

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ  
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №1  
по курсу «Основы Web-программирования»  
Тема: «Работа с сокетами»

Выполнил:

Елистратов В. Д.

Группа К32392

Проверил:

Говоров Антон Игоревич

Санкт-Петербург

2023 г.

## Содержание отчета

<b>Постановка задачи</b>	<b>3</b>
<b>Ход работы</b>	<b>4</b>
Задание №1	4
Задание №2	6
Задание №3	9
Задание №4	12
Вывод	17

## Постановка задачи

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.
  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.  
Вариант **d** – поиск площади параллелограмма.
  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.
  4. Реализовать многопользовательский чат
  5. 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.

## Ход работы

### Задание №1

Листинг программного кода

Server:

```
import socket
from time import sleep

IP = "127.0.0.1"

PORT = 14900
buffSize = 16384

listener = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

IP = socket.gethostbyname(socket.gethostname()) #получение IP
#print(IP)

listener.bind((IP, PORT))

data = (listener.recvfrom(buffSize)) #принимаем сообщение от клиента
print(data[0].decode("utf-8"))
address = data[1]

ms = "Hello, Client"
listener.sendto(ms.encode('utf-8'), address) # отправляем сообщение клиенту
```

Client:

```
import socket
from time import sleep

serverIP = "192.168.56.1" #вводим IP адрес который выдал сервер
PORT = 14900
buffSize = 16384

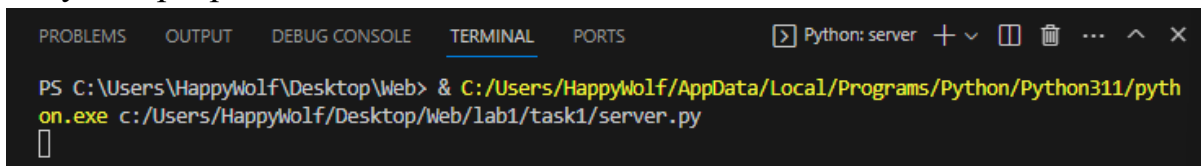
connection = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
connection.connect((serverIP, PORT))

ms = "Hello, Server"
connection.sendto(ms.encode('utf-8'), (serverIP, PORT) ) #отправляем сообщение

data = (connection.recv(buffSize)).decode("utf-8") #принимаем сообщение
print(data)
```

## Скриншоты работы программы

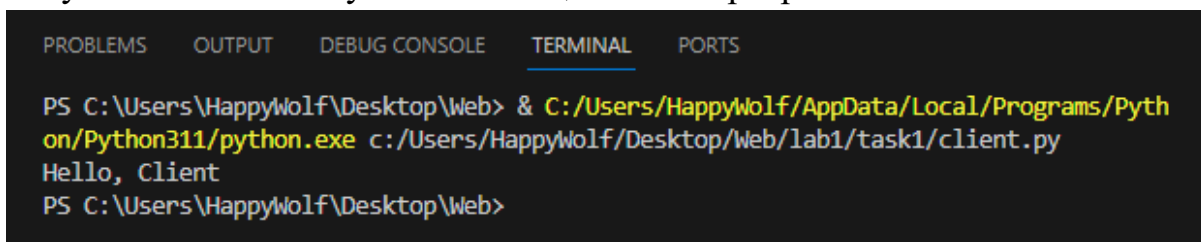
### 1. Запуск сервера



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python: server + - [ ] [ ] ... ^ X  
PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task1/server.py  
█
```

Рисунок 1.1 – запуск сервера

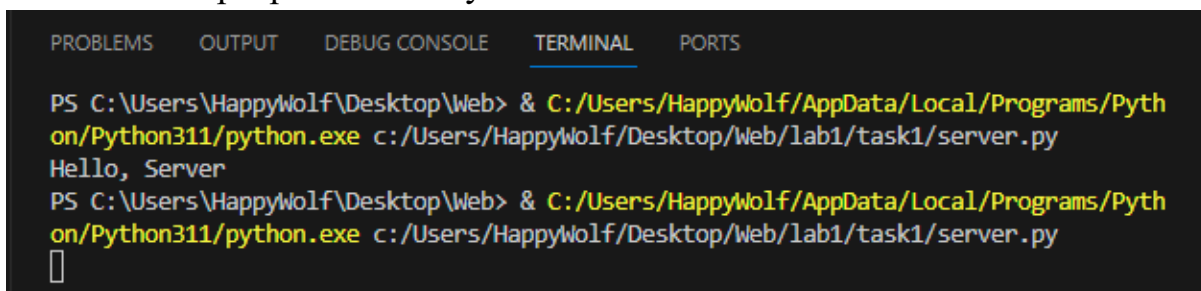
### 2. Запуск клиента и получение сообщения от сервера



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task1/client.py  
Hello, Client  
PS C:\Users\HappyWolf\Desktop\Web>
```

