

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Rrjetat Kompjuterike – 2017

Projekti 1:

Dizajnimi Klient – Server duke përdorur Socketët (ang. Sockets)

UDP dhe TCP

Studentja: Anitë Hajdari

Asistenti : Msc. Edon Mustafa

Data : 18/04/2017

Permbajtja

**1.Veglat e perdorura, versionet e tyre dhe lista e metodave te implementuara3**

**2.Hyrje4**

2.1 Pershkrim I shkurter rreth projektit4

2.2 Si punon arkitektura klient/server?4

2.3 Pershkrimi I klientit dhe serverit4

**3.Realizimi I TCP Klientit dhe serverit permes kodit** ………………………………………………………………………………**5**

3.1 Realizimi I TCPserver…………………..…………………………………………………………………………………………….5

3.2 Realizimi I TCPclient………………………………………………………………………………………………………………….6

**4.Realizimi I UDP Klientit dhe serverit permes kodit** …………………………………………………………………………….7

4.1 Realizimi I UDPserver…………………..…………………………………………………………………………………………..7

4.2 Realizimi I UDPclient………………………………………………………………………………………………………………..8.

**5.Pershkrimi I metodave dhe rezultatet pas tetstimit**.……………………………………………………………………….11

5.1 Pershkrimi i metodave pa hyrje11

5.1.1 Metoda IP…………………………………………………………………………………………………………………………….11

5.1.2 Metoda PORT………………………………………………………………………………………………………………………12

5.1.3 Metoda HOST………………………………………………………………………………………………………………………13

5.1.4 Metoda TIME……………………………………………………………………………………………………………………….15

5.1.5 Metoda KENO………………………………………………………………………………………………………………………16

5.2 Pershkrimi I metodave me hyrje……………………………………………………………………………………………………17

5.2.1 Metoda ZANORE…………………………………………………………………………………………………………………17

5.2.2 Metoda PRINTO………………………………………………………………………………………………………………….18

5.2.3 Metoda FAKTORIEL……………………………………………………………………………………………………………..19

5.2.4 Metoda KONVERTO…………………………………………………………………………………………………………….20

**6.Pershkrimi I metodave personale dhe rezultatet pas tetstimit**.…………………………………………………….23

6.1 Pershkrimi I metodes permes GUI23

6.1.1 Metoda COLORGAME………………………………………………………………………………………………………….23

6.2 Pershkrimi I metodave te tjera23

6.1.1 Metoda THJESHTE..…………………………………………………………………………………………………………….27

6.1.3 Metoda LARGO…..…………………………………………………………………………………………………………….28

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 2015 Ultimate
* Python 2.7

Libarite e Python:

* Socket
* Random
* Dat**e**Time
* Cprofile
* Decimal
* Tkinter

Lista e metodave te implementuara:

* IP
* PORT
* ZANORE
* PRINTO
* HOST
* TIME
* KENO
* FAKTORIEL
* KONVERTO

Lista e metodave personale te implementuara:

* COLORGAME
* THJESHTE
* LARGO

## 2.Hyrje

# 2.1 Pershkrim I shkurter rreth projektit

Projekti perbehet nga sistemi programimi I klienti dhe serverit te cilet komunikojne permes FIEK Protokollit.

Komunikimi ndermjet klientit dhe serverit behen permes socket-ave te cilat mundesojne komunikim e tyre ne portin default 9000 ose 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 kerkesen e cila percakton metoden te cilen do ta kthej server. Ajo ka dy versione:

1. FIEK-TCP
2. FIEK-UDP

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

# 2.2 Si punon arkitektura klient/server?

**Serveri** njëjtë sikurse edhe kompjuterët e tjerë është një makinë procesorike e fuqishme. Detyrat e kësaj makinë janë që të shërbejnë klientët. Serveri është console aplikacion që lidhet me klient përmes socketave 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. Pas seciles përgjigje mbyllet soketi.

# 2.3 Pershkrimi I klientit dhe serverit

FIEK-TCP:

Se pari vendoset nje lidhje(segment) ne mes te klientit dhe serverit permes socket-ave ne portin e caktuar(default 9000). Pastaj 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.

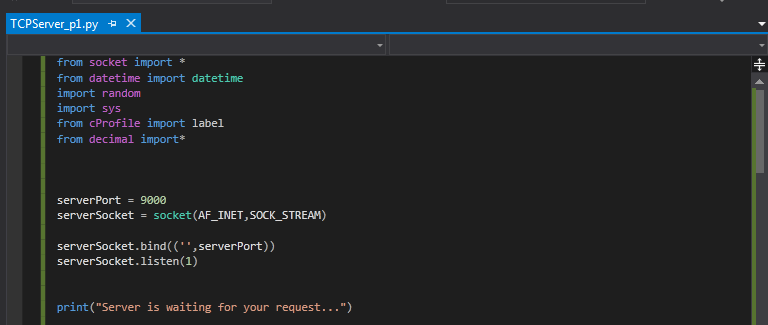
FIEK-UDP:

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 drejt. Klienti dergon kerkesen tek server permes nje UDP datagram. 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.

