

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Web-программирование

Отчет

Лабораторная работа №1

Выполнил:

До Ван Тхиен

К33401

Проверил:

Говоров А. И.

Санкт-Петербург

2021 г.

Задача

1. **Цель:** овладеть практическими навыками и умениями реализации web-серверов и использования сокетов.
2. **Оборудование:** компьютерный класс.
3. **Программное обеспечение:** Python 2.7-3.6, библиотеки Python: sys, socket.

Ход работы

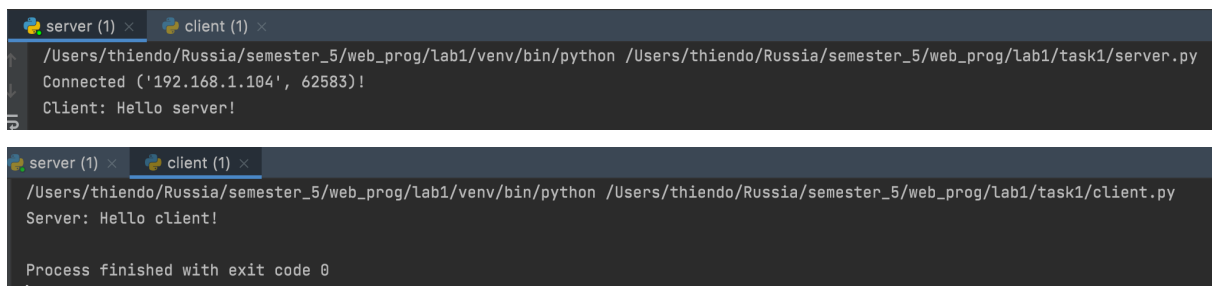
1. **Задание 1:** Реализовать клиентскую и серверную часть приложения.
Клиент отправляет серверу сообщение «Hello, server». Сообщение должно отразиться на стороне сервера. Сервер в ответ отправляет клиенту сообщение «Hello, client». Сообщение должно отобразиться у клиента.
- Северная часть

```
1  import socket
2
3  PORT = 16900
4  HOST = socket.gethostname()
5
6  serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7  serversocket.bind((HOST, PORT))
8  serversocket.listen(5)
9
10 while True:
11     try:
12         conn, addr = serversocket.accept()
13         print(f"Connected {addr}!")
14         msg = ""
15         data = conn.recv(1024)
16         if not data:
17             break
18         msg += data.decode("utf-8")
19         print("Client: " + msg)
20         conn.send(b"Hello client!")
21     except:
22         serversocket.close()
23         break
24
```

- Клиентская часть

```
1 import socket
2
3 serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 serversocket.connect((socket.gethostname(), 16900))
5
6 serversocket.send(b"Hello server! \n")
7 data = serversocket.recv(1024)
8 print("Server: " + data.decode("utf-8"))
9 serversocket.close()
10
```

- Результат



```
server (1) x client (1) x
/Users/thiend/Russia/semester_5/web_prog/lab1/venv/bin/python /Users/thiend/Russia/semester_5/web_prog/lab1/task1/server.py
Connected ('192.168.1.104', 62583)!
Client: Hello server!

server (1) x client (1) x
/Users/thiend/Russia/semester_5/web_prog/lab1/venv/bin/python /Users/thiend/Russia/semester_5/web_prog/lab1/task1/client.py
Server: Hello client!

Process finished with exit code 0
```

2. **Задание 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. Поиск площади трапеции.
- Северная часть

```

1  import socket
2  import pickle
3
4  PORT = 14900
5  HOST = socket.gethostname()
6
7  serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8  serversocket.bind((HOST, PORT))
9  serversocket.listen(5)
10
11 while True:
12     try:
13         conn, addr = serversocket.accept()
14         print(f"Connected {addr}!")
15         msg = ""
16         data = conn.recv(1024)
17         if not data:
18             break
19         msg += data.decode("utf-8")
20         print("Client: " + msg)
21         conn.send(b"Hello client!")
22
23         data_trapezoid = conn.recv(1024)
24         trapezoid = pickle.loads(data_trapezoid)
25         area = 1/2*trapezoid[2]*(trapezoid[0] + trapezoid[1])
26         conn.send(pickle.dumps(area))
27
28     except:
29         serversocket.close()
30         break
31