Рисунок 1.2 – запуск клиента

### 3. Состояние сервера после запуска клиента



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task1/server.py  
Hello, Server  
PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task1/server.py  
█
```

Рисунок 1.3 – приветствие от клиента

## Задание №2

### Листинг программного кода

Server:

```
import socket
from time import sleep

def Area(udata): #функция выполняющая подсчет площади параллелограмма
    rez = 0.0
    tp = int(udata[0])
    match tp:
        case 1:
            rez = int(udata[1]) * int(udata[2])
        case 2:
            rez = int(udata[1]) * int(udata[2]) * float(udata[3])
        case 3:
            rez = 0.5 * int(udata[1]) * int(udata[2]) * float(udata[3])
        case _:
            rez = -1

    rezMs = ""
    if (rez == -1):
        rezMs = "Error"
    else:
        rezMs = str(rez)

    return(rezMs)

IP = "127.0.0.1"

PORT = 14900
buffSize = 16384

listener = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

IP = socket.gethostbyname(socket.gethostname())
#print(IP)

listener.bind((IP, PORT))
listener.listen(10)

connection, address = listener.accept()
```

```

while True:
    try: #ожидание и получение данных от клиента, обработка, отправка
результатов
        data = connection.recv(16384).decode("utf-8")
        udata = list(data.split(' '))
        ms = Area(udata)
        connection.send(ms.encode("utf-8"))

    except KeyboardInterrupt:
        listener.close()
        print("Nope")
        break

```

Client:

```

import socket
from time import sleep

serverIP = "192.168.56.1"
PORT = 14900
buffSize = 16384

connection = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connection.connect((serverIP, PORT))

while True: #ввод данных и отправка их для обработки на сервер, поддерживаются
множественные запросы
    ms = input()
    connection.send(ms.encode('utf8'))
    data = connection.recv(buffSize)
    print(data.decode("utf-8") + '\n')

```

Скриншоты работы программы

### 1. Запуск сервера и клиента

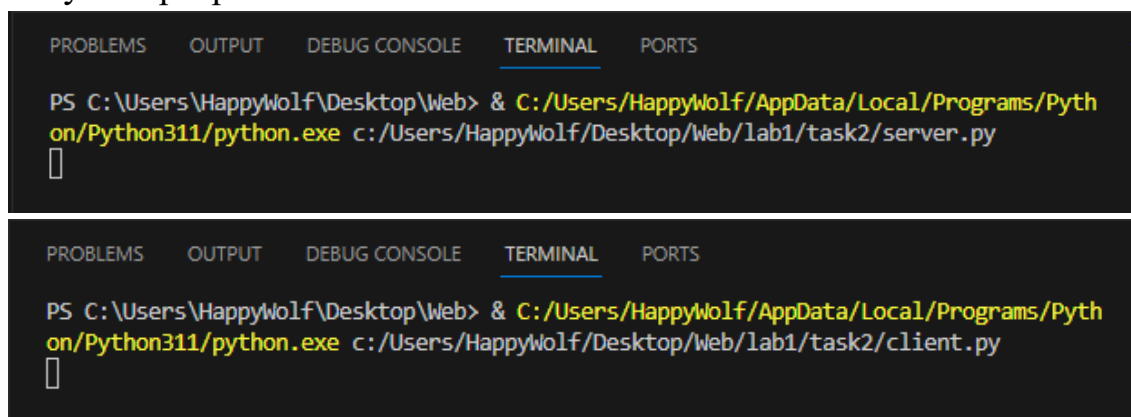
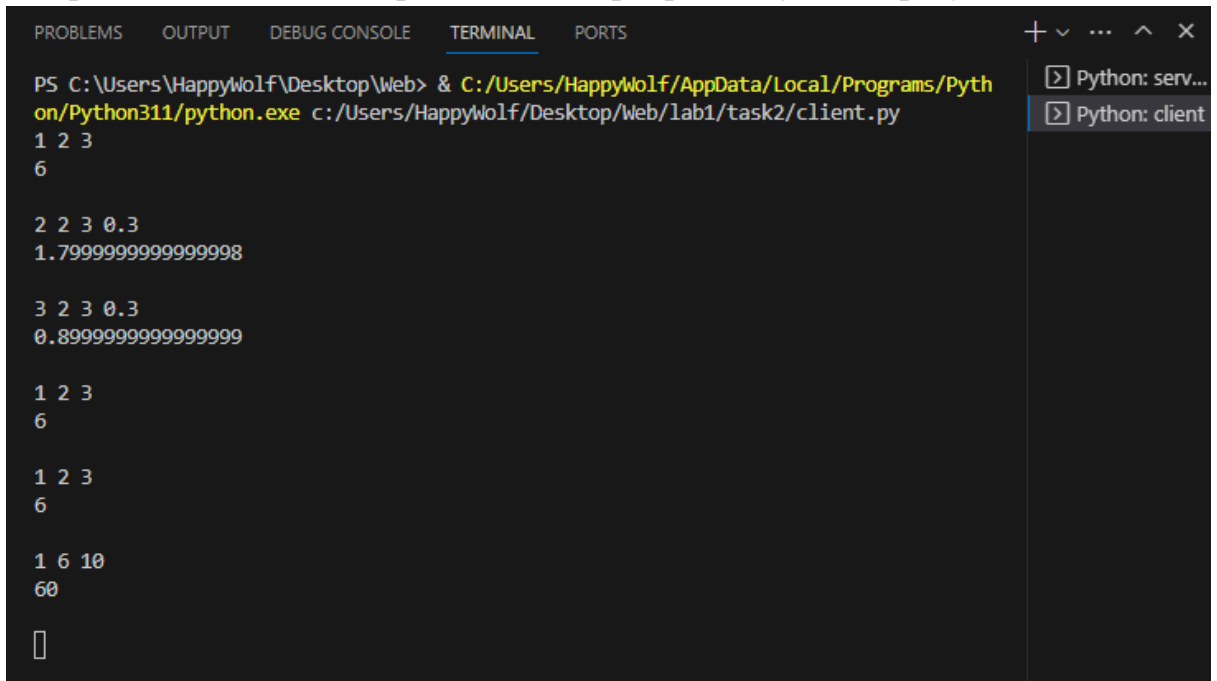


Рисунок 2.1, 2.2 – запуск сервера и клиента

## 2. Отправка данных для обработки на сервер и получение результатов



