printohet. Tekstin e marrë e enkodon përsëri dhe ia

Metoda EMRIIKLIENTIT

```
def EMRIIKOMPJUTERIT():  
    z=socket.getfqdn(servername)  
    socketKlienti.sendall(str.encode(str(z)))
```

Me anë të kësaj metode dërgohet tek klienti emrin e makinës që përdorë klienti. Ky emër mirret nga .getfqdn(hostname) që jep emrin e pajisjes hostuese.

Metoda KOHA

```
import datetime
```

 Importohet libraria datetime.

```
def KOHA():  
    y = datetime.datetime.now()  
    z=(y.strftime("%d")+'. '+y.strftime("%m")+'. '+y.strftime("%y")+ ' '+y.strftime("%I")+': '+y.strftime("%M")+': '+y.strftime("%S")+ ' '+y.strftime("%p"))  
    socketKlienti.sendall(str.encode(z))
```

Kjo metodë merr të dhënat aktuale të kohës. Pas përpunimit në string, dërgohen tek klienti.

Metoda LOJA

```
import random  
from random import choice
```

 Importohet libraria random.

```
def LOJA():  
    z='(  
    sequence = [i for i in range(50)]  
    for _ in range(7):  
        selection = choice(sequence)  
        z=z+str(selection)+' '  
    z=z+')jane 7 numra te rastesishem nga 49.'  
    socketKlienti.sendall(str.encode(z))
```

Kjo metodë ruan në varg numrat nga 1 deri në 50 pa përfshier 50. Zgjedhen random 7 numra prej vargut dhe i dërgohen klientit.

Metoda FIBONACCI

```
def FIBONACCI(n):  
    a,b = 0,1  
    for i in range(n):  
        a,b = b, a+b  
    socketKlienti.sendall(str.encode(str(a)))
```

Kjo metodë merr si parameter indeksin e numrit të fibonaccit dhe e logaritë numrin përkatës dhe ia dërgon klientit.

Metoda KONVERTIMI

```
def KONVERTIMI(a,b):  
    if(a=="KILOWATTOHORSEPOWER"):  
        z=int(b)*1.34102209  
    elif(a=="HORSEPOWERTOKILOWATT"):  
        z=int(b)/1.34102209  
    elif(a=="DEGREESTORADIANS"):  
        z=int(b)*0.0174532925  
    elif(a=="RADIANSTODEGREES"):  
        z=int(b)/0.0174532925  
    elif(a=="GALLONSTOLITERS"):  
        z=int(b)*3.78541178  
    elif(a=="LITERSTOGALLONS"):  
        z=int(b)/3.78541178  
    else:  
        z="Nuk eshte shkruar mire kerkesa per konvertim."  
    socketKlienti.sendall(str.encode(str(z)))
```

Kjo metodë merr dy parametra, i pari është lloji i konvertimit që duhet të kryhet, ndërsa parametri i dyte është vlera që duhet të konvertohet. Kthen tek klienti vlerën e caktuar.

Metoda VERSIONIIPYTHON

```
def VERSIONIIPYTHON():  
    z=sys.version  
    socketKlienti.sendall(str.encode(z))
```

Kjo metodë kthen te klienti versionin e Pythonit që përdoret nga kompjuteri i cili përdoret.

Metoda VERSIONIIOS

```
def VERSIONIIOS():  
    z=platform.machine()+" "+platform.platform()+" "+platform.node()+" "+platform.processor()  
    socketKlienti.sendall(str.encode(z))
```

Kjo metodë jep të dhënat e kompjuterit, os, emrin e njës(emrin e kompjuterit) dhe karakteristikat e procesorit.

Në rastet kur nga klienti thirret një metodë jo ekzistuese, injorohet kërkesa.

```
def NOFUN():  
    z=' '  
    socketKlienti.sendall(str.encode(str(z)))
```

Testimi TCP protocol

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
=====
Ky eshte programi FIEK-TCP Server.
Serveri eshte duke punuar ne portin 12000. Ky port mund te ndryshohet nga klienti sipas nevojës.
Serveri eshte gati per te pranuar kerkesa.
=====
Klienti i lidhur ne portin 56441

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
Ky eshte programi FIEK-TCP Client.
Klienti eshte gati per te komunikuar me serverin.
=====
A doni te caktoni vete serverin dhe portin qe deshironi te perdorni?
po
Dep emrin e serverit:
localhost
Dep numerin e portit
12000
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
ipadresa
'IP Adresa e klientit eshte: 127.0.0.1'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
numriiportit
'Klienti eshte duke perdorur portin 56441'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
bashketingellore rrjetat kompjuterike
'Teksti i pranuar permban 12 bashketingellore'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
printimi projekti i pare
'PROJEKTI I PARE'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
emriiklientit
'Emri i klientit eshte: DESKTOP-UA5CMPE'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
=====
Ky eshte programi FIEK-TCP Server.
Serveri eshte duke punuar ne portin 12000. Ky port mund te ndryshohet nga klienti sipas nevojës.
Serveri eshte gati per te pranuar kerkesa.
=====
Klienti i lidhur ne portin 55533

