

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: WEB-программирование

Отчет

Лабораторная работа 1

Выполнил:
Комиссаров Александр
К33402

Проверил:
Говоров А. И.

Санкт-Петербург

2022 г.

Ход работы

1. Реализовать клиентскую и серверную часть приложения. Клиент отправляет серверу сообщение «Hello, server». Сообщение должно отразиться на стороне сервера. Сервер в ответ отправляет клиенту сообщение «Hello, client». Сообщение должно отобразиться у клиента.

```
client.py - C:\Users\Alex\Desktop\donca\WEB\students\K33402\Komissarov Alexander\LW1\task1\client.py (3.8.0)
File Edit Format Run Options Window Help

#client

import socket
import time

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 10203))
conn.send(b'Hello, server! \n')
data = conn.recv(9988)
data = data.decode('utf-8')
print(data)
conn.close()
time.sleep(55)
```

```
server.py - C:\Users\Alex\Desktop\donca\WEB\students\K33402\Komissarov Alexander\LW1\task1\server.py (3.8.0)
File Edit Format Run Options Window Help

#server

import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(('', 10203))
sock.listen(1)

while True:
    clientsocket, address = sock.accept()
    print ('Incoming connection:', address)
    msg=clientsocket.recv(10203).decode('utf-8')
    print(msg)
    clientsocket.send(b'Hello, client! \n')
    clientsocket.close()
```

```
Hello, client!
= RESTART: C:\Users\Alex\Desktop\donca\WEB\students\K33402\Komissarov Alexander\LW1\task1\server.py
Incoming connection: ('127.0.0.1', 61862)
Hello, server!
```

2. Реализовать клиентскую и серверную часть приложения. Клиент запрашивает у сервера выполнение математической операции, параметры, которые вводятся с клавиатуры. Сервер обрабатывает полученные данные и возвращает результат клиенту.

```
client.py - C:\Users\Alex\Desktop\donca\WEB\students\K33402\Komissarov Alexander\LW1\task2\client.py (3.8.0)
File Edit Format Run Options Window Help

#client

import socket
import time

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 10203))
print("Введите длины двух катетов прямоугольного треугольника через пробел:")
raw_input=str(input()).encode()
conn.send(raw_input)
data = conn.recv(9988)
data = data.decode('utf-8')
print(data)
conn.close()
time.sleep(55)
```

```
server.py - C:\Users\Alex\Desktop\donca\WEB\students\K33402\Komissarov Alexander\LW1\task2\server.py (3.8.0)
File Edit Format Run Options Window Help

#server

import socket
import math

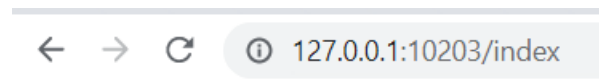
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(('', 10203))
sock.listen(1)

while True:
    clientsocket, address = sock.accept()
    print ('Incoming connection:', address)
    msg=clientsocket.recv(10203).decode('utf-8')
    a, b = msg.split()
    a, b = int(a), int(b)
    c = str(math.sqrt(a*a+b*b))
    print(c)
    sending="Длина гипотенузы равна: "+c
    clientsocket.send(str(sending).encode())
    clientsocket.close()
```

```
Введите длины двух катетов прямоугольного треугольника через пробел:
3 4
Длина гипотенузы равна: 5.0
Incoming connection: ('127.0.0.1', 62595)
5.0
```

3. Необходимо написать простой web-сервер для обработки GET и POST http-запросов средствами Python и библиотеки socket. Сделать сервер, который может:

- Принять и записать информацию о дисциплине и оценке по дисциплине.
- Отдать информацию обо всех оценках по дисциплине в виде html-страницы.



Hello, world!

4. Реализовать двухпользовательский или многопользовательский чат.

```
#client
import socket
from threading import Thread

def send():
    while True:
        message = input()
        conn.send(message.encode('utf-8'))

def get():
    while True:
        data = conn.recv(9988).decode('utf-8')
        print(data)

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(('127.0.0.1', 10203))
Thread(target=send).start()
Thread(target=get).start()
```

```
#server
import socket
from threading import Thread

def send(message, sender):
    for client in clients:
        if sender != client:
            client.send(message)

def listen(client):
    while True:
        message = client.recv(9988)
        send(message, client)

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(('127.0.0.1', 10203))
conn.listen()
clients = []
while True:
    new_client, address = conn.accept()
    if new_client not in clients:
        clients.append(new_client)
        Thread(target=listen, args=[new_client]).start()
```

