Отчет по четвертой практической работе

Выполнение задания 1:

Код программы:

Клиент:

import socket

messages = ["Hello, bro!", "How are you?", "See you later!"]

host = '127.0.0.1'

port = 5555

for message in messages:

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client.connect((host, port))

client.send(message.encode('utf-8'))

client.close()

Сервер:

import socket

import threading

def handle\_client(client\_socket):

while True:

data = client\_socket.recv(1024)

if not data:

break

print(f"Received from {client\_socket.getpeername()}: {data.decode('utf-8')}")

client\_socket.close()

host = '127.0.0.1'

port = 5555

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server.bind((host, port))

server.listen(5)

print(f"[\*] Listening on {host}:{port}")

while True:

client, addr = server.accept()

print(f"[\*] Accepted connection from {addr[0]}:{addr[1]}")

client\_handler = threading.Thread(target=handle\_client, args=(client,))

client\_handler.start()

Результат работы программы:

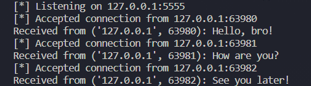


Рисунок 1 – Результат работы программы

Выполнение задания 2:

Код программы:

Клиент:

import socket

import pickle

from myClass import MyClass

my\_object = MyClass("Dmitry", 19)

host = '127.0.0.1'

port = 5555

serialized\_object = pickle.dumps(my\_object)

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client.connect((host, port))

client.send(serialized\_object)

client.close()

Сервер:

import socket

import pickle

from myClass import MyClass

host = '127.0.0.1'

port = 5555

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server.bind((host, port))

server.listen(5)

print(f"[\*] Listening on {host}:{port}")

while True:

client, addr = server.accept()

print(f"[\*] Accepted connection from {addr[0]}:{addr[1]}")

data = client.recv(1024)

if not data:

break

received\_object = pickle.loads(data)

print("Received object:")

print(received\_object.get\_info())

received\_object.increment\_age()

print("Updated object:")

print(received\_object.get\_info())

client.close()

Класс:

class MyClass:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def get\_info(self):

return f"First name: {self.name}, Age: {self.age}"

def increment\_age(self):

self.age += 1

Результат работы программы:

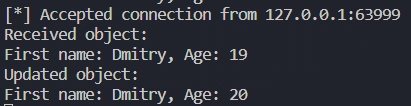


Рисунок 2 – Результат работы программы

Выполнение задания 3:

Код программы:

import socket

import threading

def start\_server():

host = '127.0.0.1'

port = 5555

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server.bind((host, port))

server.listen(1) # Ожидаем только одно соединение

print(f"[\*] Listening on {host}:{port}")

client\_socket, addr = server.accept()

print(f"[\*] Accepted connection from {addr[0]}:{addr[1]}")

handle\_connection(client\_socket)

def start\_client():

host = input("Enter server IP: ")

port = 5555

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client.connect((host, port))

print(f"[\*] Connected to {host}:{port}")

handle\_connection(client)

def handle\_connection(socket):

while True:

message = input("Enter message (or 'exit' to quit): ")

if message.lower() == 'exit':

break

socket.send(message.encode('utf-8'))

data = socket.recv(1024)

print(f"Received from server: {data.decode('utf-8')}")

socket.close()

def main():

role = input("Choose role (server/client): ")

if role.lower() == 'server':

start\_server()

elif role.lower() == 'client':

start\_client()

else:

print("Invalid role. Choose either 'server' or 'client'.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

Результат работы программы:



Рисунок 3 – Сервер



Рисунок 4 – Клиент

Выполнение задания 4:

Код программы:

import requests

from bs4 import BeautifulSoup

url = 'http://example.org'

response = requests.get(url)

if response.status\_code == 200:

html\_content = response.text

soup = BeautifulSoup(html\_content, 'html.parser')

title\_tag = soup.find('title')

if title\_tag:

title\_text = title\_tag.text

print(f"Title: {title\_text}")

else:

print("Title tag not found in the HTML.")

else:

print(f"HTTP request failed with status code {response.status\_code}.")

Результат работы программы:



Рисунок 5 – Результат работы программы