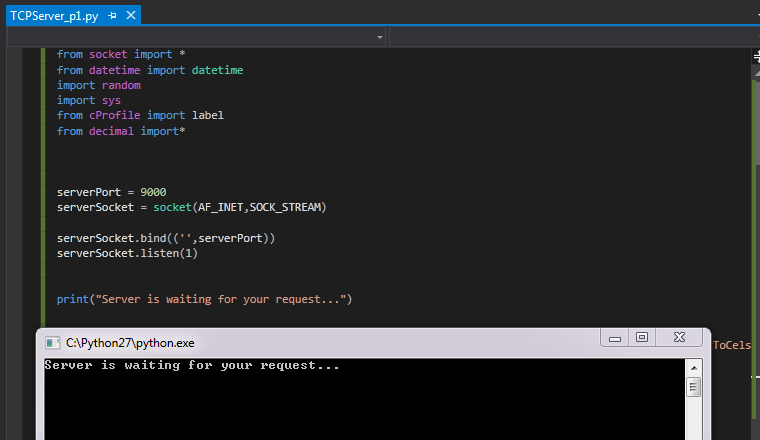
## 3. Realizimi I TCP klientit dhe serverit permes kodit

# 3.1 Realizimi I TCPserver

Se pari deklarohet variabla statike serverPort e cila perdoret gjate ekzekutimit te programit , dhe sherbejne per tu lidhur me klientin.

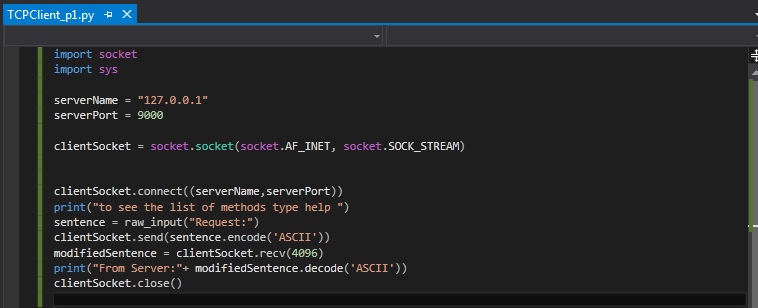


Pas ekzekutimit serveri eshte I gatshem per pranimin e kerkesave nga klienti

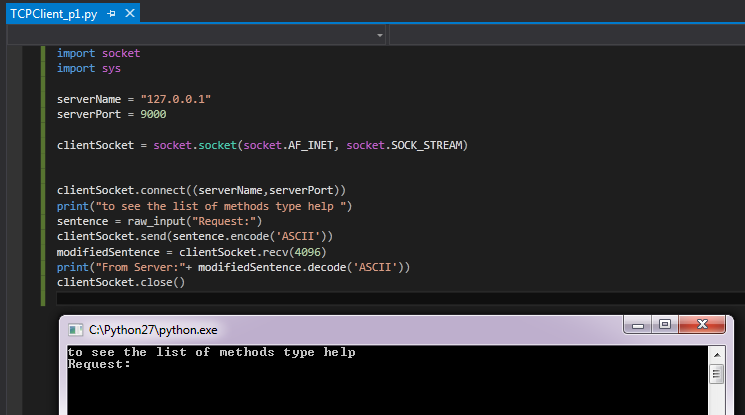


# 4.1 Realizimi I TCPclient

Ne fillim se pari deklarohen variabla serverName dhe serverPort me te cilat do te mundesojme lidhjen e klientit ne server dhe permes te ciles kryhet komunikimi.



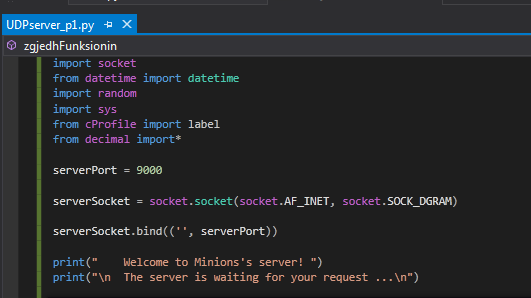
Pas ekzekutimit te TCPclient , klienti mund te beje kerkesen dhe pergjigjien e merr nga TCPserver



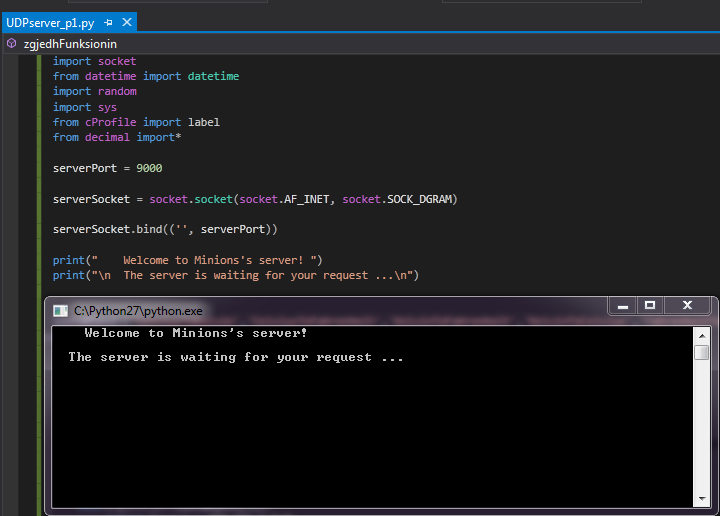
## 4.Realizimi I UDP klientit dhe serverit permes kodit

# 4.1 Realizimi I UDPserver

Se pari deklarohen variablat statike te cilat perdoren gjate ekzekutimit te programit , dhe sherbejne per tu lidhur me klientin.