```
PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task2/client.py
1 2 3
6

2 2 3 0.3
1.7999999999999998

3 2 3 0.3
0.8999999999999999

1 2 3
6

1 2 3
6

1 6 10
60


```

Рисунок 2.3 – отправка запросов на сервер



### Задание №3

#### Листинг программного кода

Server:

```
import socket

def sendResponse(connection): #построение HTTP сообщения и отправка его
клиенту
    status = 'HTTP/1.1 200 OK\r\n'
    header = 'Content-Type: text/html\r\n'

    ms = status + header + "\r\n"

    htmlFile = open("lab1/task3/index.html", 'r') #чтение html кода из файла
    for string in htmlFile:
        ms += string

    connection.send(ms.encode("utf-8"))

PORT = 14900

IP = socket.gethostbyname(socket.gethostname())

listener = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

listener.bind((IP, PORT))
listener.listen(10)

while True:
    connection, address = listener.accept()
    try:
        sendResponse(connection)
        if connection:
            connection.close()

    except Exception as ex:
        print("=== Client connection failed ===\n", ex)
```

Client:

```
import socket

serverIP = "192.168.56.1"
PORT = 14900
buffSize = 16384

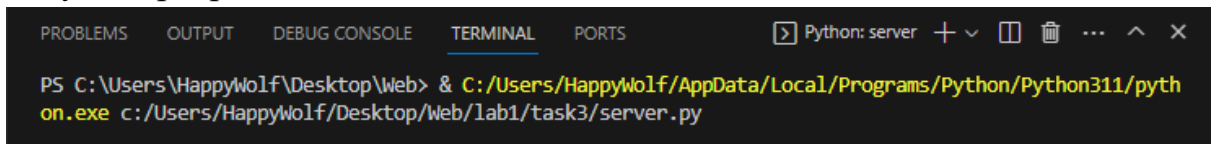
connection = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connection.connect((serverIP, PORT))

try:
    data = connection.recv(buffSize)
    data = data.decode("utf-8")
    print(data)

