МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №2**

з дисципліни «ВЕБ-технології та ВЕБ-дизайн-2. Веб-орієнтована розробка програмного забезпечення»

на тему: «Створення простого веб-сервісу»

Варіант №13

**Виконав:**

студент гр. БС-83

Розмариця О.А.

**Перевірив:**

ас. Матвійчук О. В.

ас. Давидько О. Б.,

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Завдання:**

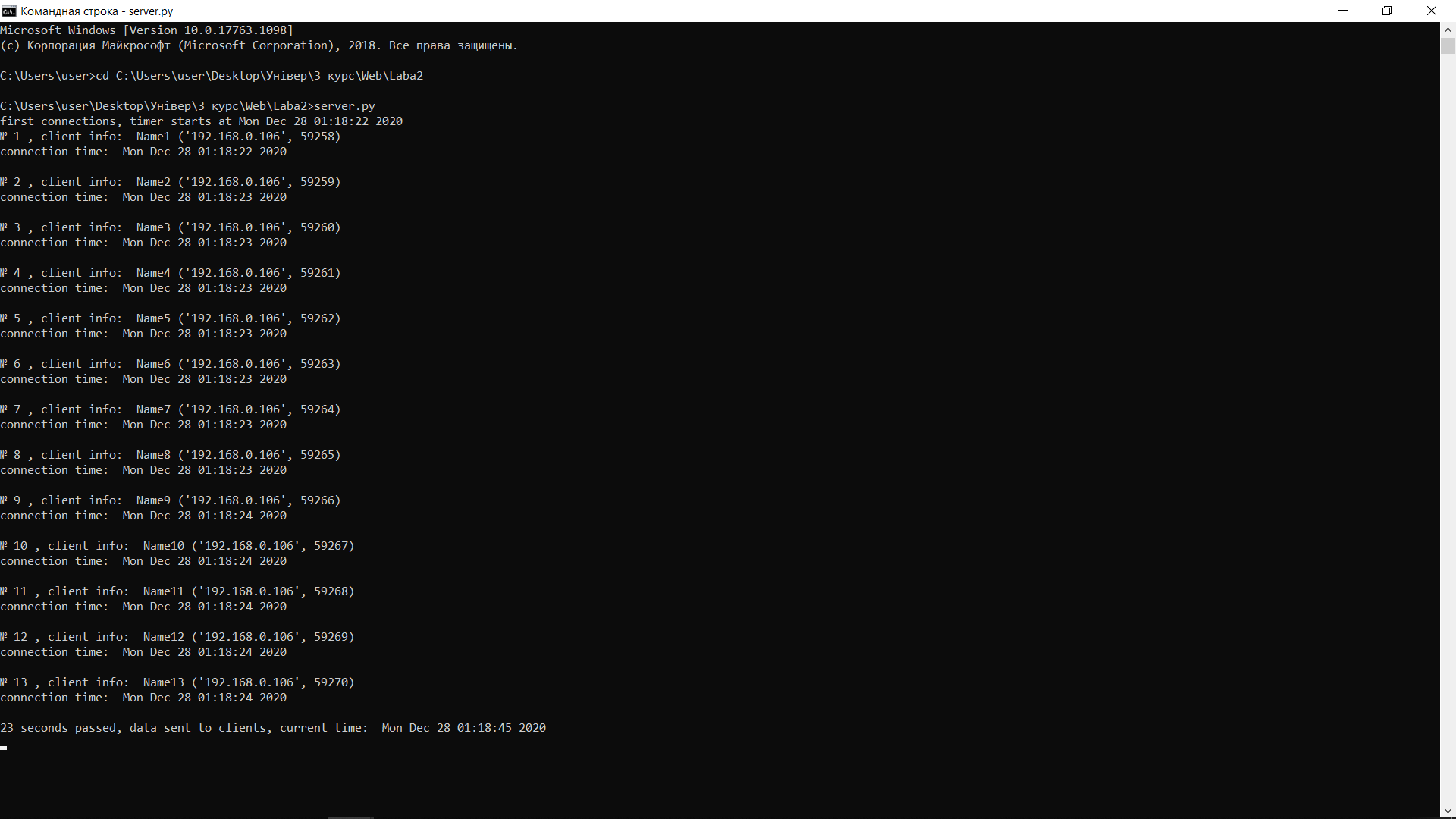
1. Створити веб-сервіс та клієнт до нього.
2. Сервіс та клієнт повинні взаємодіяти шляхом передачі даних по TCP сокетах в форматі Прізвище[0] <= “О” ? JSON : XML
3. Кожен з клієнтів після підключення повинен передати серверу структуру з 1 ключем “id” та значенням “Прізвище\_(#клієнта)”
4. Після підключення першого клієнта сервер має запустити глобальний таймер на #Варіант + 10 секунд
5. По завершенню таймера відправити кожному з клієнтів список з усіма підключеними клієнтами з датами їх підключення, а також дату старту таймера
6. Для отримання змін в сокетах скористатись механізмами для роботи із сокетами, що підтримуються обраною вами мовою. наступним системним інструментом Select()

      6\*) Екстра варіант [epoll/kqueue/I/O Completion Ports] (в залежності від обраної вами системи)

1. Перевірити роботу системи з (#варіант mod 10) + 10 клієнтами
2. В процесі роботи не використовувати сторонні бібліотеки для роботи в сокетами, з XML та  JSON по необхідності

**Сервер:**

import socket  
import time  
from json import dumps  
from threading import Thread  
  
connections\_array = []  
connect\_time\_array = []  
start\_time = 0  
connections\_amount = 0  
sockets = []  
connections\_list = {}  
sock = socket.socket()  
sock.bind(('', 5050))  
sock.listen(13)  
  
  
def sending(connections\_array):  
 for i in range(13):  
 convert\_to = dumps(connections\_array)  
 sockets[i].send(convert\_to.encode("utf-8"))  
 i = i + 1  
 print("23 seconds passed, data sent to clients, current time: ", time.ctime())  
  
  
def timer(connections\_array):  
 while time.process\_time() < 23:  
 do\_nothing = 0  
 else:  
 sending(connections\_array)  
  
  
while True:  
  
 conn, addr = sock.accept()  
 time.sleep(0.1)  
 connections\_amount = connections\_amount + 1  
 if connections\_amount == 1:  
 start\_time = time.ctime()  
 print("first connections, timer starts at", start\_time)  
 th\_1 = Thread(target=timer, args=(connections\_array,))  
 th\_1.start()  
 connections\_array.append(start\_time)  
 connection\_time = time.ctime()  
 data = conn.recv(4096)  
 if not data:  
 break  
 print("№", connections\_amount, ", client info: ", data.decode('utf-8'), addr)  
 print("connection time: ", connection\_time, "\n")  
 connections\_list = {"name: ": data.decode('utf-8'), "adress: ": addr, "connection time: ": connection\_time}  
 connections\_array.append(connections\_list)  
 sockets.append(conn)



**Клиент:**

import socket  
  
  
def users(i):  
 Names = ["Name1", "Name2", "Name3", "Name4", "Name5", "Name6", "Name7", "Name8",  
 "Name9", "Name10", "Name11", "Name12", "Name13"]  
 return Names[i]  
  
  
sock = [socket.socket() for i in range(13)]  
  
for i in range(13):  
 sock[i].connect((socket.gethostname(), 5050))  
 sock[i].send(users(i).encode())  
  
for i in range(13):  
 data = sock[i].recv(4096)  
 print("user: ", users(i), "\n timer starts at: ", data.decode())