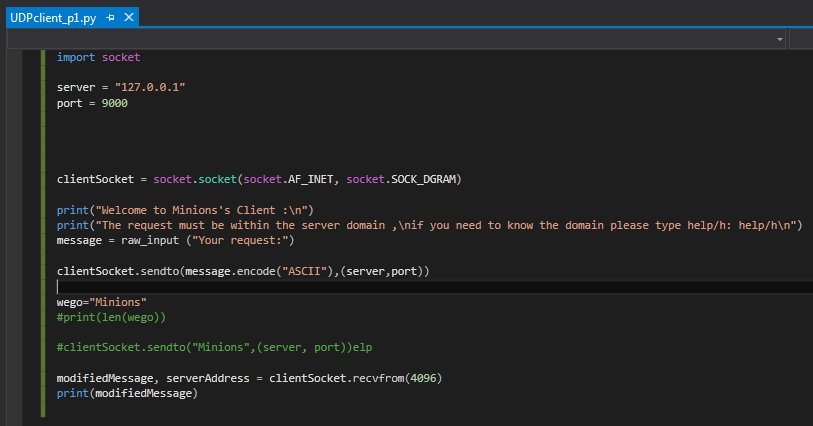


Pas ekzekutimit serveri eshte I gatshem per pranimin e kerkesave nga klienti

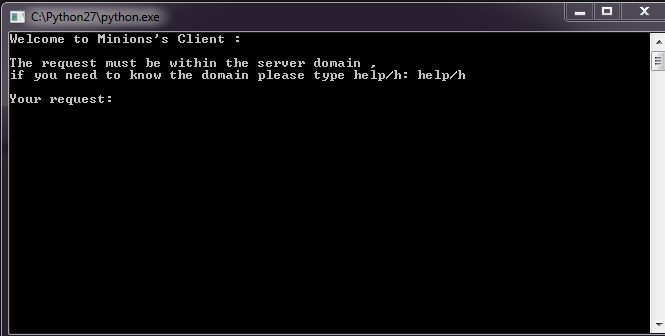


# 4.1 Realizimi I UDPclient

Ne fillim se pari deklarohen variabla server dhe port me te cilat do te mundesojme lidhjen e klientit ne server dhe permes te ciles kryhet komunikimi.



Pas ekzekutimit te UDPclient , klienti mund te beje kerkesen dhe pergjigjien e merr nga UDPserver



Per implementimin e metodave se pari deklarohet nje unaze while True: ne UDPserver, mbrenda se ciles realizohet kodi per metodat.

Fillimisht per ti mundesuar klientit njohjen e domenit eshte realizuar kerkesa HELP mbrenda unazes while True: e cila mundeson njohjen e kerkesave per te cilat UDPserver mund te kthej pergjigjie.

Realizimi permes kodit:

else: mesazhi="\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n"+\

"Lista e metodave eshte brenda domenes:\n"+\

"IP\n"+"PORT\n"+"ZANORE\n"+"PRINTO\n"+"HOST\n"+"TIME\n"+\

"KENO\n"+"Konverto\n"+"\tCelsiusToKelvin\n"+"\tCelsiusToFahrenheit\n"+\

"\tKelvinToFahrenheit\n"+"\tKelvinToCelsius\n"+"\tFahrenheitToCelsius\n"+\

"\tFahrenheitToKelvin\n"+"\tPoundToKilogram\n"+"\tKilogramToPound\n"+\

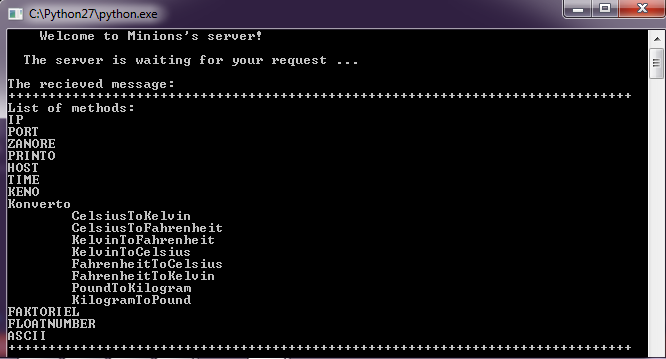
"FAKTORIEL\n"+"FLOATNUMBER\n"+"ASCII\n"+\

"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n"

print("Mesazhi i pranuar: " + mesazhi)

serverSocket.sendto(mesazhi, clientAddress)

Pas pranimit te kerkeses HELP ne server :



Pergjigjia qe kthehet nga serveri, pasi klienti te beje kerkesen Help :



## 5.Pershkrimi i metodave dhe rezultatet pas testimit

# 5.1 Pershkrimi i metodave pa hyrje

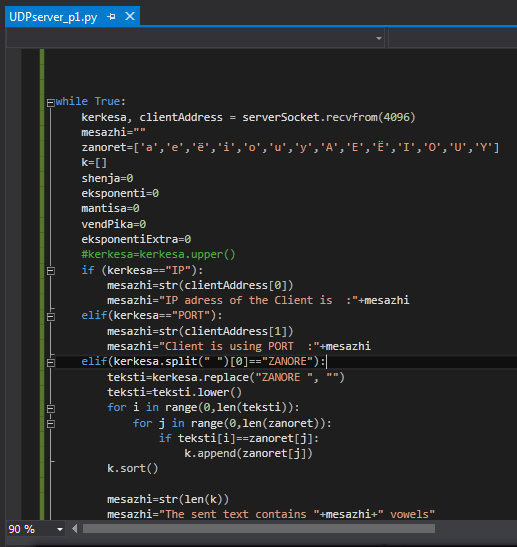
* + 1. Metoda IP

Request : IP

Pergjigjia nga serveri : Përcakton dhe kthen IP adresën e klientit në formë dhjetore me pike në një tekst mesazh p.sh. (10.10.7.251)

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

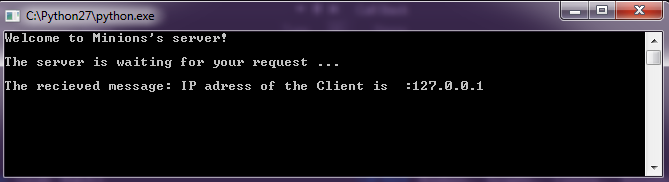
Realizimi permes kodit:

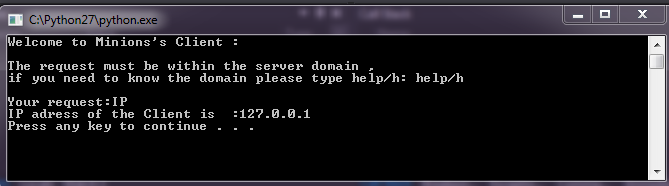


Shpjegim I kodit:Permes metodes clientAddress serveri merr ip e klientit dhe e kthen prap te ai.