finally:
    connection.close()
```

Скриншоты работы программы

### 1. Запуск сервера

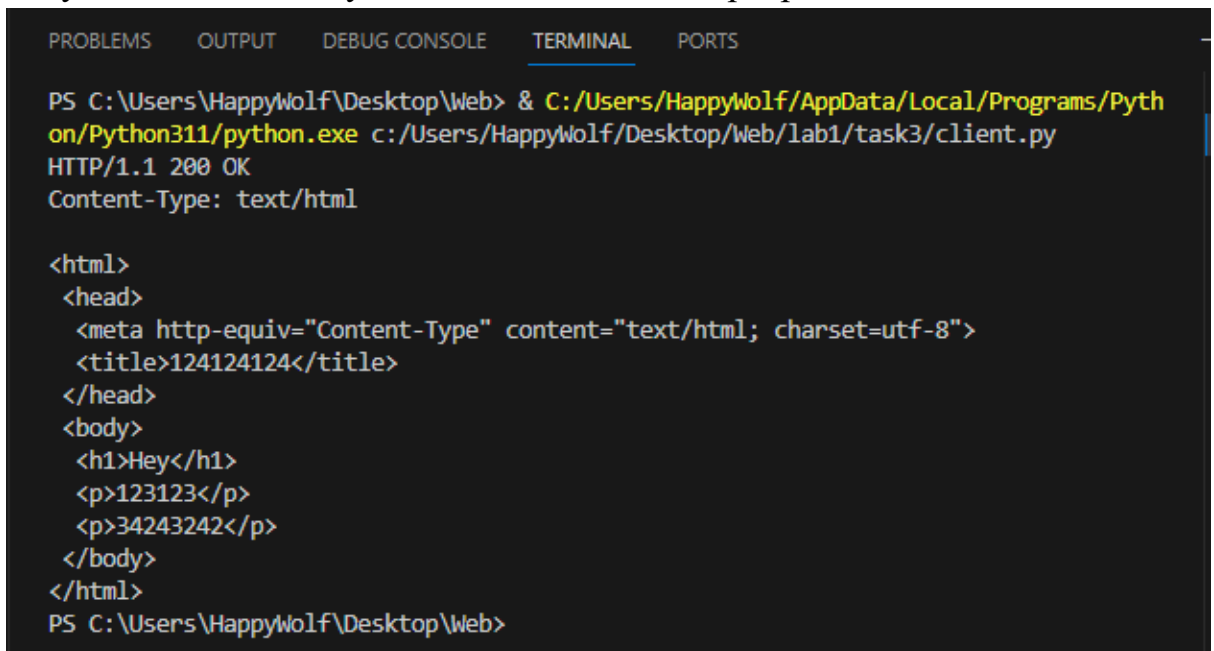
A screenshot of a terminal window with a dark background. The terminal shows a command prompt 'PS C:\Users\HappyWolf\Desktop\Web>' followed by the command to run a Python script: '& C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task3/server.py'. The terminal window has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. There are also window control icons on the right side of the terminal header.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python: server + - [] ... ^ x

PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python
on.exe c:/Users/HappyWolf/Desktop/Web/lab1/task3/server.py
```

Рисунок 3.1 – запуск сервера

### 2. Запуск клиента и получение сообщения от сервера

A screenshot of a terminal window with a dark background. The terminal shows a command prompt 'PS C:\Users\HappyWolf\Desktop\Web>' followed by the command to run a Python script: '& C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task3/client.py'. Below the command, the output of the client script is shown: 'HTTP/1.1 200 OK' and 'Content-Type: text/html'. Then, the received HTML content is displayed: '<html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <title>124124124</title> </head> <body> <h1>Hey</h1> <p>123123</p> <p>34243242</p> </body> </html>'. The terminal window has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. There are also window control icons on the right side of the terminal header.

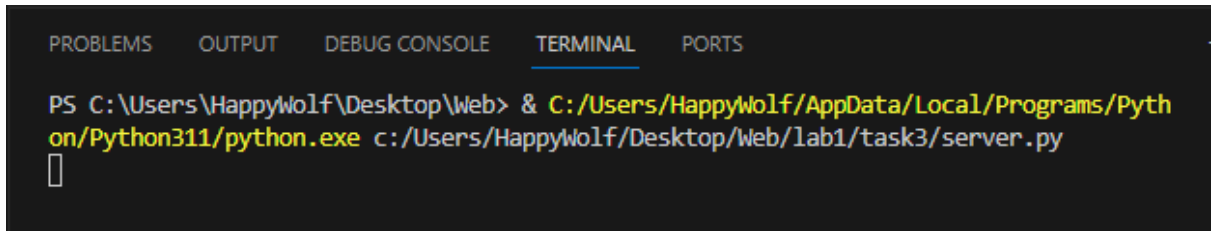
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Pyth
on/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task3/client.py
HTTP/1.1 200 OK
Content-Type: text/html

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>124124124</title>
</head>
<body>
  <h1>Hey</h1>
  <p>123123</p>
  <p>34243242</p>
</body>
</html>
PS C:\Users\HappyWolf\Desktop\Web>
```

Рисунок 3.2 – запуск клиента

### 3. Состояние сервера после запуска клиента

A screenshot of a terminal window with a dark background. At the top, there are five tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. The terminal shows a command prompt 'PS C:\Users\HappyWolf\Desktop\Web>' followed by a command to run a Python script: '& C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task3/server.py'. A cursor is visible on the line below the command.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task3/server.py
█
```

Рисунок 3.3 – состояние сервера

## Задание №4

### Листинг программного кода

Server:

```
import socket
import threading
from time import sleep

IP = "127.0.0.1"

PORT = 14900
buffSize = 16384

session = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
session.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

IP = socket.gethostbyname(socket.gethostname())

session.bind((IP, PORT))
session.listen(10)

userList = []

def listener(cSession):
    while True:
        ms = cSession.recv(buffSize).decode('utf-8')
        for i in userList:
            i.send(ms.encode('utf-8'))

        if(ms.find("exit_T") != -1):
            #print("EXIT")
            #print(ms, ' ', ms.find("exit_t"), '\n')
            userList.remove(cSession)
            break

while True:
    client_conn, address = session.accept()

    userList.append(client_conn)

    clientsServer = threading.Thread(target=listener, args=(client_conn,))
    clientsServer.daemon = True
    clientsServer.start()
```