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
=====
Ky eshte programi FIEK-TCP Client.
Klienti eshte gati per te komunikuar me serverin.
=====
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
koha
'18.04.19 06:57:15 PM'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
loja
'(34 10 36 20 32 10 31 )jane 7 numra te rastesishem nga 49.'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
konvertimi literstogallons 50
'13.208602631864796'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
versioniipython
'3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
versioniios
'AMD64 Windows-10-10.0.17134-SP0 DESKTOP-UA5CMPE Intel64 Family 6 Model 142 Stepping 9, GenuineIntel'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKOMPJUTERIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
=====
Ky eshte programi FIEK-TCP Server.
Serveri eshte duke punuar ne portin 12000. Ky port mund te ndryshohet nga klienti sipas nevojës.
Serveri eshte gati per te pranuar kerkesa.
=====
Klienti i lidhur ne portin 56672

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
=====
Ky eshte programi FIEK-TCP Client.
Klienti eshte gati per te komunikuar me serverin.
=====
A doni te caktoni vete serverin dhe portin qe deshironi te perdorni?
jo
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
fibonacci 10
'55'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
fibonacci 6
'8'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
fibonacci 15
'610'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
```

Testimi UDP protocol

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
=====
Ky eshte programi FIEK-TCP Server.
Serveri eshte duke punuar ne portin 12000. Ky port mund te ndryshohet nga klienti sipas nevojës.
Serveri eshte gati per te pranuar kerkesa.
=====

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
Ky eshte programi FIEK-TCP Client.
Klienti eshte gati per te komunikuar me serverin.
=====
A doni te caktoni vete serverin dhe portin qe deshironi te perdorni?
jo
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
ipadresa
'IP Adresa e klientit eshte: 127.0.0.1'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
numriiportit
'Klienti eshte duke perdorur portin 50765'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
bashketingellore rrjetat
'Teksti i pranuar permban 5 bashketingellore'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
printimi rrjeta kompjuterike
'RRJETA KOMPJUTERIKE'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
emriiklientit
'Emri i klientit eshte: DESKTOP-UASCMPE'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
koha
'18.04.19 08:48:12 PM'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe

=====
Ky eshte programi FIEK-TCP Server.
Serveri eshte duke punuar ne portin 12000. Ky port mund te ndryshohet nga klienti sipas nevojës.
Serveri eshte gati per te pranuar kerkesa.
=====

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe

Ky eshte programi FIEK-TCP Client.
Klienti eshte gati per te komunikuar me serverin.
=====

A doni te caktoni vete serverin dhe portin qe deshironi te perdorni?
jo
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
konvertimi literstogallons 50
'13.208602631864796'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
versioniipython
'3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
versioniios
'AMD64 Windows-10-10.0.17134-SP0 DESKTOP-UA5CMPE Intel64 Family 6 Model 142 Stepping 9, GenuineIntel'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
metodhbagim
' '
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
```

```
Select C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe

=====
Ky eshte programi FIEK-TCP Server.
Serveri eshte duke punuar ne portin 12000. Ky port mund te ndryshohet nga klienti sipas nevojës.
Serveri eshte gati per te pranuar kerkesa.
=====

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe

=====
Ky eshte programi FIEK-TCP Client.
Klienti eshte gati per te komunikuar me serverin.
=====

A doni te caktoni vete serverin dhe portin qe deshironi te perdorni?
jo
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
fibonacci 10
'55'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
fibonacci 13
'233'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
fibonacci 100
'354224848179261915075'
Operacioni (IPADRESA, NUMRIIPORTIT, BASHKETINGELLORE, PRINTIMI, EMRIIKLIENTIT, KOHA, LOJA, FIBONACCI, KONVERTIMI,
VERSIONIIPYTHON, VERSIONIIOS)?
```


Përmbledhje e rezultateve të testimit

Kërkesat e projektit janë përmbushur duke filluar nga implementimi i metodave. Siç shihet testimi kryhet në mënyrën e paraparë, dmth. të gjitha metodat ekzekutohen ashtu siç duhet. Poashtu edhe dy metodat shtesë ekzekutohen pa problem. Në rastet kur klienti dërgon ndonjë kërkesë joekzistenti, ajo thjesht injorohet dhe vazhdon ekzekutimi normalisht.

Në program është përfshirë edhe error handling përmes blloqeve try dhe except, i cili mundëson që çdo gabim eventual gjatë ekzekutimit të mos e prish stabilitetin e programit.

Tek server exception handling përdoret në dy raste:

```
try:
    serversocket.listen()
    socketKlienti, addr = serversocket.accept()
    print("Klienti i lidhur ne portin "+str(addr[1]))
    start_new_thread(clientthread, (socketKlienti, ))
except:
    print("Ka ndodhur nje gabim ne krijim e socketKlientit!")
```

```
try:
    while True:
        opsioni = socketKlienti.recv(128).decode()
        kerkesat(opsioni)
        socketKlienti.close()
except:
    print("Ka ndodhur nje gabim gjate marrjes se kerkeses nga klienti!")
```

Multi-Threads