Rezultatet pas testimit

Pasi klienti te beje kerkesen IP I kthehet pergjigjia nga serveri





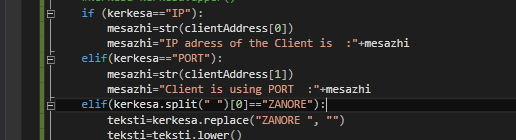
* + 1. Metoda PORT

Request : PORT

Pergjigjia nga serveri : Përcakton dhe kthen portin e klientit. (Ky duhet te jete porti I klientit dhe jo porti I serverit.)

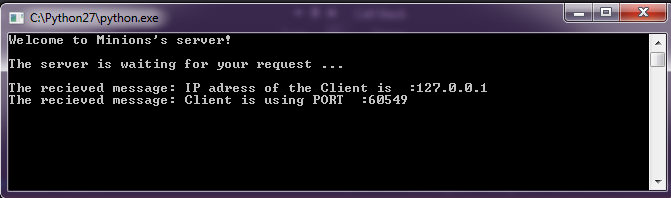
* Një mesazh jo me i gjate se 128 karaktere qe përmban IP adresën e klientit.
* Mesazhi nuk guxon te përmbaj NULL karakterin

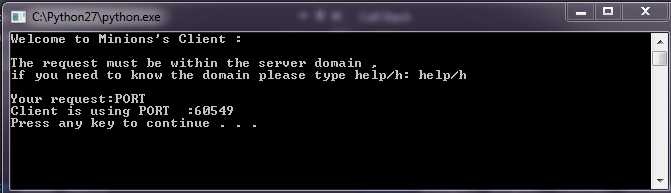
Realizimi permes kodit:



Rezultatet pas testimit

Pasi klienti te beje kerkesen PORT I kthehet pergjigjia nga serveri





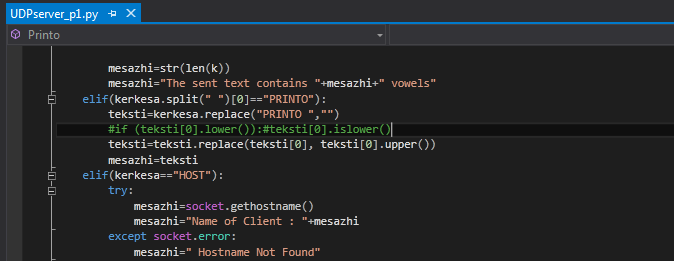
* + 1. Metoda HOST

Request : HOST

Pergjigjia nga serveri : 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

* 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:



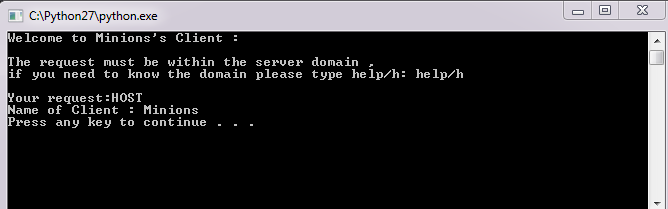
Shpjegimi I kodit:

Me ane te metodes .GetHostName() mirret emri i hostit i cili pastaj kthehet tek klienti, ne qofte se marrje e emrit te hostit ka qene e pasukseshme klienti lajmerohet qe emri i hostit nuk eshte marre.

Rezultatet pas testimit

Pasi klienti te beje kerkesen HOST I kthehet pergjigjia nga serveri





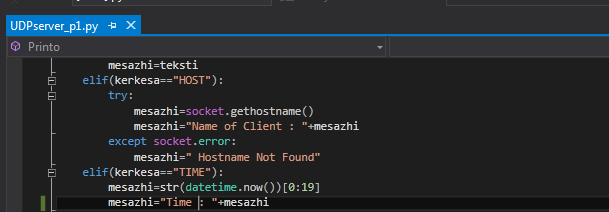
* + 1. Metoda TIME

Request : TIME

Pergjigjia nga serveri : Përcakton kohen aktuale ne server dhe e dërgon atë tek klienti si format te

lexueshme për njerëzit.