Client:

```
import socket
import random
import threading
from time import sleep

name = input("Enter your name: ")

serverIP = "192.168.56.1"
PORT = 14900
buffSize = 16384

session = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
session.connect((serverIP, PORT))

def messWait():
    while True:
        data = session.recv(buffSize)
        print(data.decode("utf-8"))

listenerClient = threading.Thread(target=messWait)
listenerClient.daemon = True
listenerClient.start()

while True:
    ms = input()
    ms = name + ': ' + ms
    session.send(ms.encode('utf-8'))
    if(ms == (name + ": " + "exit_T")):
        break
```

Скриншоты работы программы

### 1. Запуск сервера

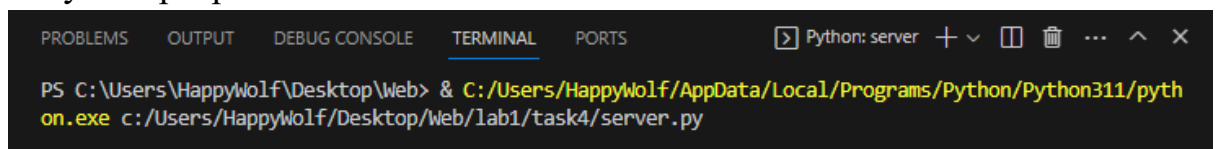
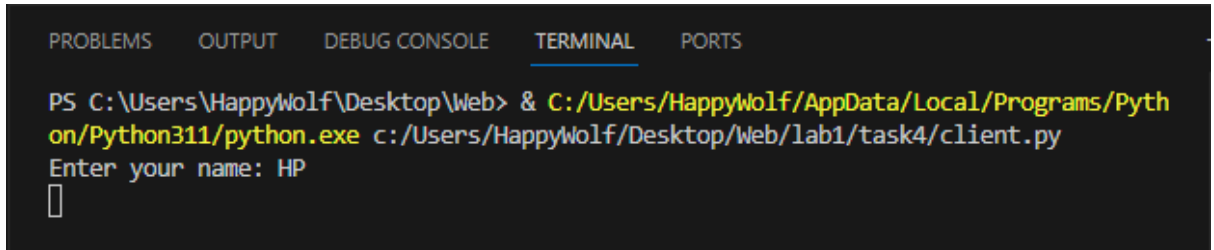


Рисунок 4.1 – запуск сервера

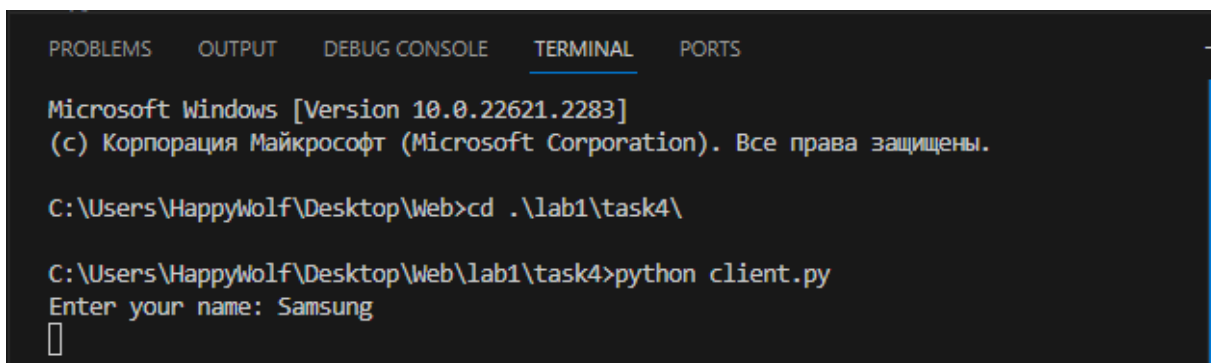
2. Множественный запуск клиентского кода (симуляция многопользовательского чата). При входе в чат запрашивается имя, чтобы различать пользователей будут выданы различные имена



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task4/client.py
Enter your name: HP
█
```

Рисунок 3.2 – запуск клиента «HP»



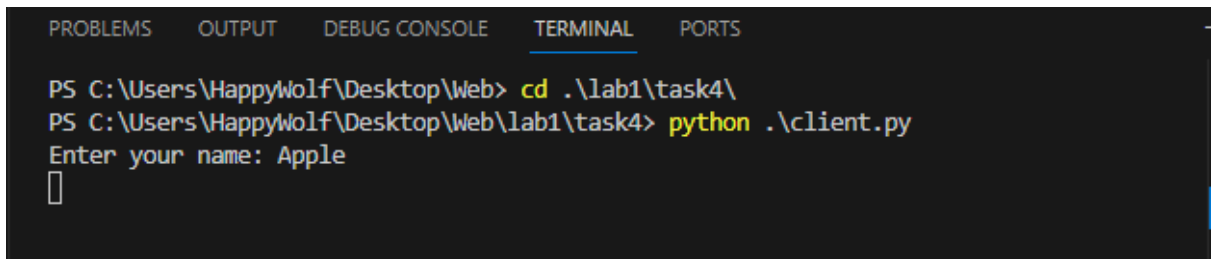
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2283]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\HappyWolf\Desktop\Web>cd .\lab1\task4\

C:\Users\HappyWolf\Desktop\Web\lab1\task4>python client.py
Enter your name: Samsung
█
```