```

- Клиентская часть

```

1  import socket
2  import pickle
3
4  serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5  serversocket.connect((socket.gethostname(), 14900))
6
7  serversocket.send(b"Hello server! \n")
8  data = serversocket.recv(1024)
9  print("Server: " + data.decode("utf-8"))
10
11
12  print("Enter the lengths of the side of the trapezoid!")
13  print("(Form: Top_edge Bottom_edge High_line)")
14  trapezoid = list(map(int, input("Input: ").strip().split()))[:3]
15
16  data_trapezoid = pickle.dumps(trapezoid)
17  serversocket.send(data_trapezoid)
18  data_area = serversocket.recv(1024)
19  area = pickle.loads(data_area)
20  print("The area of the trapezoid: " + str(area))
21  serversocket.close()
22

```

- Результат

```

server x client x
/Users/thiend0/Russia/semester_5/web_prog/lab1/venv/bin/python /Users/thiend0/Russia/semester_5/web_prog/lab1/task2/server.py
Connected ('127.0.0.1', 62600)!
Client: Hello server!

server x client x
/Users/thiend0/Russia/semester_5/web_prog/lab1/venv/bin/python /Users/thiend0/Russia/semester_5/web_prog/lab1/task2/client.py
Server: Hello client!
Enter the lengths of the side of the trapezoid!
(Form: Top_edge Bottom_edge High_line)
Input: 3 3 4
The area of the trapezoid: 10.0
Process finished with exit code 0

```

3. Задание 4: Реализовать многопользовательский чат.

- Северная часть

```

1  import socket
2  import threading
3
4  PORT = 7000
5  HOST = '127.0.0.1'
6  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7
8  try:
9      sock.bind((HOST, PORT))
10 except socket.error as e:
11     print(str(e))
12
13     print(".....")
14     sock.listen(5)
15
16     clients = []
17     names = []
18
19 def send_mess(message):
20     for client in clients:
21         client.send(message)
22
23 def handle(client):
24     while True:
25         try:
26             message = client.recv(1024)
27             send_mess(message)
28         except:
29             index = clients.index(client)
30             clients.remove(client)
31             client.close()
32             names.remove(names[index])
33             break
34
35 def thread_client():
36     while True:
37         client, addr = sock.accept()
38         print("Connected with " + str(addr))
39         client.send("NAME".encode('utf-8'))
40         name = client.recv(1024).decode('utf-8')
41         names.append(name)
42         clients.append(client)
43         print("Name is " + name)
44         send_mess("{} joined. \n".format(name).encode('utf-8'))
45         client.send("Connected to server.".encode('utf-8'))
46         thread = threading.Thread(target=handle, args=(client, ))
47         thread.start()
48
49     thread_client()

```

- Клиентская часть

```
1  import socket
2  import threading
3
4  PORT = 7000
5  HOST = '127.0.0.1'
6  clientSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7  clientSock.connect((HOST, PORT))
8
9  name = input("Choose your name: ")
10
11 def receive():
12     while True:
13         try:
14             message = clientSock.recv(1024).decode('utf-8')
15             if message == "NAME":
16                 clientSock.send(name.encode('utf-8'))
17             else:
18                 print(message)
19         except socket.error as e:
20             print(str(e))
21             break
22
23 def write_mess():
24     while True:
25         message = "{} : {}".format(name, input())
26         clientSock.send(message.encode('utf-8'))
27
28 receive_thread = threading.Thread(target=receive)
29 receive_thread.start()
30 write_thread = threading.Thread(target=write_mess)
31 write_thread.start()
```

- Результат

```
server (2) x client1 x client2 x client3 x
/Users/thiend0/Russia/semester_5/web_prog/lab1/venv/bin/python /Users/thiend0/Russia/semester_5/web_prog/lab1/task4/server.py
.....
Connected with ('127.0.0.1', 62671)
Name is A
Connected with ('127.0.0.1', 62676)
Name is B
Connected with ('127.0.0.1', 62678)
Name is C

server (2) x client1 x client2 x client3 x
/Users/thiend0/Russia/semester_5/web_prog/lab1/venv/bin/python /Users/thiend0/Russia/semester_5/web_prog/lab1/task4/client1.py
Choose your name: A
A joined.
Connected to server.
B joined.

C joined.

A : hi
B : hi
C : hello

server (2) x client1 x client2 x client3 x
/Users/thiend0/Russia/semester_5/web_prog/lab1/venv/bin/python /Users/thiend0/Russia/semester_5/web_prog/lab1/task4/client2.py
Choose your name: B
B joined.
Connected to server.
C joined.

A : hi
B : hi
C : hello
|

server (2) x client1 x client2 x client3 x
/Users/thiend0/Russia/semester_5/web_prog/lab1/venv/bin/python /Users/thiend0/Russia/semester_5/web_prog/lab1/task4/client3.py
Choose your name: C
C joined.

Connected to server.
A : hi
B : hi
hello
C : hello
```

Вывод

- Работал с сокетами, использовал библиотеки Socket, Threading, Pickle Python.