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

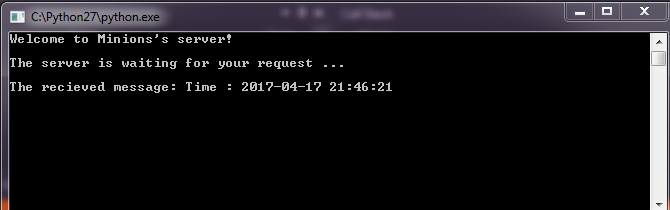
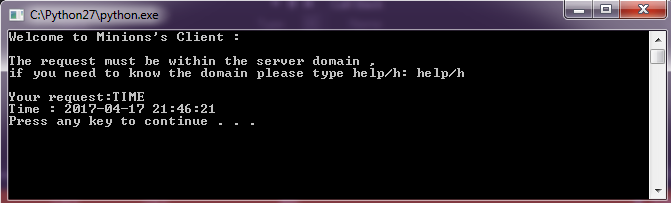
Realizimi permes kodit: 

Shpjegimi I kodit:

Permes klases DateTime mirret koha aktuale e sistemit dhe kthehet ne string ne formatin “HH:mm:ss tt

Rezultatet pas testimit

Pasi klienti te beje kerkesen TIME I kthehet pergjigjia nga serveri

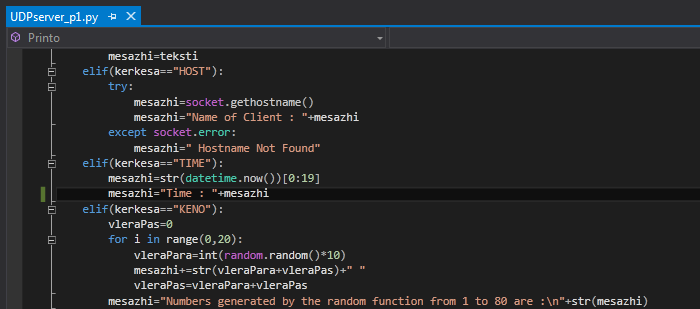
 

* + 1. Metoda KENO

Request : KENO

Pergjigjia nga serveri : Kthen 20 numra nga rangu [1,80].

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

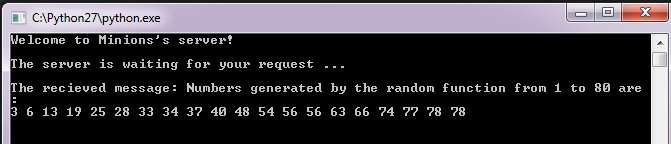
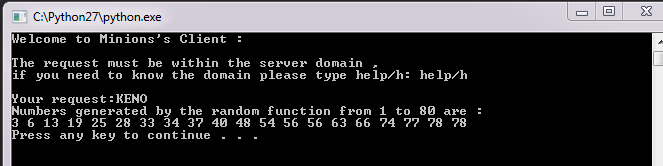
Realizimi permes kodit: 

Shpjegimi I kodit:

Klasa Random perdoret per gjenerimin e numrave te rastesishem te cilat ruhen ne nje array te tipit int dhe permes unazes for gjenerehen 20 numra random nga 1 deri 80 permes 20 cikleve . ku ne secilin cikil shikohet se a eshte numer I perseritur ne vargun e numrave te meparshem qe te mos perseritet.

Rezultatet pas testimit

Pasi klienti te beje kerkesen KENO I kthehet pergjigjia nga serveri

# 5.2 Pershkrimi i metodave me hyrje

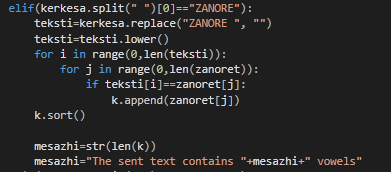
5.2.1 Metoda ZANORE

Request : ZANORE{HAPESIRE}tekst

Pergjigjia nga serveri : Gjen numrin e zanoreve ne tekst dhe kthen pergjigjien

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

Realizimi permes kodit:

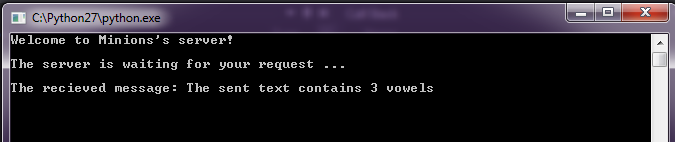
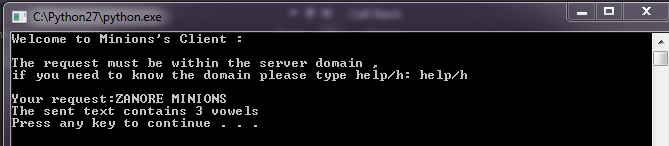
C:\Users\SINKOPA\Desktop\18009582_1757915510901747_2057160418_n.png 

Shpjegimi I kodit:

Fillimisht deklarohet vargu I zanoreve “zanoret”,pasi klienti te jep fjalen serveri e krahason nese ne fjalen e dhene nga klienti ndodhet ndonje karakter I vargut “zanoret” dhe rrit variablen per nje.Ne fund kthen numrin e zanoreve qe ndodhen ne fjalen e dhene nga klienti.

Rezultatet pas testimit

Pasi klienti te beje kerkesen ZANORE FJALE I kthehet pergjigjia nga serveri

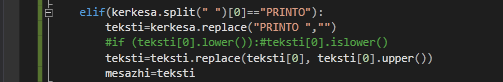
 

5.2.2 Metoda PRINTO

Request : PRINTO

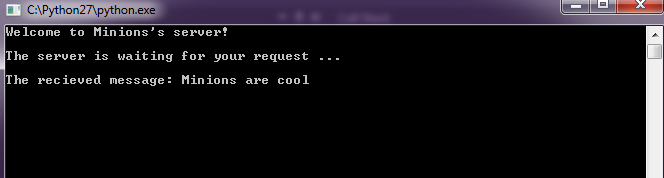
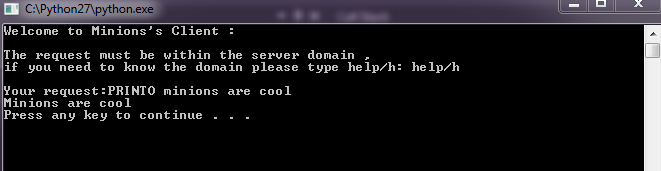
Pergjigjia nga serveri : Kthen fjalinë e shtypur ne tekst.Hapësirat ne fillim dhe ne fund te fjalisë nuk duhet te kthehen