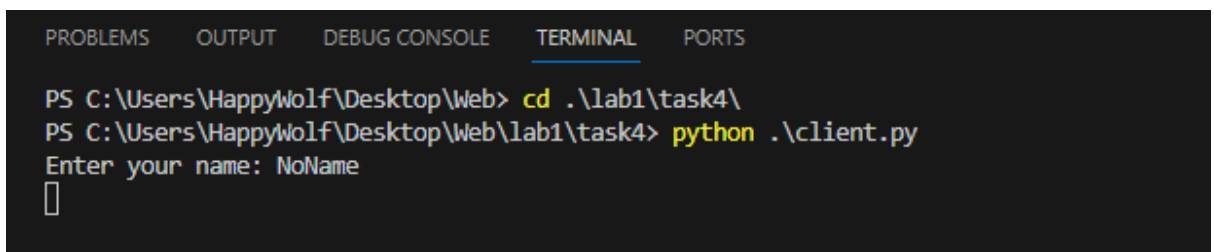
Рисунок 3.3 – запуск клиента «Samsung»



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> cd .\lab1\task4\
PS C:\Users\HappyWolf\Desktop\Web\lab1\task4> python .\client.py
Enter your name: Apple
█
```

Рисунок 3.4 – запуск клиента «Apple»



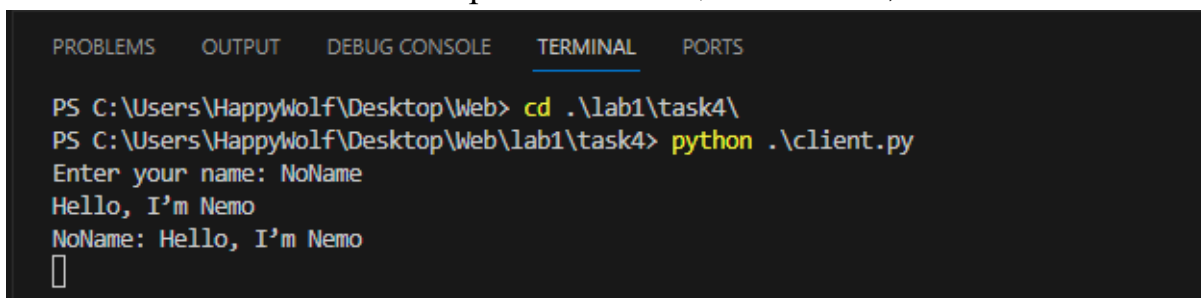
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> cd .\lab1\task4\
PS C:\Users\HappyWolf\Desktop\Web\lab1\task4> python .\client.py
Enter your name: NoName
█
```

Рисунок 3.5 – запуск клиента «NoName»

### 3. Отправка сообщений пользователями

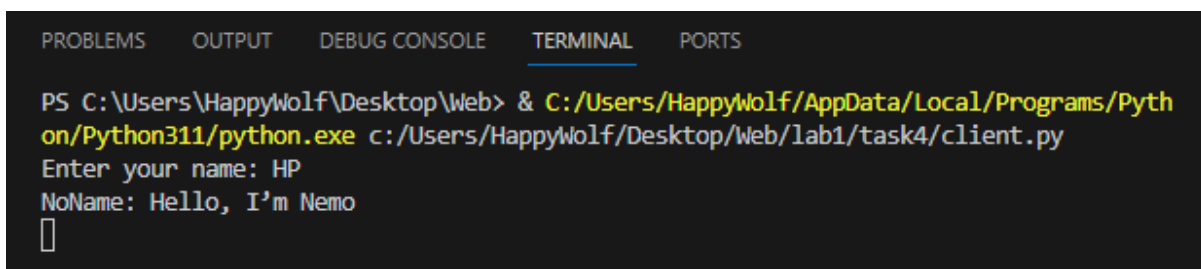
Пользователь «NoName» отправляет сообщение «Hello, I'm Nemo»



