ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет «Инфокоммуникационных технологий»

Направление подготовки «09.03.03 Мобильные и сетевые технологии»

О Т Ч Е Т

Тема задания: Работа с сокетами

Выполнил

студент: Сурина Елизавета Сергеевна К33402
(Фамилия И.О) номер группы

Проверил:
Преподаватель Говоров Антон Игоревич
(Фамилия И.О)

Санкт-Петербург

2020

ЛАБОРАТОРНАЯ РАБОТА №1

Работа с сокетами

Цель: овладеть практическими навыками и умениями реализации web-серверов и использования сокетов.

Практическое задание:

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

```python
import socket

sock = socket.socket()
sock.bind(('', 9090))
sock.listen(10)

conn, addr = sock.accept()
data = conn.recv(16384)
udata = data.decode("utf-8")
print("Client: " + udata)
conn.send(b"Hello, client.\n")

conn.close()
```

скриншот 1 - код сервера

```python
import socket

conn = socket.socket()
conn.connect(('localhost', 9090))
conn.send(b"Hello, server. \n")

data = conn.recv(1024)
msg = data.decode("utf-8")
print("Server: " + msg)

conn.close()
```

скриншот 2 - код клиента

Результат:

```
(venv) C:\Users\Kwin\PycharmProjects\web>py -3.7 server.py
Client: Hello, server
```

```
(venv) C:\Users\Kwin\PycharmProjects\web>py -3.7 client.py
Server: Hello, client.
```

2. Вариант 15 - с. Поиск площади трапеции.

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.

```python
import socket
import pickle

BUFFER_SIZE = 4096

sock = socket.socket()
sock.bind(('', 9090))
sock.listen(10)

conn, addr = sock.accept()

all_data = bytearray()
data = conn.recv(BUFFER_SIZE)

all_data += data

obj = pickle.loads(all_data)
print('Obj:', obj)
result = 0.5 * (obj['a'] + obj['b']) * obj['h']
print(result)
conn.send(str(result).encode())

conn.close()
```

скриншот 3 - код сервера

```python
import socket
import pickle

conn = socket.socket()
conn.connect(('localhost', 9090))

obj = {
    'a': int(input("основание a = ")),
    'b': int(input("основание b = ")),
    'h': int(input("высота h = "))
}

data = pickle.dumps(obj)

conn.sendall(data)

da = conn.recv(4096)
msg = da.decode()
print("Площадь трапеции равна: " + msg)

conn.close()
```

скриншот 4 - код клиента

Результат:

```
(venv) C:\Users\Kwin\PycharmProjects\web>py server.py
Obj: {'a': 12, 'b': 2, 'h': 5}
```

```
(venv) C:\Users\Kwin\PycharmProjects\web>py client.py
основание a = 12
основание b = 2
высота h = 5
Площадь трапеции равна: 35.0
```

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.

```python
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(("localhost", 9090))
sock.listen(10)

conn, add = sock.accept()
file = open("index.html", "r")
text = file.read()
conn.send('HTTP/1.0 200 OK\nContent-Type: text/html\n\n{}'.format(text).encode())
sock.close()
```

скриншот 5 - код сервера

```python
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("localhost", 9090))
data = conn.recv(2048)
msg = data.decode("utf-8")
print(msg)
conn.close()
```

скриншот 6 - код клиента

Результат:

```
(venv) C:\Users\Kwin\PycharmProjects\web>py client.py
HTTP/1.0 200 OK
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My html page</title>
</head>
<body>

    <p>
        Today is a beautiful day. We go swimming and fishing.
    </p>

    <p>
         Hello there. How are you?
    </p>

</body>
</html>
```

4. Реализовать двухпользовательский или многопользовательский чат.

```
import socket
import threading

debug = True
running = True

HOST = "localhost"
PORT = 9090
maxClient = 999
BUFSIZE = 1024

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind((HOST, PORT))
conn.listen(maxClient)

CONNS = []


def printed(aString):
    if debug:
        print(aString)
```

скриншот 7

```
class talkToClient(threading.Thread):
    def __init__(self, clientSock, addr):
        self.clientSock = clientSock
        self.addr = addr
        threading.Thread.__init__(self)

    def run(self):
        while True:
            recvData = self.clientSock.recv(BUFSIZE)
            if recvData == "stop":
                break
            printed('Client ' + str(self.addr) + ' say "' + str(recvData) + '"')
            for c in CONNS:
                c.sendData(recvData)
        self.clientSock.close()

    def sendData(self, data):
        print('sending to ', self.addr)
        self.clientSock.send(data)
```

скриншот 8

```
while running:
    printed('Running on ' + HOST + ': ' + str(PORT) + '.')
    channel, details = conn.accept()
    printed('Connect on : ' + str(details))
    CONNS.append(talkToClient(channel, details))
    print(CONNS)
    CONNS[-1].start()

conn.close()
```

скриншот 9 - код сервера

```
import socket
import sys

debug = True

conn = None
running = True

HOST = "localhost"
PORT = 9090
BUFSIZE = 1024

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((HOST, PORT))

while running:
    sendDat = input(">> ")
    line = "\n"
    sendData = "client: " + sendDat + line
    sock.send(bytes(sendData, 'utf-8'))
    if sendDat == "stop":
        sys.exit()

    recvData = sock.recv(BUFSIZE)
    print(recvData.decode())

sock.close()
```

скриншот 10 - код клиента

Результат:

```
(venv) C:\Users\Kwin\PycharmProjects\web>py server.py
Running on localhost: 9090.
Connect on : ('127.0.0.1', 62249)
[<talkToClient(Thread-1, initial)>]
Running on localhost: 9090.
Connect on : ('127.0.0.1', 62250)
[<talkToClient(Thread-1, started 5708)>, <talkToClient(Thread-2, initial)>]
Running on localhost: 9090.
Connect on : ('127.0.0.1', 62251)
[<talkToClient(Thread-1, started 5708)>, <talkToClient(Thread-2, started 11288)>, <talkToClient(Thread-3, initial)>]
Running on localhost: 9090.
Client ('127.0.0.1', 62249) say "b'client: Hello\n'"
sending to  ('127.0.0.1', 62249)
sending to  ('127.0.0.1', 62250)
sending to  ('127.0.0.1', 62251)
Client ('127.0.0.1', 62250) say "b'client: Hi\n'"
sending to  ('127.0.0.1', 62249)
sending to  ('127.0.0.1', 62250)
sending to  ('127.0.0.1', 62251)
Client ('127.0.0.1', 62251) say "b'client: hello!!\n'"
sending to  ('127.0.0.1', 62249)
sending to  ('127.0.0.1', 62250)
sending to  ('127.0.0.1', 62251)
Client ('127.0.0.1', 62249) say "b'client: \n'"
sending to  ('127.0.0.1', 62249)
sending to  ('127.0.0.1', 62250)
```

```
(venv) C:\Users\Kwin\PycharmProjects\web>py client.py
>> Hello
client: Hello

>>
client: Hi
client: hello!!
```

Вывод: В результате выполнения лабораторной работы, мне удалось овладеть практическими навыками и умениями реализации web-серверов и использования сокетов.