Realizimi permes kodit:



Rezultatet pas testimit

Pasi klienti te beje kerkesen PRINTO FJALINE I kthehet pergjigjia nga serveri

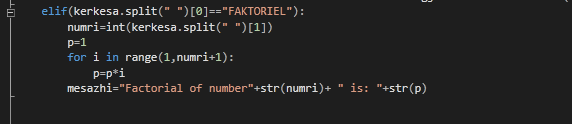
5.2.3 Metoda FAKTORIEL

Request : FAKTORIEL{HAPESIRE}numer

Pergjigjia nga serveri : Gjen faktorielin si rezultat i parametrit të dhënë hyrës. Numër i plotë psh.

FAKTORIEL 5 kthen si rezultat 120

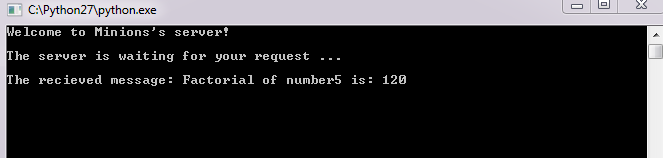
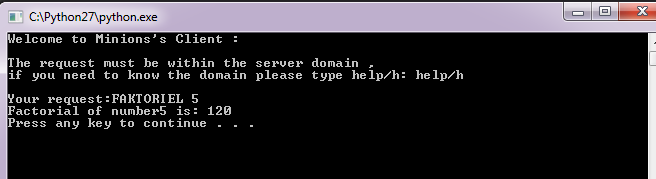
Realizimi permes kodit:



Shpjegimi I kodit:Sipas nje unaze for e cilva fillon nga 1 deri te nr qe e jep klienti dhe llogaritet si p\*I

Rezultatet pas testimit

Pasi klienti te beje kerkesen FAKTORIEL 5 I kthehet pergjigjia nga serveri

5.2.4 Metoda KONVERTO

Request : KONVERTO{HAPESIRE}opcioni{HAPESIRE}numer

Pergjigjia nga serveri : Kthen si rezultat konvertimin e opcioneve varësisht opcionit të zgjedhur.

* Lista e parametrave opcioni janë:
* CelsiusToKelvin
* CelsiusToFahrenheit
* KelvinToFahrenheit
* KelvinToCelsius
* FahrenheitToCelsius
* FahrenheitToKelvin
* PoundToKilogram
* KilogramToPound

Realizimi permes kodit:

listaNjesi=['CelsiusToKelvin','CelsiusToFahrenheit','KelvinToFahrenheit','KelvinToCelsius','FahrenheitToCelsius','FahrenheitToKelvin','PoundToKilogram','KilogramToPound']

def zgjedhFunksionin(zgjedhja, numri):

if(zgjedhja==listaNjesi[0]):

return float(numri)-273.15

elif(zgjedhja==listaNjesi[1]):

return numri \* 9/5 + 32

elif(zgjedhja==listaNjesi[2]):

return numri \* 9/5 - 459.67

elif(zgjedhja==listaNjesi[3]):

return numri - 273.15

elif(zgjedhja==listaNjesi[4]):

return (numri- 32) \* 5/9

elif(zgjedhja==listaNjesi[5]):

return (numri + 459.67) \* 5/9

elif(zgjedhja==listaNjesi[6]):

return numri \* 0.45359237

elif(zgjedhja==listaNjesi[7]):

return numri / 0.45359237

else:

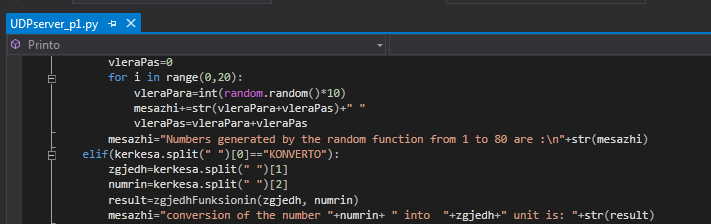
Printo()

def Printo():

print("Please choose a measure unit from the list and type a number you want to convert :")

for i in range(0,8):

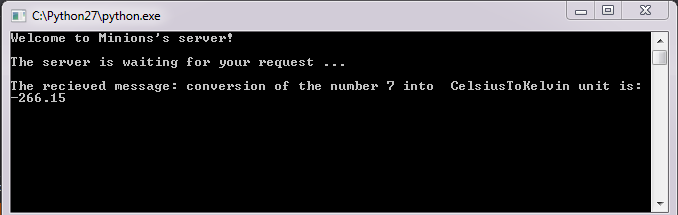
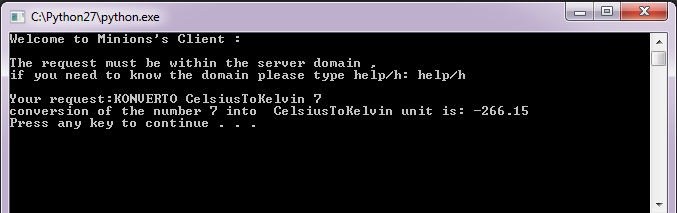
print(listaNjesi[i])



Shpjegimi I kodit:Varsishet se cilen nga mundesit e dhena per konvertim e zgjedh klienti I kthehet pergjigjia nga serveri.