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> cd .\lab1\task4\
PS C:\Users\HappyWolf\Desktop\Web\lab1\task4> python .\client.py
Enter your name: NoName
Hello, I'm Nemo
NoName: Hello, I'm Nemo
█
```

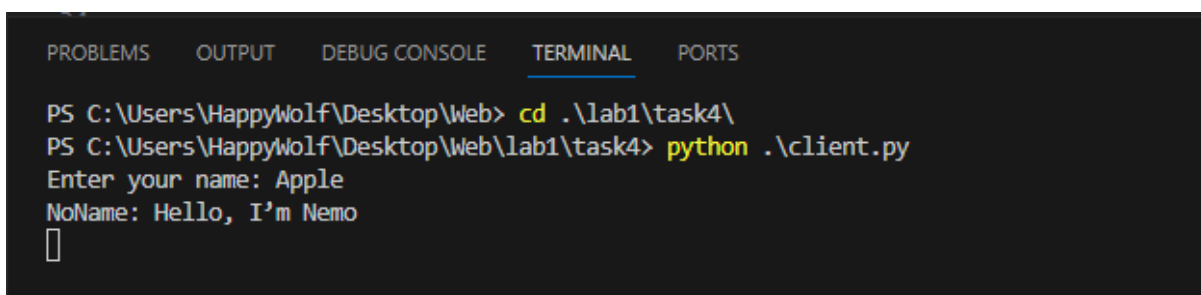
Рисунок 3.6 – состояние клиента «NoName»



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task4/client.py
Enter your name: HP
NoName: Hello, I'm Nemo
█
```

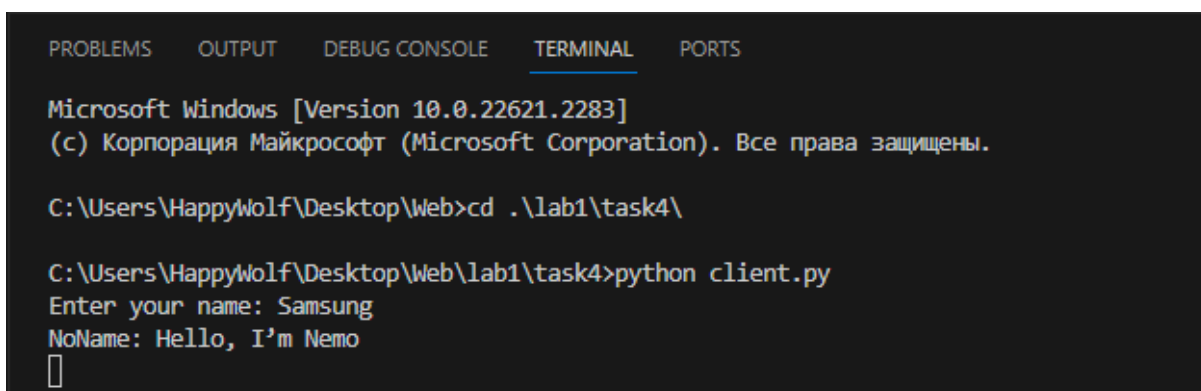
Рисунок 3.7 – состояние клиента «HP»



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> cd .\lab1\task4\
PS C:\Users\HappyWolf\Desktop\Web\lab1\task4> python .\client.py
Enter your name: Apple
NoName: Hello, I'm Nemo
█
```

Рисунок 3.8 – состояние клиента «Apple»



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2283]
(с) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

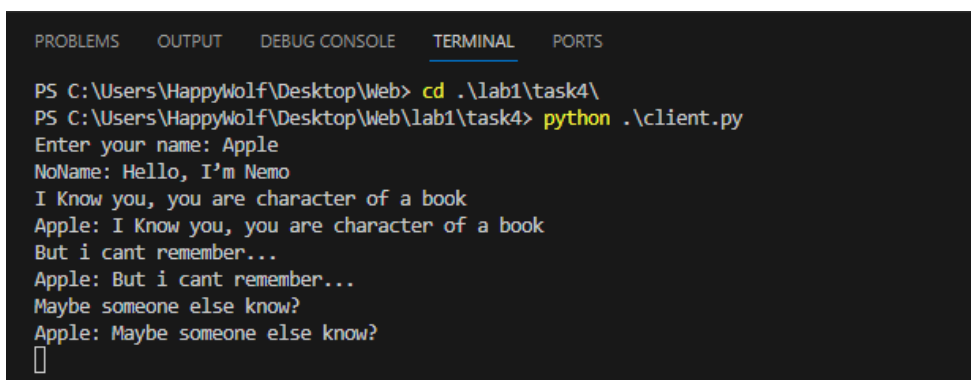
C:\Users\HappyWolf\Desktop\Web>cd .\lab1\task4\

C:\Users\HappyWolf\Desktop\Web\lab1\task4>python client.py
Enter your name: Samsung
NoName: Hello, I'm Nemo
█
```

Рисунок 3.9 – состояние клиента «Samsung»

