

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Лабораторная работа № 1**

на тему  
“Работа с сокетами”  
по дисциплине **«WEB программирование»**

Выполнила: Исхакова Эмина

Группа: К33422

Проверил: Говоров А.И.



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург, 2021

### Задание 1.

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.

```
task1_client.py x ask1_server.py x
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('localhost', 7777))
sock.send('Hello, server! :).encode('utf-8'))

d = sock.recv(1024)

print(d.decode('utf-8'))
sock.close()
```

```
task1_client.py x ask1_server.py x
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(('localhost', 7777))
sock.listen(1)
conn, addr = sock.accept()

while True:
    data = conn.recv(1024)
    if not data:
        break
    print(data.decode('utf-8'))
    conn.send('Hello, client! :).encode('utf-8'))

conn.close()
```

Сервер включился, клиент подключился, сервер отправил приветствие, клиент его получил

```
ask1_server x task1_client x
V:\lab1_web\venv\Scripts\python.exe V:/lab1_web/ask1_server.py
Hello, server! :)

Process finished with exit code 0
```

```
ask1_server x task1_client x
V:\lab1_web\venv\Scripts\python.exe V:/lab1_web/task1_client.py
Hello, client! :)

Process finished with exit code 0
```

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. У меня 3ий вариант по списку, поэтому ищем площади трапеции.

```
task2_client x task2_server x
V:\lab1_web\venv\Scripts\python.exe V:/lab1_web/task2_server.py
connected: ('127.0.0.1', 49221)

V:\lab1_web\venv\Scripts\python.exe V:/lab1_web/task2_client.py
Enter two bases and its high or "exit" to finish the program
3 9 4
Square of the figure is 24.0;
```

3. Реализовать серверную часть приложения. Клиент подключается к серверу. В ответ клиент получает http-сообщение, содержащее html-страницу, которую сервер подгружает из файла

index.html.

Подключаем сервер

```
task3 x task3_server x task3_client x
V:\lab1_web\venv\Scripts\python.exe V:/lab1_web/task3.py
|
```

```
task3 x task3_server x task3_client x
V:\lab1_web\venv\Scripts\python.exe V:/lab1_web/task3_server.py
|
```

Подключаем клиента, который получает html-страницу

```
task3 x task3_server x task3_client x
V:\lab1_web\venv\Scripts\python.exe V:/lab1_web/task3_client.py
HTTP/1.0 200 OK

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Hello! I'm going to Germany!!!</title>
</head>
```

4. Реализовать двухпользовательский или многопользовательский чат.

В чате указан адресант каждого сообщения (какой именно user его отправил), а также подключения к серверу.

```
import socket
from threading import Thread
import time

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(('127.0.0.1', 7774))
sock.listen(10)
sock.setblocking(False)
users = []

def new_users():
    while True:
        sock.setblocking(True)
        clientsocket, addr = sock.accept()
        sock.setblocking(False)
        if clientsocket not in users:
            users.append((clientsocket, addr))
            print('New user:', addr)

def chat():
    while True:
        try:
            for user in users:
                message = user[0].recv(1024).decode('utf-8')
```

```
            for send_user in users:
                if send_user[0] == user[0]:
                    continue
                data = f'User {user[1]}: ' + message
                send_user[0].send(data.encode('utf8'))
            except socket.error:
                time.sleep(5)
            except KeyboardInterrupt:
                for user in users:
                    user[0].close()
                break

read = Thread(target=new_users)
read = Thread(target=chat)

read.start()
read.start()
```

```

import socket
from threading import Thread

def send_message():
    while True:
        sock.send(input().encode('utf-8'))

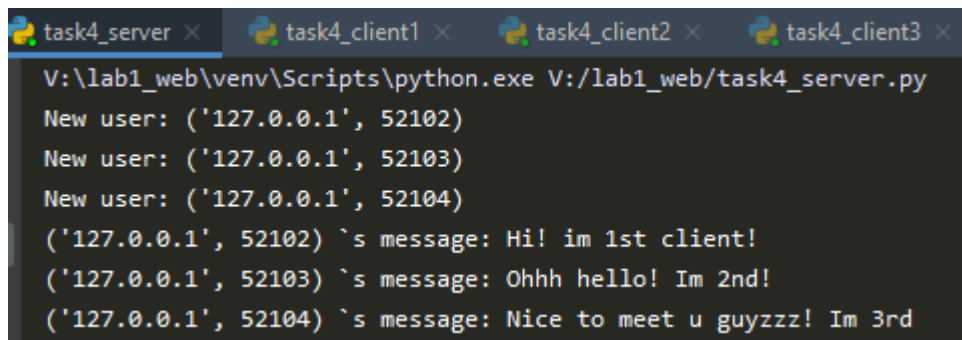
def receive_message():
    while True:
        data = sock.recv(1024)
        print(data.decode('utf-8'))

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('127.0.0.1', 7774))

send_thread = Thread(target=send_message)
get_thread = Thread(target=receive_message)

send_thread.start()
get_thread.start()

```



```

task4_server × task4_client1 × task4_client2 × task4_client3 ×
V:\lab1_web\venv\Scripts\python.exe V:/lab1_web/task4_server.py
New user: ('127.0.0.1', 52102)
New user: ('127.0.0.1', 52103)
New user: ('127.0.0.1', 52104)
('127.0.0.1', 52102) `s message: Hi! im 1st client!
('127.0.0.1', 52103) `s message: Ohhh hello! Im 2nd!
('127.0.0.1', 52104) `s message: Nice to meet u guyzzz! Im 3rd

```

Вывод:

В результате выполнения лабораторной работы были получены практические навыки, умения реализации web-серверов и использования сокетов.