Rezultatet pas testimit

Pasi klienti te beje kerkesen I kthehet pergjigjia nga serveri

## 6.Pershkrimi i metodave personale dhe rezultatet pas testimit

# Pershkrimi i metodes e realizuar ne GUI

* + 1. Metoda COLOR

Kerkesa :COLORGAME

Pergjigjia: Color Game eshte nje loje e krijuar permes GUI(Graphical User Interface) ideja e se ciles eshte gjetja e ngjyres se tekstit dhe jo teksti.

Realizimi permes kodit :

elif(kerkesa =="COLORGAME"):

#importohet moduli per krijimin e GUI

import tkinter

#per gjenerimin e numrave random

import random

#lista e ngjyrave te mundshme

colours = ['Red','Blue','Green','Pink','Black','Yellow','Orange','White','Purple','Brown']

#piket ne fillim inicializohen ne zero

score=0

#koha(timeleft) inicializohet ne 30 sekonda

timeleft=30

#funksioni I cili starton lojen

def startGame(event):

#nese ka ende kohe…

if timeleft == 30:

#fillo countdown timer.

countdown()

#thirrja e funskionit per zgjedhjen e ngjyrave

nextColour()

#funksioni per zgjedhjen e ngjyrave

def nextColour():

global score

global timeleft

#nese loja starton

if timeleft > 0:

#aktivizimi I tekst box

e.focus\_set()

#nese ngjyra e shtypur ne text box eshte e njejte me ngjyren e teksit

if e.get().lower() == colours[1].lower():

#rrit piket per nje

score += 1

#pasi te klikohe Enter pastohet tekst box

e.delete(0, tkinter.END)

#lista e ngjyrave ne baze random

random.shuffle(colours)

#ndryshon ngjyren , duke ndryshuar tekstin edhe ngjyren ne baze random

label.config(fg=str(colours[1]), text=str(colours[0]))

#ndrysho piket

scoreLabel.config(text="Score: " + str(score))

elif timeleft==0: #nese mbaron koha (timeleft=0) at’her mbaron loja

Label1 = tkinter.Label(root, text="Game Over !" , font=('Helvetica', 20))

Label1.pack();

#funksioni countdown timer

def countdown():

global timeleft

#nese eshte duke u luajtur loja

if timeleft > 0:

#zvoglo kohen per 1

timeleft -= 1

#ndrysho time left label.

timeLabel.config(text="Time left: " + str(timeleft))

#eksekutohet perseri funksioni pas 1 sekonde

timeLabel.after(1000, countdown)

#krijon nje GUI window.

root = tkinter.Tk()

#vendos titullin

root.title("Color Game")

#vendos madhesin e dritares

root.geometry("400x220")

#instructions label.

instructions = tkinter.Label(root, text="Type the colour of the words, and not the word text!", font=('Helvetica', 12))

instructions.pack()

#score label.

scoreLabel = tkinter.Label(root, text="Press enter to start", font=('Helvetica', 12))

scoreLabel.pack()

#time left label.

timeLabel = tkinter.Label(root, text="Time left: " + str(timeleft), font=('Helvetica', 12))

timeLabel.pack()

#label per ti shfaqur ngjyrat

label = tkinter.Label(root, font=('Helvetica', 60))

label.pack()

#tekst boxi per ti shtyput ngjyrat

e = tkinter.Entry(root)

#ekzekutohet 'startGame' funksioni nese shtypet Enter

root.bind('<Return>', startGame)

e.pack()

#vihet fokusi ne tekst box

e.focus\_set()

#startohet GUI

root.mainloop()

Shpjegimi I kodit:

Filliisht importohet moduli tkinter i cili mundson krijimiin e GUI(Graphical

User Interface) dhe per krijmin e random numbers importohet moduli random .

Per te krijuar nje GUI dritare(window) perdoret root.tkinter.Tk() (root-emri I dritares) , me pas vendoset titulli I dritares permes title("Color Game")dhe madhesia e dritares permes geometry("400x220").