#### 4. Отправка сообщений пользователями

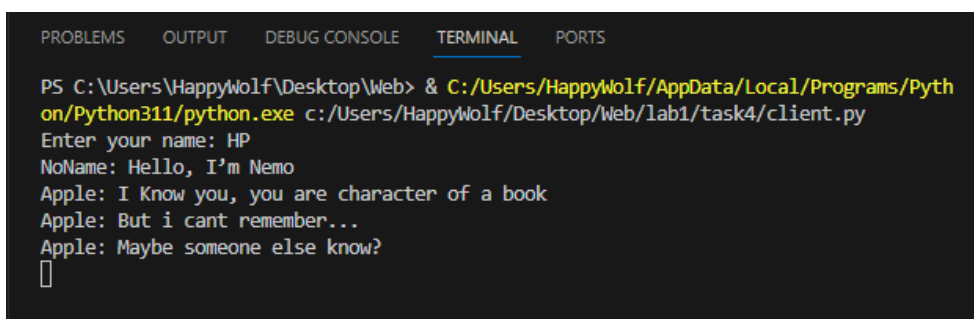
Пользователь «Apple» отправляет множество случайных сообщений в чат



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> cd .\lab1\task4\
PS C:\Users\HappyWolf\Desktop\Web\lab1\task4> python .\client.py
Enter your name: Apple
NoName: Hello, I'm Nemo
I Know you, you are character of a book
Apple: I Know you, you are character of a book
But i cant remember...
Apple: But i cant remember...
Maybe someone else know?
Apple: Maybe someone else know?
█
```

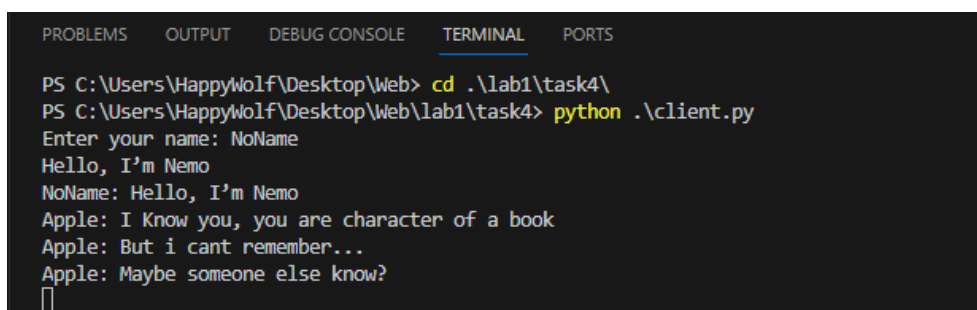
Рисунок 3.10 – состояние клиента «Apple»



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> & C:/Users/HappyWolf/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HappyWolf/Desktop/Web/lab1/task4/client.py
Enter your name: HP
NoName: Hello, I'm Nemo
Apple: I Know you, you are character of a book
Apple: But i cant remember...
Apple: Maybe someone else know?
█
```

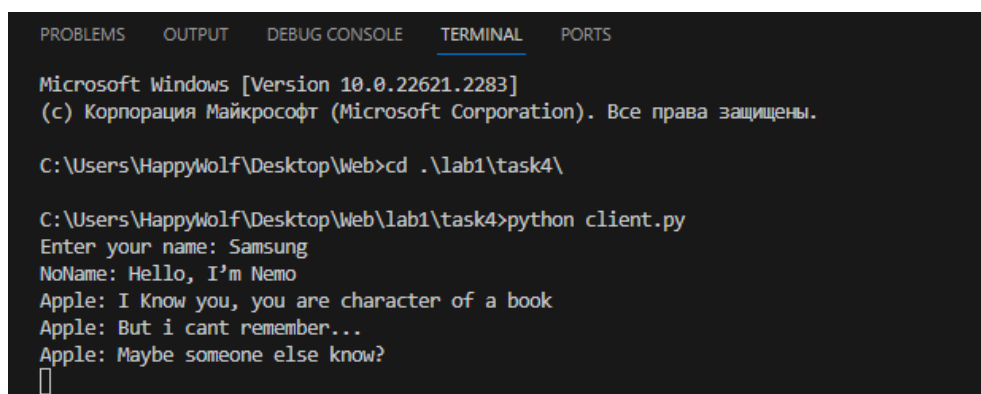
Рисунок 3.11 – состояние клиента «HP»



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HappyWolf\Desktop\Web> cd .\lab1\task4\
PS C:\Users\HappyWolf\Desktop\Web\lab1\task4> python .\client.py
Enter your name: NoName
Hello, I'm Nemo
NoName: Hello, I'm Nemo
Apple: I Know you, you are character of a book
Apple: But i cant remember...
Apple: Maybe someone else know?
█
```

Рисунок 3.12 – состояние клиента «NoName»



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2283]
(с) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\HappyWolf\Desktop\Web>cd .\lab1\task4\

C:\Users\HappyWolf\Desktop\Web\lab1\task4>python client.py
Enter your name: Samsung
NoName: Hello, I'm Nemo
Apple: I Know you, you are character of a book
Apple: But i cant remember...
Apple: Maybe someone else know?
█
```

Рисунок 3.13 – состояние клиента «Samsung»



## **Вывод**

В данной лабораторной работе были приобретены начальные практические навыки и умения по реализации web-серверов и использования сокетов. Произведено ознакомление с библиотекой socket для ЯП Python.