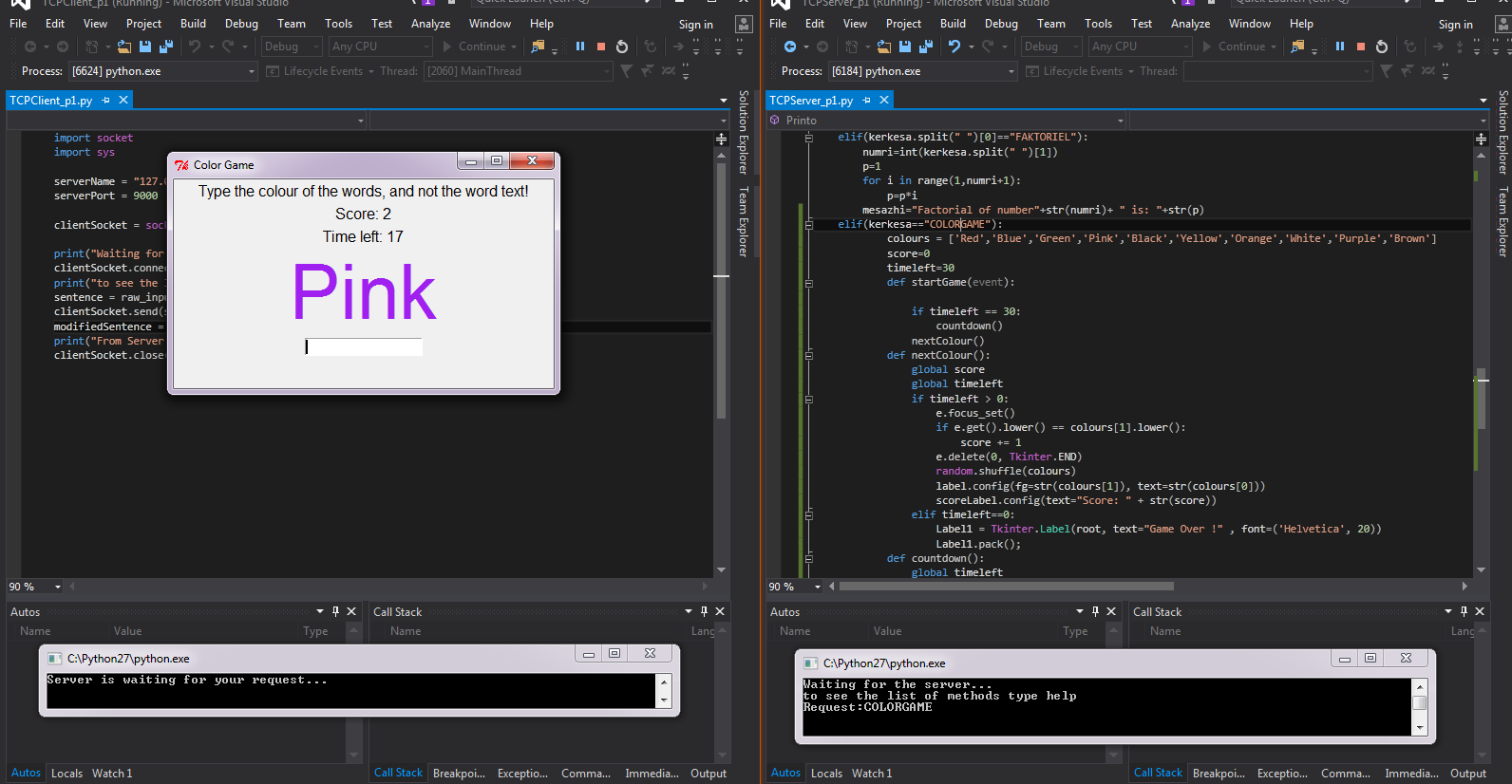
Deklarohet nje liste me ngjyrat e mundshme "colors", dhe dy variabla globale score dhe timeleft te cilat fillimisht incializohen ne score=0 dhe timeleft=30 sekonds

Me pas implementohen funksionet startGame(event): , countdown(),nextColour()

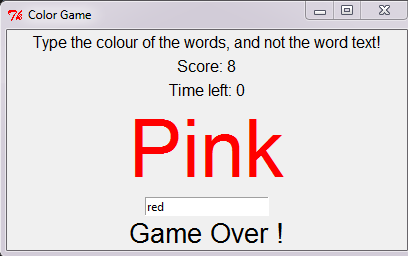
Me ane te funksionit startGame(event): startohet loja Parametri event percakton qe ky funksion thirret vetem kur klikohet tasti, mbrenda tij gjenden dy funksione tjera countdown() permes se ciles mundesohet qe koha (timeleft) te zbritet per nje pas qdo sekondi , dhe nextColor() e cila mundeson nderrimin e ngjyres ne baze te metodes random. Ne fund te kodit eshte implementuar pjesa e dizajnit te GUI dritares , dhe mainloop() e cila mundeson startimin e dritares GUI.

Rezultatet pas testimit

Pasi klienti te beje kerkesen I kthehet pergjigjia nga serveri



Nese timeleff==0 :



# Pershkrimi i metodave te tjera

* + 1. Metoda THJESHTE

Kerkesa :THJESHTE

Pergjigjia nga serveri:Tregon se cilet jane numrat e thjeshte ne nje rang I cili caktohet nga klienti

Realizimi permes kodit:

elif(kerkesa =="THJESHTE"):

lower = int(input("Enter lower range: "))

upper = int(input("Enter upper range: "))

print("Prime numbers between",lower,"and",upper,"are:")

for num in range(lower,upper + 1):

# prime numbers are greater than 1

if num > 1:

for i in range(2,num):

if (num % i) == 0:

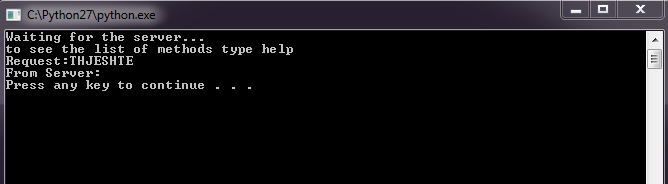
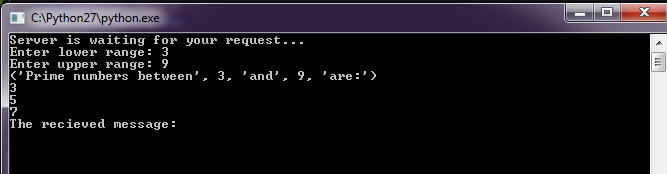
break

else:

print(num)

Rezultatet pas testimit

Pasi klienti te beje kerkesen I kthehet pergjigjia nga serveri

* + 1. Metoda LARGO

Kerkesa :LARGO

Pergjigjia nga serveri:Funsion I cili largon shenjat e pikesimit ne fjalin e dhene nga klienti

Realizimi permes kodit:

elif(kerkesa =="LARGO"):

shenjat = '''!()-[]{};:'"\,<>./?@#$%^&\*\_~'''

str = input("Shkruani nje fjale qe permban shenjat e pikesimit : ")

#largimi I shenjave te pikesimit nga fjalia

no\_punct = ""

for char in str:

if char not in shenjat:

no\_punct = no\_punct + char

#Paraqitja e fjales pa shenja te pikesimit

print(no\_punct)

Rezultatet pas testimit

Pasi klienti te beje kerkesen I kthehet pergjigjia nga serveri

